Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

Description of Goods and Services

SuperMUC-NG

(August 21, 2017)

Aug 2017 **LRZ-Bericht 2017-2**

1 General information on the RFP

1.1 Introduction

Apart from theory and experiment, simulations on HPC systems are now established as indispensable instruments for most scientific disciplines as well as for industry. This was one reason for the German Federal Government and for the States of Baden-Wuerttemberg, Bavaria, and North Rhine-Westphalia for creating the Gauss Centre for Supercomputing (GCS) as a leadership HPC infrastructure for science in Germany and Europe in 2007. As part of this effort, three multi-petascale systems were installed in staggered order at the national centres JSC in Juelich, HLRS in Stuttgart and LRZ in Garching. Also, a technical diversity of the GCS systems, programming models and system software was ensured.

A second round of funding for the three centres is announced for the timeframe of 2017-2025, under the project designation SiVeGCS.

The current system operated by LRZ is SuperMUC, a 6.6 PFlop/s system from IBM and Lenovo. It is comprised of standard Xeon architecture nodes and uses a novel form of direct warm water cooling developed by IBM, which made it one of the most energy-efficient systems in the world.

SuperMUC is used for a broad range of applications from many areas of science, in particular astrophysics, plasma and high energy physics, earth and environmental sciences, life and material sciences, engineering as well as computational fluid dynamics. Information about the scientific projects and results from SuperMUC can be found in the book series

High Performance Computing in Science and Engineering-Garching/Munich,

which are also electronically available[1].

The successor of the current system at LRZ, codenamed SuperMUC-NG ("SuperMUC Next Generation"), is again intended for use as a national Tier-1 high performance computing resource as well as a European Tier-0 system in the context of the PRACE research infrastructure[2]. SuperMUC-NG will be operated by the Leibniz Supercomputing Centre (LRZ) of the Bavarian Academy of Sciences and Humanities on behalf of the Gauss Centre for Supercomputing. SuperMUC-NG is expected to deliver outstanding international application performance for a wide range of user applications and to achieve one of the top positions among supercomputers worldwide.

[1] https://www.lrz.de/services/compute/supermuc/magazinesbooks/index.html#Books

[2] http://www.prace-ri.eu/

1.2 General objectives for the SuperMUC-NG Tier-0/Tier-1 system

The range of applications targeted for deployment on SuperMUC-NG will be comparable to that on the current system, and is expected to widen into additional scientific domains. Also, the future job profile will impose demands on compute performance, memory size and bandwidth, network latency and bandwidth as well as storage capacity and bandwidth of the system it executes on, that cannot be fulfilled by the currently deployed technology. SuperMUC-NG should therefore be a system characterized by leading-edge performance and reliability in all aspects of its delivered hardware.

The amount of data produced by traditional HPC and upcoming Big Data applications requires data centric memory and storage hierarchies. Furthermore, a robust and scalable software stack and programming environment is of major importance. It should enable users to build, debug and reliably run applications that use the entire system.

It is the explicit goal of the LRZ to provide these services without imposing many additional constraints on its current and future user base. To this end, only a computing system that is usable in a highly versatile manner will be considered for acquisition. Special-purpose architectures that offer advantages for only a small subset of algorithms and applications will not be taken into consideration. Following this, the primary goal for the procurement of SuperMUC-NG is:

> To establish a reliable and versatile leadership-class system for large-scale compute and data-intensive workloads.

With an expected compute power of SuperMUC-NG Phase 1 an order of magnitude higher than the current system, efficient execution of large jobs is a matter of great importance. However, extensive parameter studies using many small and medium-sized jobs are very common in many scientific areas, e.g. replica exchange methods in molecular dynamics or data processing of Large Hadron Collider experiments. Furthermore, a steadily growing number of projects demand the support of complete application workflows, or virtualization concepts such as Cloud Computing. Efficient support of such user requirements should therefore be possible on the new supercomputer.

The use of cutting-edge numerical algorithms as well as careful program development and debugging are mandatory requirements for the efficient usage of modern supercomputer resources, particularly, regarding the proficiency that modern multi- and many-core processors demand with respect to parallel program design. Within this context, the theoretical peak performance of the system is less important than sustained system performance for the entire spectrum of applications of LRZ’s current and future user base. For the assessment of the SuperMUC-NG application performance all Tenderers must commit to performance numbers for a set of real application as well as synthetic benchmarks. Please refer to the document *“Decision Criteria and Benchmark Description SuperMUC-NG”* for further details on the SuperMUC-NG benchmark suite and evaluation schemes.

Given the usage profile of the current SuperMUC system, the acquisition goal is to install a heterogeneous supercomputer based on general purpose compute nodes, and, to a small extent, of compute nodes with large shared memory. A significant part of the system may be composed of many-core or accelerated compute nodes. All compute nodes must be connected via a common high-speed network.

Because programs will have run times up to several days, the offered SuperMUC-NG system must provide a high degree of stability and usability under permanent load with changing usage profiles. Finally, reliable file systems are required to ensure the integrity and availability of scientific data.

1.3 LRZ System Concept

In its concept for SuperMUC-NG, LRZ starts out from the assumption that many current and future projects will tackle problem sizes with parallel code that can no longer be completed in reasonable time on a quarter of the current system. The system design of SuperMUC-NG must ensure that such applications can be executed with leading edge application performance. However, it should not negatively impact medium sized parameter studies and high throughput requirements.

LRZ envisions that the system will be composed of **islands of highly interconnected compute nodes with non-blocking or nearly optimal communication links between all nodes within an island.** It is desired that each island either contains at least **1024 compute nodes** or provides a (double precision floating point) peak performance of at least **1.5 PFlop/s.**

The link bandwidth and the number of links for the **communication between these islands** may have smaller aggregate bandwidth than within an island **(pruned interconnect[3])** depending on the involved costs and the deployed technology. The rationale and experience underlying the pruned interconnect requirement between islands are:

- Many communication patterns are of nearest-neighbor type and do not need a full set of links across islands due to the volume-surface ratio of data exchange.
- Large-scale applications, which need more cores than one island provides, may be scheduled round-robin over all or many islands and may consume the link bandwidth that is not used by other applications that are running entirely in one island.

The above considerations assume a fat tree network topology. However, **other implementations of network topologies** such as multi-dimensional torus or dragonfly network topologies are also considered acceptable, especially if no substantial commercial or technical advantage can be gained by implementing a pruned fat tree network topology.

It is expected that **topology-aware placement** of jobs will be necessary for most solutions, handling of which should be flexible and easy to manage. It is also desired that **adaptive routing** algorithms prevent contention within the interconnect network, as well as provide failover or fast re-routing in case of hardware failures. Provisions for controlling the **quality of service** for specific services and controlling congestion are also of interest.

For the bulk of the system, two options may be considered, either **stand-alone many-core nodes** or **thin shared memory nodes** that optionally may contain **one or more accelerators („GPGPUs“).**

Stand-alone many-core nodes should contain at least 96 GBytes of main memory and might also contain a sizeable portion of **high bandwidth memory.**

[3] The term “pruned interconnect” is used for tree topologies in which the connection of the highest switch level hierarchies is realized with a reduced number of network links compared to a fat tree.

Thin shared memory nodes that optionally may contain **one or more accelerators (“GPGPUs”)** should contain at least 40 physical cores of a general-purpose architecture, and sufficient memory to ensure that at least 2 GBytes per core are available to an MPI application in user space if the node is completely filled with MPI tasks. For accelerated nodes, it must be assured that ease of programming and efficient use are not significantly curtailed compared to a general-purpose architecture. In particular, it may be necessary to equip accelerated nodes with significantly more main memory to achieve proper balancing of compute power against memory requirements of applications.

Whatever node design is chosen for the bulk of the system, the **general purpose compute performance** (i.e. excluding accelerators or many-core nodes) of SuperMUC-NG Phase 1 **should not be less than** the compute performance of SuperMUC Phase 2.

The aggregate size of the DDR-based main memory of the Phase 1 system should exceed 500 TBytes.

To provide similar shared memory characteristics as those presently available on the **"fat" node** island of SuperMUC Phase 1, the new system should supply a number of **large shared memory nodes.**

Message passing will continue to be the dominant programming paradigm and will be used for coarse-grained parallelism. Consequently, the system must be equipped with an **efficient internal interconnect** as well as an **efficient MPI implementation** that makes optimal use of the hardware and optionally off-loads workload to the network devices. Efficient implementations for the evolving **PGAS (Partitioned Global Address Space)** programming languages are desired.

The experience of LRZ shows that efficient parallelization is easier to achieve with a moderate number of high-performance shared-memory computational units than with many low-performance units. Therefore, most if not all users will utilize **the hybrid programming model** with MPI between such units, and auto-parallelization, OpenMP or OpenACC within the units. Both experiences and theoretical considerations show that for achieving optimal performance with the hybrid model, a **fully thread-safe** implementation of all relevant libraries (specifically MPI) is essential.

A **complete HPC software stack** necessary for development and deployment of scientific and technical applications must be available (a Linux-based operating system suitable for deployment on the targeted system, highly efficient system drivers for interconnect and I/O, highly efficient compilers, optimized libraries, tools, ISV applications). Support for efficient execution of the most commonly used scripting languages and problem solving environments is considered important. **Domain specific languages, machine learning and scalable data analytics** will become increasingly important on HPC systems.

Rapid deployment of applications can be aided by **virtualization** concepts; it should therefore be possible to pack an application and its dependencies into a virtual **container**. This concept of HPC node virtualization will give users the opportunity to tailor and adapt the run-time environment to the needs of their applications.

Together with the compute nodes, the Tenderer must offer sufficiently **large, efficient, secure and reliable storage systems** for user data. The **integrity of the data** must be assured, since in the many-petabyte regime dump-restore of file systems is no longer feasible. Advances in software technologies will be necessary for both the management and efficient usage of the storage subsystems.

The usage profile calls for two types of storage:

- A **parallel file system,** which should contain temporary data and the large result files (termed SCRATCH and WORK). For these data, **high I/O bandwidth** and moderate metadata performance are needed. Specialized support for efficient parallel I/O to a single file from multiple nodes is expected. This storage component must be integrated with the system via the high-performance network.
- Highly reliable **home and project file systems** for user data (HOME and PROJECT), such as source and input files, libraries, configuration and simulation data sets. Metadata performance should be very high, while bandwidth may be moderate. The requirements for reliability, availability and for data safety are very high. In addition, **the accessibility of user data should not depend on the operating condition of SuperMUC-NG**. LRZ has excellent experiences with IBM Spectrum Scale storage (formerly GPFS) and is operating and extending a GPFS-based **Data Science Storage (DSS)** for this purpose.

The leverage of **cloud mechanisms** via a **login cloud** should provide universal and simplified access to the HPC resources and the data residing on the system. Main goals for the login cloud are:

- To provide the users a programming environment
- To give users access control for their data, also for access from the outside world
- To give users the possibility to construct portals for access
- To provide means for pre- and post-processing, and for remote visualization

Only the hardware needed for the cloud (including hardware maintenance) need to be offered within this procurement, and some level of support in interfacing the cloud component with the computational system.

Outside the scope of this procurement, **systems for archiving and backup** will be acquired. It is required that these systems are efficiently connected and operated together with SuperMUC-NG. Therefore, the necessary hardware and software interfaces for this connection are part of this procurement.

Finally, a **software porting and transition path** from the current system to the new one should be established. The tradeoff between potential performance gains and the necessity as well as the extent of algorithmic changes should be assessable.

1.4 Potential sources for performance increase

LRZ considers the following technologies as potential sources for further increasing the performance of HPC systems:

- Growth of the number of cores and threads within general purpose nodes
- Accelerated nodes (GPUs or many-core)
- Further increase of peak performance by means of vectorization
- High Bandwidth Memory, whether directly addressed or used as large cache, in addition to or in lieu of DRAM
- Use of non-volatile memory/storage class memory for efficient I/O
- Faster and better-integrated interconnects
- New interconnect topologies, dynamic routing and resource scheduling
- Fast connection links between node-attached accelerators and host processors (e.g. NVLINK)
- Improved energy efficiency
- Improved scalability

On the software side LRZ sees the following important technologies:

- Extensions of directive-based (OpenMP, OpenACC) programming models for accelerated node architectures
- Extension of existing development, debugging and tuning tools to encompass accelerated node architectures while maintaining ease of use
- Support for the efficient management of data staging between various levels of storage, and for the management of checkpoint data that need to be available for a new job instance
- Availability of resilience features in the batch scheduling component as well as in the parallel programming APIs/semantics to enable continuing job execution even if the nodes assigned to that job fail
- Availability of domain specific libraries and frameworks
- Programming methods that permit to reduce the message passing overheads (MPI+X)
- Implementation of e.g. PGAS programming languages as a complement to message passing for higher programming productivity on distributed memory architectures.

The response to this RFP should take these technologies into consideration. The above items are not intended to limit the system selection; other innovative approaches that might realistically be deployed in the procurement time frame can become part of an offer. Also, the order in which the above items appear do not imply a priority.

1.5 Two-phase installation

The installation of SuperMUC-NG shall proceed in **two phases**:

- Installation of **Phase 1** of the system should start early in **2018** and be completed by the **end of 2018**.

- **Phase 2**[4] should be installed approximately two and a half years after operational readiness of Phase 1**.** It should deliver a substantial performance increase compared to Phase 1.

Phase 2 may consist of installing **additional nodes or of a replacement** of the Phase 1 system. The upgrade should neither involve an interruption of Phase 1 operation that takes longer than a typical maintenance interval, nor a serious degradation of performance and stability of Phase 1 operation.

For submission of the bid, the Tenderer needs to supply a conceptual description for Phase 2. A commitment of the performance increase must be delivered. The technical possibilities, the required additional porting effort, and the capability of interoperating Phase 1 with the upgrade must be described.

[4] For information about Phase 2 see also „*Anschreiben SuperMUC-NG*“.

1.6 Draft schedule for installation

A **draft schedule** for the Site Preparation and Installation Plan (see Section 3.1.2) for SuperMUC-NG is given in the following table.

Milestone or activity	**Estimated dates and major dependencies**	**Referenced in Section**
Site preparation and installation plan	"At most three months after signing of the contract" Q1/2018	3.1.1
Data Science Storage delivered, installed and accepted	H1/2018	3.5.3
Phase 1 Test system (incl. management) delivered and installed	Several months in advance before "Phase 1 delivery of main hardware"	3.10.2
Phase 1 system management components delivered and installed	Three months before "Phase 1 delivery of main hardware"	3.10.1
Site preparation by vendor	Expected in Q1 2018	
Phase 1 delivery of main hardware	Preferably started in Q2/2018	
Phase 1 of previous LRZ supercomputer is decommissioned	End of January, 2019	
Completion of Phase 1 system installation	Preferably in Q3/2018 Expected in early Q4/2018	
System Readiness ("Betriebsbereitschaft")	Expected November 1, 2018	
Earliest possible acceptance of Phase 1	System Readiness + 30 calendar days	
Latest possible acceptance of Phase 1	System Readiness + 100 calendar days	
Installation of Phase 2	Approximately two and a half years after system readiness of Phase 1	

1.7 Categorization of requirements

All requirements of this procurement that will be relevant for acceptance testing fall into one of three categories; these are indicated by the appearance of colored boxes:

M	Requirements within red boxes and marked by "M" are considered **mandatory**. Offers that do not indicate fulfillment of any mandatory requirement will be excluded from the procurement process.[5] Such requirements include the phrase MUST or MUST NOT.

I	Requirements within yellow boxes and marked by "I" are considered **important** for the operation and usability of the system. Offers that fail to fulfill such a requirement will remain included in the procurement process if they specify alternatives that adequately cover the needed functionality. The proposed alternatives and their implications must be explained in detail. Such requirements include the phrase SHOULD or SHOULD NOT.

T	Requirements within green boxes and marked by "T" are considered **targets** for the proposed system. An offer that fulfils these requirements can achieve a qualitatively better ranking through the evaluation procedure. Such requirements include the words MAY, OPTIONAL, IS PREFERRED or IS DESIRED.

Many requirements contain checkboxes. **All checkboxes must be filled in** as follows:

[X] a checkbox marked by the Tenderer with "X" indicates that the requirement **will be fulfilled** and the feature is included in the Tenderer's offer.

[O] a checkbox marked by the Tenderer with "O" indicates that the requirement **cannot be fulfilled**.

1.8 Requirements with respect to the content of the proposal

For all requirements contained in this document, an offer must indicate whether they can be fulfilled or not. If the former is the case, the offer must provide an explanation how the Tenderer intends to fulfil the given requirements.

If not explicitly stated otherwise, the offer must provide detailed information only for Phase 1.

The answers should refer only to offered goods and services. Answers and explanations must be inserted into the appropriate sections of this document and should be as concise as possible. However, any relevant difference between Phase 1 and Phase 2 must be explained in this description. Wherever appropriate and possible, there should be references to common standards of information technology. Bibliographical references (e.g. brochures and manuals)

[5] cf. "*Anschreiben SuperMUC-NG*"

are desirable but can only be considered as supplementary information i.e. they cannot replace the required answers or explanations.

Unanswered or insufficiently answered questions will be considered as answered in the negative.

The Tenderer is encouraged to provide additional information at or near the end of a subsection. However, the Tenderer should not rephrase/reiterate the requirements.

The Tenderer may hand in the specifications stipulated below in German or English or a mixture of both. The Tenderer should use the provided fields at or near the end of most subsections that are labelled as follows:

“ [Insert text here] “

The supplied text should be clearly structured such that the reader can deduce which requirement is being referred to.

1.9 Sources for requirements

The primary sources of requirements can be grouped into the following categories. LRZ will evaluate how well the proposed solution satisfies the resulting requirements.

Category	**The requirement is specified because …**
ADM	… it is essential for efficiently performing system administration procedures
ENGY	… it enables energy efficient operation of the system
INFRA	… it is necessary to operate the system in the LRZ infrastructure
PERF	… it is essential for achieving high application performance on the system
POL	… it is a prerequisite for adhering to LRZ’s long term goal, management policies, and procedures
PROJ	… it is necessary to assure timely conclusion of project milestones
REL	… it is a prerequisite for reliable and stable operation of the system
STD	… users or administrators expect that the functionality under consideration conforms to published or de facto standards
SUPP	… support, maintenance and bug fixes are essential
TECH	… the system must provide state-of-the-art HPC technology in the specified area to achieve the procurement goals
USER	… it is an essential component of the system’s user interface and enables its efficient and correct operation as well as ease of use

For more details on the evaluation procedure see the document “*Decision Criteria and Benchmark Description SuperMUC-NG”.*

1.10 Checking of requirements

During the acceptance phase of the system the requirements will be checked primarily by the following methods.

Method	**The acceptance procedure will be based on …**
DOC	… checking the supplied documentation of the requirement's features
FUNC	… performing a functionality test
INSP	… inspection of a delivered hardware component
BENCH	… execution of benchmark programs for performance verification

1.11 Additional material

The following non-contained material is referenced in this document:

- Anschreiben SuperMUC-NG
- Bewerbungs- und Vertragsbedingungen SuperMUC-NG.
- The floor plans of the computer rooms.
- Decision Criteria and Benchmark Description SuperMUC-NG.

This material is provided as a separate set of documents in the procurement download area.

1.12 Glossary

This document makes use of the following terms:

Accelerator: Processing hardware to perform some calculations faster than possible on general-purpose central processing units, e.g. GPGPUs.

Accelerated node: A node containing general purpose processors as host processing units and one or more accelerators.

Cluster Export Services (CES): provides highly available file and object services to an IBM Spectrum Scale GPFS™ cluster by using Network File System (NFS), Object, or Server Message Block (SMB) protocols.

Combined Phases 1+2: see **Phases 1+2.**

Compute Nodes: That part of the system used for performing computations. May be one of **Thin Node, Large Shared Memory Node, Accelerated Node** or **Many-Core Node.**

Core: Portion of a Multi- or Many-Core Processor that contains execution units, registers and caches.

CPU: Central Processing Unit.

Data Science Storage (DSS): LRZ’s storage cluster based on **IBM Spectrum Scale** for sharing large scientific data within the HPC ecosystem.

Fat Node: see **Large Shared Memory Node.**

File system: This term is not always used in strict technical sense throughout this document. It describes the collection of file storage for specific purposes and implementation (such as storage for users’ HOME files).

General purpose compute node: In the context of this document, a self-contained unit of hardware each (logical or physical) core of which can execute the instruction sequence generated by a standard C, C++ or Fortran compiler from any standard-conforming source code without use of additional language directives. A broad range of applications should be available and can be run with decent performance. Furthermore, the expectation is that the physical socket encompassing such a node has a significant market share in the server market; this currently implies that many-core architectures are not considered general purpose.

GPGPU: General Purpose Computing on Graphics Processing Unit.

HPC Project: A container – usually connected to a UNIX group – that defines scope, resource assignment and duration of a scientific project via a refereeing process. The duration of such a project is typically limited to a few years.

HPL: HIGH PERFORMANCE LINPACK Benchmark

IOPS: Refers to the I/O metadata operations per second supplied by a storage system down to a client.

Island: Refers to a group of nodes with the best interconnect network characteristics in the system (e.g. blocking factor, bandwidth, latency). Between islands, the network characteristics may be degraded. If the internal network has no hierarchical structure, then the term refers to just an arbitrary grouping of processors (e.g. for use within the benchmarking procedure).

kByte, MByte, GByte, TByte in the context of memory and cache mean 10^3, 10^6, 10^9, 10^{12} Bytes. This also applies for memory/cache related bandwidths.

kiByte, MiByte, GiByte, TiByte or PiByte in the context of disk space mean 2^{10}, 2^{20}, 2^{30}, 2^{40}, 2^{50} Bytes. These procurement documents explicitly use GiByte, TiByte etc. for capacity and bandwidth specifications (including benchmark metrics) in this context.

Large Shared Memory Node: Also termed **fat node**. A node that contains significantly more main memory than a standard “Thin” computational node.

Login Node: System that is deployed as a front end for user access.

Many-Core Node: A node which contains one or more many-core processors as central processing units

Many-Core-Processor: A multi-core processor architecture with a high number of cores (typically more than 60) sharing a random-access memory.

Node: Computer connected to the high-performance network interconnect running a single Linux OS instance. It contains a set of cores and possibly attached accelerator devices that share a random-access memory within the same memory address space.

PGAS: Partitioned Global Address Space, a parallel programming model.

Phase 1: Refers to goods and services delivered for the first installation phase (in 2018/19).

Phase 2: Refers to the **additional** goods and services delivered for the second phase.

Phases 1+2: Refers to aspects of the goods and services which are established through the combination of Phase 1 and Phase 2 (e.g. performance of an application that runs on the entire system)

RAS: Reliability, availability and serviceability.

RDMA: Remote Direct Memory Access. A network feature that enables efficient inter-node memory addressing.

RFP: Request for Procurement.

Server Node: A system used for management purposes.

SSD: Solid State Disk.

SuperMUC-NG: SuperMUC Next Generation.

Thin Shared Memory Node: A shared-memory node with two CPUs and a memory size typical for distributed memory HPC processing.

Tenderer: The company that submits this proposal and, if awarded the contract, is the sole general contractor for the SuperMUC-NG system (also: Vendor, Offeror)

VM: Virtual Machine (used in context of the Cloud component).

2 Project Management Requirements

2.1 Project Management

SuperMUC-NG will be a costly and complex system with significant technological risks and therefore requires professional project management during delivery, installation, system acceptance and early life support.

M 1: The Tenderer must assign a project manager and install a Project Management Office with at least one additional person to offload administrative tasks, ensure a “four-eyes” principle, and provide redundancy.

Check here that this requirement will be fulfilled: [ ]

I 1: The project manager should have previous installation experience with similar projects and hold a valid certification in project management (e.g. IPMA Level B or PMI PMP[6]).

Check here that this requirement will be fulfilled: [ ]

The Tenderers' project manager will work together with LRZ's project manager. Main principles of this collaboration shall be documented in a shared project management plan.

The Tenderer is solely responsible for the main part of delivery and installation but LRZ requests continuous read access to relevant project management documentation. The goal is to make sure that any unclear requirements or problems can be addressed as early as possible and formal acceptance is not delayed unnecessarily.

M 2: The Tenderer must develop and submit a Project Management Plan and must manage the installation and commissioning of the system according to this Plan.

The following documents must be developed, actively maintained and made available to LRZ:

- Project Management Plan, including at least sub-plans for Scope, Schedule, Quality, Risk, Stakeholder, Communications[7] and Project Change Management
- Initial Project Milestone Plan
- Work Breakdown Structure[8]
- Schedule Baseline and Project Schedule[9]
- Risk register of the Tenderer

Check here that these requirements will be fulfilled: [ ]

[6] See: http://de.wikipedia.org/wiki/International_Project_Management_Association

[7] Inhalts- und Umfangs-, Termin-, Qualitäts-, Risiko-, Stakeholder- und Kommunikationsmanagement.

[8] Projektstrukturplan

[9] Original- und tatsächlicher Terminplan

I 2: The Tender should specify which industry standard is followed for the Project Management Plan (e.g., PMBoK, PRINCE2, PMI, etc.)

[Insert text here]

2.2 Risk Assessment

I 3: The Tenderer should comment on the following risks and their probability.

Although some of the risks are within the responsibility of LRZ, Tenderers are asked to provide their assessment and suggestions for possible countermeasures. For all items, a probability estimate of the form High, Medium or Low should be assigned.

I 4: For all items with an assigned probability estimate of Medium or High, a mitigation description should be supplied.

The Tenderer or one of its main suppliers may cease operation e.g. due to bankruptcy. In this case it may not be possible to deliver or operate the system or parts of it.

Comments and mitigation measures: [Insert text here]

The Tenderer or one of its main suppliers may run into financial instability. This may cause delays in the installation process. It may also reduce the availability of human staff on the Tenderers' or its suppliers' side.

Comments and mitigation measures: [Insert text here]

Technology road maps may be considerably delayed or cancelled, Tenderers or one of their suppliers cannot adhere to contracted time schedules e.g. because technical problems delay the availability of components or their interoperability. It may take considerable time to fix these technical problems.

Comments and mitigation measures: [Insert text here]

Contracted technical properties may be not available. The Tenderer or one of its suppliers cannot adhere to contracted technical properties. As can be seen from known issues with global parallel file systems this risk includes (system) software. It may take considerable time to fix these technical problems.

Comments and mitigation measures: [Insert text here]

Unexpected failure of a large number of critical components due to previously unknown problems. This includes defective motherboards, cables, failures of cooling components, etc. It may require extensive replacement programs.

Comments and mitigation measures: [Insert text here]

Operational limits (AC power, cooling, temperature) may be exceeded, or support budget may become overstrained.

Comments and mitigation measures: [Insert text here]

Software scalability limits: Software (e.g. OS, batch system, MPI, parallel file system etc.) may be not mature enough to scale with the size of the system, the number of jobs, etc.

Comments and mitigation measures: [Insert text here]

Although a system passes the acceptance test, it may not be reliable enough for long program runs. Possible causes may be insufficient node reliability, insufficient reliability of the communication network or insufficient reliability of the I/O subsystem (most likely for shared global file systems), frequent disk failures or defect batches of cables or connectors. Our experience with previous systems indicate a **medium to high probability** that such issues will happen.

Comments and mitigation measures: [Insert text here]

Applications do not compile or fail to execute as designed: Examples are programs that do not compile/link and programs that produce wrong results. The main causes are compiler errors, library errors, CPU errors and errors in the communication hardware (including firmware) and communication software.

Comments and mitigation measures: [Insert text here]

Applications suffer from unexpected performance degradation: This may be related to the code generation of a compiler (e.g. an update), to a suboptimal implementation of library routines or missing optimized library routines for special purposes.

Comments and mitigation measures: [Insert text here]

Tools for performance analysis and debugging turn out to be **too unstable or immature for working on** large-scale applications.

Comments and mitigation measures: [Insert text here]

Tax increases (VAT) or the development of the **currency exchange rate** may make the Tenderer’s calculations invalid.

Comments and mitigation measures: [Insert text here]

Qualified personnel is not available during the installation or the early life support phase or changes frequently.

Comments and mitigation measures: [Insert text here]

Other simultaneous installation projects of the Tenderer divert resources and/or qualified personnel from the SuperMUC-NG project.

Comments and mitigation measures: [Insert text here]

Key personnel in the project becomes unavailable.

Comments and mitigation measures: [Insert text here]

Please describe **additional risks** which appear significant, assess these risks, where applicable and describe potential countermeasures.

Comments and mitigation measures: [Insert text here]

3 Requirements for SuperMUC-NG

3.1 Installation requirements

The Tenderer's proposal for installation must ensure that the offered system is physically installable and operable, and it must allow a reliable and comprehensible prediction of air conditioning, cold and warm water cooling demands, and energy consumption.

3.1.1 Proposal for installation

M 3: A technical proposal for the installation for both phases must be provided.

The proposal must include the following items:

- A detailed space assignment plan for Phase 1 (including rack layout draft) and a conceptual plan for a potential Phase 2
- Cabinet dimensions, packaging diagrams, weights (in all configurations – in packaging, dry, with any liquid coolant) and electrical requirements
- Raised floor requirements and cutouts
- Cable tray requirements
- Environmental requirements
- Expected AC power and cooling requirements
- Cooling water quality requirements
- Safety and health requirements

I 5: This criterion has been converted to a mandatory one.

M 3a:

Information about:
- the earliest and latest installation date of Phase 1
- the earliest installation date of Phase 2
- the time needed to complete the installations
- the length of interruption of service of Phase 1 needed for installation of Phase 2

must be submitted.

Table 1 with milestones, activities and dependencies must be provided.

Milestone or activity	**Estimated dates and major dependencies**	**Latest Date**
Site preparation and installation plan		
Data Science Storage delivered, installed and accepted		
Phase 1 Test system (incl. management) delivered and installed		

Phase 1 system management core components delivered and installed		
Phase 1 delivery of main hardware		
Phase 1 of previous LRZ supercomputer is decommissioned and site is available for installation		
Site preparation by vendor		
Phase 1 system installation		
System Readiness (“Betriebsbereitschaft”)		
Date for acceptance of Phase 1		
Estimated start of installation of Phase 2		

Table 1: System installation milestones, activities and dependencies.

[Insert text here]

3.1.2 Site Preparation and Installation Plan

I 6: Not later than three months after signing of the contract, the Tenderer should provide a Site Preparation and Installation Plan. This plan will be updated as necessary during performance of the contract and will be subject to perusal and approval by LRZ.

The plan should cover at least the following items:

- Cabinet dimensions, packaging diagrams, weights (in all configurations – in packaging, dry, with any liquid coolant) and electrical requirements
- System layout and cabling requirements, including expansion options
- Raised floor requirements and cutouts
- Cable tray requirements
- Environmental requirements
- Expected AC power and cooling requirements
- Cooling water quality requirements
- Safety and health requirements
- Equipment delivery schedule
- Staging and temporary storage area needs
- Pre-delivery access and work needs
- Shipping plans
- Equipment movement process, from truck to computer room floor, to final locations
- Equipment layout and installation sequence (for multi-stage deliveries)
- Bring-up plan
- Quality Assurance plan
- Verification and Validation

Check here that this requirement will be fulfilled: [ ]

[Insert text here]

3.1.3 Installation Locations

The compute part of the offered system shall be installed in the HRR[10] computer room of the LRZ data centre building; for other parts of the offered system different rooms are targeted. The building has two **freight elevators** with the following clearance dimensions:

Freight elevator on the west side of the building:

- width x depth x height = 1,95m x 2,95m x 2,58m
- maximum payload = 3000 kg

Freight elevator on the east side of the building:

- width x depth x height = 1,60m x 2,40m x 2,30m
- maximum payload = 2000 kg

In order to enable a transport of the system components to the designated computer room in the third floor, all system components shall not exceed the following dimensions:

- width x depth x height = 1,60m x 1,60m x 2,58m for cubic based components, or

[10] **H**öchstleistungs-**R**echner-**R**aum

- width x depth x height = 1,20m x 2,50m x 2,58m for very deep components
- width x depth x height = 2,50m x 1,20m x 2,58m for very wide components

The floor plan of the designated computer room of LRZ is included with the tender documents. This room is reserved for Phase 1 and Phase 2 of SuperMUC-NG; it is free of pillars and has the following dimensions and false floor specifications:

- width x depth x height = 52,5m x 21m x 8m (= 1102,5m^2 x 8m)
- false floor with a maximum area load of 1500 kg per m^2 (~ 15 kN/m^2), a maximum "*point load*" of 500 kg (5 kN) and a height of 1.8m

After shutdown and decommissioning of SuperMUC Phase 1 the facilities used by SuperMUC Phase 1 will be available for the installation of SuperMUC-NG Phase 1.

After shutdown and decommissioning of SuperMUC Phase 2 the facilities used by SuperMUC Phase 2 will be available for the installation of SuperMUC-NG Phase 2.

A floor plan containing details of this area is included in the tender documents.

M 4: Note – this requirement has been modified:

The transportation of the offered system (this includes Phase 1 and Phase 2) to the place of installation must be feasible under the constraints mentioned above.

The bulk of the compute part of Phase 1 of the offered system must be installable in the currently free area of the HRR computer room. This includes the constraints for floor space, power and cooling, maintenance areas. If the area or point load constraints are violated, the necessary reinforcements for the false floor and the transportation path must be part of the offer.

By checking the box, the Tenderer also declares that the complete installation process and preparation for operation of the system (for both installation phases) is incumbent on the Tenderer[11].

Check here that this requirement will be fulfilled: **[ ]**

If the offer for the compute nodes does not fit completely into the currently free area, it may be possible to supply additional space for it by removing the southernmost two rows of SuperMUC phase 1 compute nodes somewhat earlier than end of 2018.

M 4a Note - new Requirement:

The Tenderer must indicate whether this additional area (and how much of it) will be needed, and when it will be needed.

Some components of the system (in particular, see sections 3.5.8 and 3.14.2) should be installed one floor level lower, in the NSR0 or NSR1 computer rooms. The following table provides an overview of the facilities available there:

[11] LRZ only lends the necessary technical support.

Room	NSR0	NSR1
False floor height (cm)	80	80
Available area	Rows 2 and 3 north, 15 rack units per row	No restrictions
Max. area load (kN/m^2)	15	15
Max. point load (kN)	5	5
Air cooling capacity (kW)	400	50 (dehydration)
Cold water cooling capacity (kW)	2 x 300 (south/north)	360
Electrical power (kW)	800	800

For NSR0, the supplied floor plan indicates where SuperMUC-NG related hardware can be installed.

[Insert text here]

3.1.4 Conformity with electrical standards

LRZ provides power supply of 50 Hz alternating current, with 230 V single-phase and 400 V three-phase voltages, as commonly available in Germany. LRZ provides 175 power distribution boxes in the false floor for the electrification of the system, each containing two IEC 60309 32 Ampere five-pole, three-phase power connectors.

M 5: The required connectors, frequency or voltage transformers must be included in the tender in case that other electrical power connectors or other electrical power characteristics are needed.

The installation costs for such components is part of the contract and must not be charged separately.

Check here that this requirement will be fulfilled: **[ ]**

The *EC Directive on Electromagnetic Compatibility of Devices* is the guideline for all manufacturers, importers, and distributors of electric and electronic devices in the European Union as far as development, production and distribution of devices, systems and equipment are concerned. In addition, the *Low-Voltage Directive 2014/35/EU* and the *Act about Product Liability* must be considered. These regulations and the corresponding directives and standards are EU stipulations that have been adopted by German Law. By attaching a “CE marking” on a device, the manufacturer gives a legally binding declaration that the device conforms to all EMC relevant directives and standards of the EU and with the regulations and laws.

M 6: The system and its components must conform to German and European directives and laws at the time of delivery.

All necessary documentation must be submitted to LRZ before delivery of the system.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.1.5 Power supply

The usage of highly energy efficient cooling technologies and all other aspects leading to a high Data Center Infrastructure Efficiency (DCIE) is an important goal of the SuperMUC-NG procurement. In this spirit, aspects such as free cooling and a possible use of waste heat to heat the LRZ buildings as well as possible use of waste heat to operate adsorption chillers are important and will be honored respectively.

M 7: The expected AC power draw of the offered system in normal user mode and the expected AC power draw of the system running the HPL benchmark must be specified.

The appropriate fields in the tables in Section 4 of this document must be filled in.

I 7: The total electrical AC power requirement of Phase 1 of SuperMUC-NG under HPL load should not exceed 5500 kW.

This does not include the additional AC power needed for the cooling of the system and the compute room.[12]

Check here that this requirement will be fulfilled: **[ ]**

I 8: The total electrical AC power requirement of the combined Phases 1+2 of SuperMUC-NG under HPL load should not exceed 7150 kW.

This does not include the additional AC power needed for cooling of the system and the compute room.

Check here that this requirement will be fulfilled: **[ ]**

If any of the above two requirements cannot be fulfilled, the provisioning of additional electrical AC power must be discussed.

T 1: It is desired that redundantly powered critical components (e.g. management servers, storage servers, and network components) are supplied via connections to different power bars.

Check here that this requirement will be fulfilled: [ ]

[Insert text here]

3.1.6 Cooling

To reduce cost and to improve energy efficiency, a major goal of LRZ is to use free cooling to as large as possible extent for the cooling of its HPC infrastructure. Hence, two distinct water cooling loops called Loop 1 and Loop 2, operating at different temperature levels will be available for the cooling of the system. The cooling loops will use the following water temperatures:

[12] cf. “*Anschreiben SuperMUC-NG* "

Loop 1: T_{in}=18°C; T_{out}= 24°C; ΔT=6K;
Loop 2: 35°C ≤ T_{in} ≤ 45°C; ΔT ≥ 6K (LRZ can only guarantee that throughout the year T_{in} does not exceed 45°C);

Here T_{in} denotes the water inlet temperature to the compute racks and T_{out} denotes the preferred water temperatures at the outlet side of the racks. Throughout the year, the maximum cooling power of these loops is

Loop 1: 1.3 MW
Loop 2: 8.0 MW.

Due to its high inlet temperature, Loop 2 is operated **without any chillers** only using the free cooling capacities at the roof of the building. Loop 1 and Loop 2 are closed internal cooling loops, separated by heat exchangers (Loop 2) and mechanical chillers (Loop 1) from the glycol water cooling loops connected to the cooling towers at the roof of the building.

The water within these internal loops contains no glycol and only small amounts of additives such as corrosion inhibitors and biocides are to be expected according to VDI 2035[13].

For direct cooling of IT equipment, a total of 5 stainless steel heat exchangers with accumulated heat transfer capacity of up to 6 MW are in use. Each unit is equipped with pressure, volume flow and temperature control unit, redundant water pumps and very fine grained filters (50 µm) are installed in the HRR-room. The system setup of these so-called CooLManagers is shown in Figure 1. The Tichelmann Cooling Loops of SuperMUC Phase 1 and Phase 2 are shown in Figure 2.

M 8: In case the existing Tichelmann Cooling Loops cannot be reused, the Tenderer must deliver and install the warm water cooling supply lines for SuperMUC-NG Phase 1 and Phase 2.

Check here that this requirement will be fulfilled: **[ ]**

The operation points and maximum volume flows of the CoolManagers are listed below.

CoolManager A + B:

- Operating point: 115 m^3/h (pumping head 23.4 mWS)
- Pumping head for 126 m^3/h: approx. 22 m (2.2 bar)

CoolManager C + D:

- Operating point: 86.3 m^3/h (pumping head 18.5 mWS)
- Pumping head for 126 m^3/h: approx. 14 m (1.4 bar)

CoolManager E:

- Operating point: 143 m^3/h (pumping head 20.4 mWS)
- Pumping head for 180 m^3/h: approx. 14 m (1.4 bar)

CoolManager F:

- Operating point: 134 m^3/h (pumping head 21.4 mWS)
- Pumping head for 180 m^3/h: approx. 14 m (1.4 bar)

[13] See https://www.vdi.de/technik/fachthemen/bauen-und-gebaeudetechnik/fachbereiche/technische-gebaeudeausruestung/richtlinienarbeit/vdi-2035/

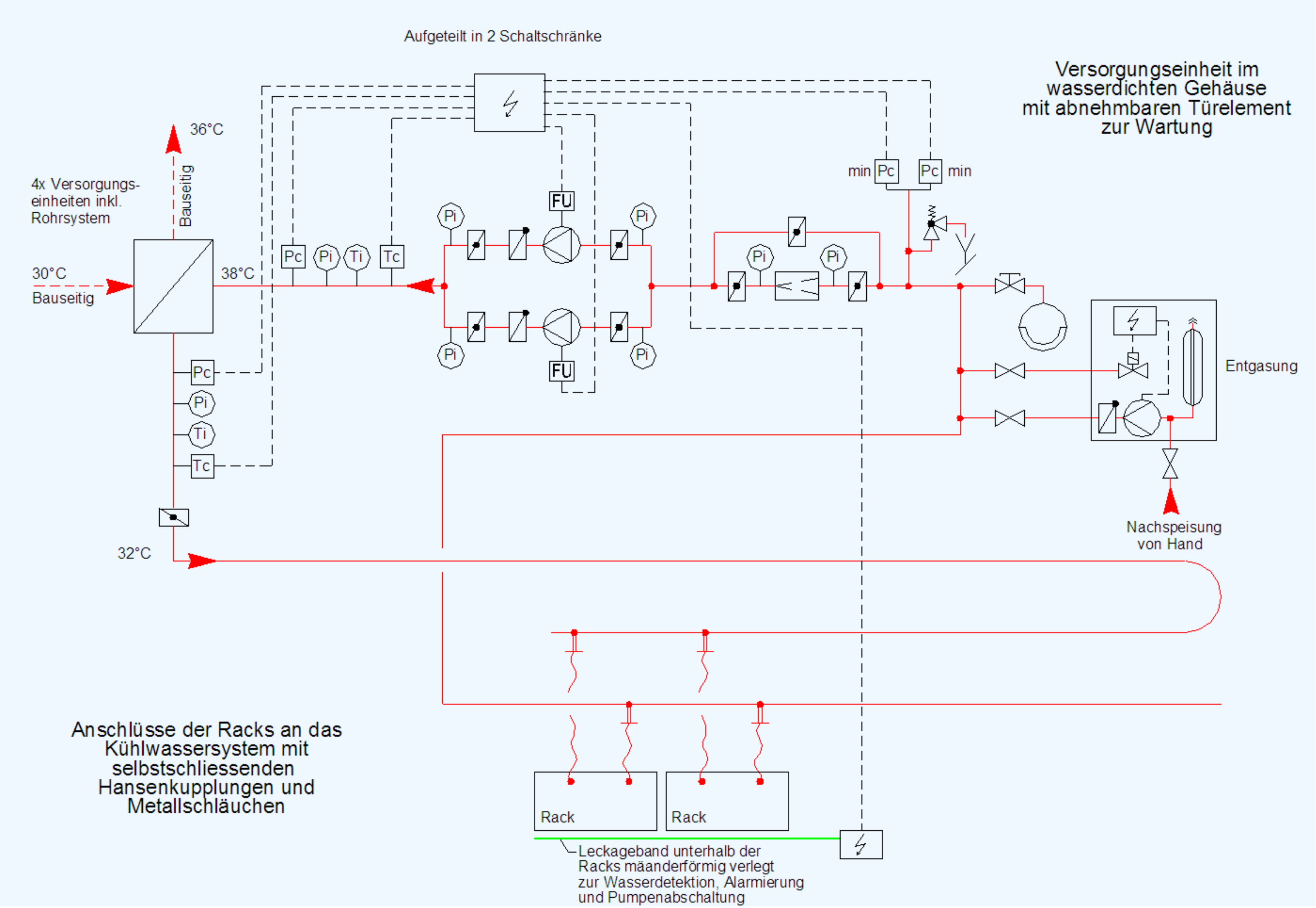
Aufgeteilt in 2 Schaltschränke
Versorgungseinheit im wasserdichten Gehäuse mit abnehmbaren Türelement zur Wartung
36°C
Bauseitig
4x Versorgungs-einheiten inkl. Rohrsystem
30°C
Bauseitig
38°C
Pc
Pi
Ti
Tc
FU
min Pc
Pc min
Entgasung
Nachspeisung von Hand
32°C
Anschlüsse der Racks an das Kühlwassersystem mit selbstschliessenden Hansenkupplungen und Metallschläuchen
Rack
Rack
Leckageband unterhalb der Racks mäanderförmig verlegt zur Wasserdetektion, Alarmierung und Pumpenabschaltung

Figure 1: CooLManager hydraulic system

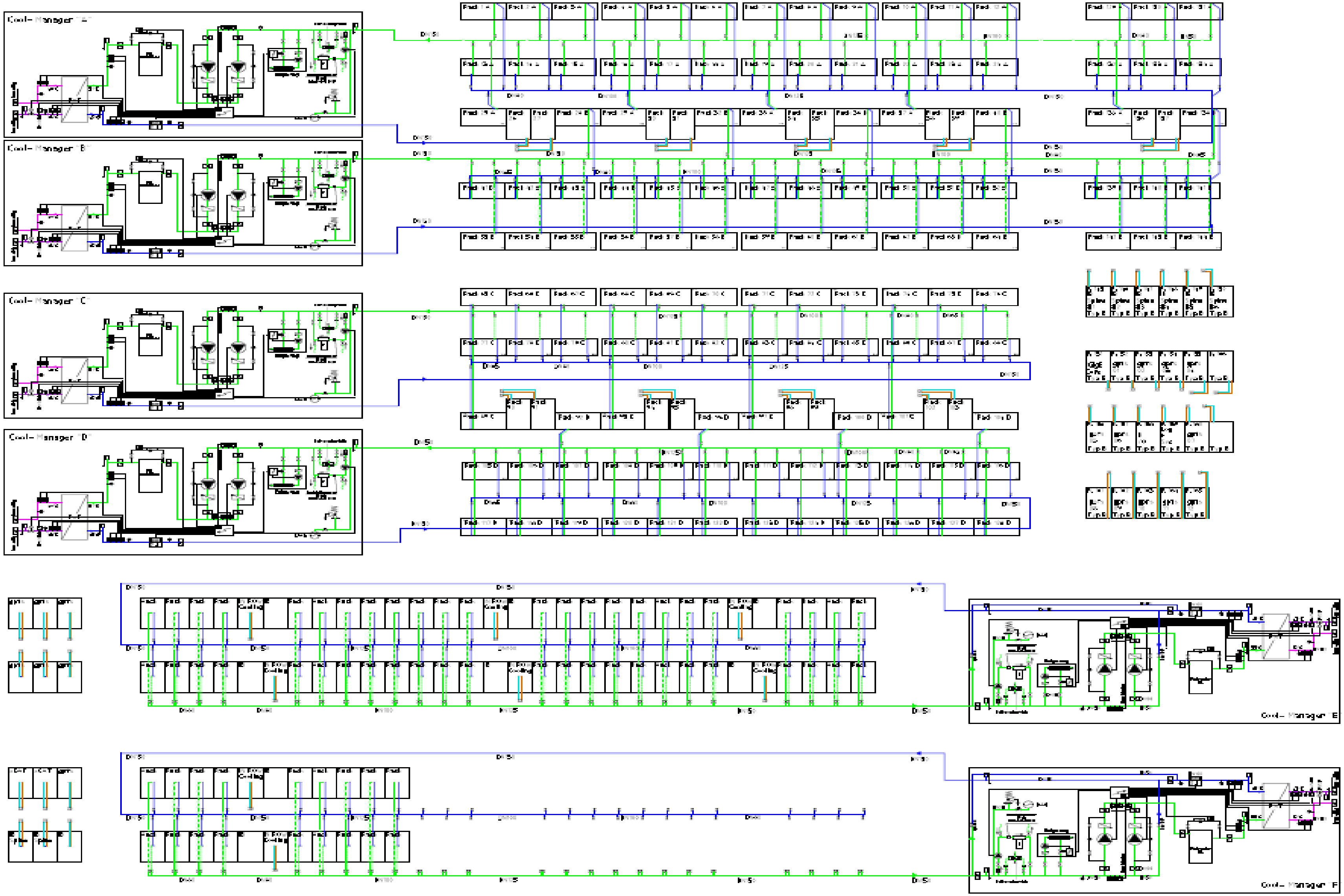

Figure 2: Tichelmann Cooling Loops of SuperMUC Phase 1 and Phase 2

LRZ operates a BACnet/IP building automation system by Johnson Controls to monitor its cooling infrastructure. In case a critical component of the cooling infrastructure fails, a 24/7 on-call technician can be notified by the building automation to take action.

M 8a: Note – new requirement:

Any cooling infrastructure that is going to be installed by the Tenderer and which is critical for operating SuperMUC-NG must be connected to LRZ's BACnet/IP network and integrated into the building automation system. The integration will be the duty of the Tenderer and must allow for monitoring the operational parameters and the status of this cooling infrastructure.

Check here that this requirement will be fulfilled: **[ ]**

M 9: The type of cooling system for all devices (air-cooling, water-cooling, etc.) and details about the cooling system (e.g. in- and outlet temperature, temperature variation tolerance, pressure, purity requirements, etc.) and the required environmental conditions (e.g. ambient temperature, humidity, dust free conditions, etc.) must be specified.

In addition, the appropriate fields in the tables in Section 4 must be filled in.

M 10: The heat emission of all devices of the offered system, broken down into air and water cooling, i.e. the maximum values and estimated values for permanent load, must be specified.

Also the heat emission released into each of the two separate water cooling loops “Loop 1” and “Loop 2” must be specified.

In addition, the appropriate fields in the tables in Section 4 must be filled in.

I 9: The heat emission released into the **room air** should not exceed 120 kW for Phase 1 and 200 kW for Phase 1 and Phase 2.

Pease note that the air handlers for room air conditioning are connected to Loop 1.

Check here that this requirement will be fulfilled: **[ ]**

I 10: At least 97 % of the waste heat of the compute node racks should be removed by means of direct water cooling of the components with cooling Loop 2 under the following environmental circumstances:

Loop 2: T_{in}=40°C
ambient room temperature: 25°C

Check here that this requirement will be fulfilled: [ ]

I 11: The total heat emission of SuperMUC-NG Phase 1 released into the water cooling loop called Loop1 (chilled water cooling loop) should not exceed 0.8 MW.

Check here that this requirement will be fulfilled: **[ ]**

I 12: The heat emission released into the water cooling loop called Loop 2 in normal user mode should not exceed 4.0 MW in Phase 1 (warm water cooling loop).

Check here that this requirement will be fulfilled: **[ ]**

If one of these requirements cannot be fulfilled, the provisioning of additional cooling capacity must be discussed.

T 2: Higher permissible water inlet temperatures T_{in} than 45 °C are preferred for the compute nodes.
If feasible, specify the maximum permissible inlet temperature.

Check here that this requirement will be fulfilled: **[ ]**

T 3: Almost zero heat emission of the total system into the room air is desirable.
If feasible, describe the technical means to achieve this.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.1.7 Documentation and Labelling

M 11: After installation, the final position of all components and the connections between components must be clearly documented and all components must be clearly and consistently labelled and named.

Check here that this requirement will be fulfilled: **[ ]**

I 13: An electronic database of all labels should be supplied. For hardware components such as compute nodes or network switches a bar-code associated with each label should be supplied and the electronic database should contain the alpha numeric keys of these bar codes.

Check here that this requirement will be fulfilled: [ ]

[Insert text here]

3.2 System Architecture and Compute Nodes

The overall system architecture, its components, and their lifetime as well as the RAS features of the system must be suitable for typical HPC applications and the intended use.

3.2.1 Overall System Architecture

M 12: The Tenderer must provide a concise description of the proposed SuperMUC-NG Phase 1 architecture, including all major system components, and a general description of the proposed SuperMUC-NG Phase 2 architecture.

The description of Phase 1 must include:

- An overall system architectural diagram or description showing all node types and their quantity, the interconnect, and connections to the I/O subsystem.
The diagram should also indicate the latencies and bandwidths of the data pathways between components.
- An architectural diagram of each node type showing the components of the node along with the relevant latencies and bandwidths.

I 14: The aggregate DDR-type main memory capacity of the Phase 1 system should be at least 500 TBytes.

Check here that this requirement will be fulfilled: **[ ]**

I 15: The general purpose nodes should have at least the same performance as Phase 2 of SuperMUC.

This should be demonstrated

by the HPL benchmark running on the general purpose nodes thereby having a performance of more than 2.8 PFlop/s (see Section 2.2.4 in the Decision Criteria and Benchmark Description SuperMUC-NG),

OR alternatively

by the SUSPENSE benchmark running on the general purpose nodes thereby having an aggregate performance of more than 0.33/sec (see Section 2.3.6 in the Decision Criteria and Benchmark Description SuperMUC-NG).

Check here that this requirement will be fulfilled: **[ ]**

I 16: The Tenderer should describe how the proposed system fits into their long-term product roadmap, particularly for Phase 2 of the system.

The Tenderer commits to informing LRZ in a timely manner about any substantial changes of the implementation of Phase 2 relative to the delivered description.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.2.2 General Purpose Compute Nodes

M 13: The Tenderer must provide the specifications of all offered compute node variants.

The following items must be covered:

- general description of the compute node technology
- type of processors and number of execution units of cores
- frequency, performance, bandwidths and latencies (including thermal dependencies, if applicable)
- multiprocessor node architecture (e.g. NUMA, ccNUMA, etc.)
- number of physical processor cores (physical and virtual)
- number of memory controllers, memory type and memory read/write bandwidth available per processor chip
- size and types of memory (optionally including High Bandwidth Memory,)
- maximum configurable size of memory
- size, line length and associativity of caches
- maximum number of floating point as well as general purpose operations per clock (separately for scalar and/or vector units, if applicable)
- number of registers (separately for floating point, integer and general purpose operations, and for scalar and/or vector operations)
- if vector units are available, describe the types of available vector operations
- intra-node connections (latencies, bandwidths, network type, topology)
- number and type of PCI-E Gen-X and/or other (proprietary) slots
- number and type of network and I/O links
- RAS features
- upgradability and expandability under the constraints of the proposed system architecture
- virtualization support

For the targeted operation of the system the following items are of specific interest:

- coherency mechanisms
- hardware performance counters
- temperature, AC and DC node power as well as electrical energy monitors [14]
- means for controlling node power draw and temperature
- means for controlling CPU and/or memory frequency
- hardware support for debugging and error detection

In addition, the appropriate fields in the table in Section 4 must be filled in.

I 17: Details and the modes of usage (e.g. usage as cache, as addressable memory) should be described for High Bandwidth Memory (HBM) in case nodes are equipped with this type of memory.

[14] Power measurement specifications are discussed in Section 3.6.1.

I 18: A conceptual description for the compute node and processor architecture of Phase 2 and a description of the cooling solution of compute nodes should be provided. The rationales for performance improvements should be supplied, particularly covering the items mentioned in Section 1.4.

The description of Phase 2 should include the technical details from above, but only conservative estimates or ranges for realization and not exact numbers need to be supplied.

[Insert text here]

3.2.3 Configuration of the Compute Nodes

The offered compute nodes should permit high compute performance but should also be balanced in terms of memory bandwidth. The memory must have an appropriate size and bandwidth in relation to the compute power, the main memory needed by the operating system (in the case of diskless nodes) and the memory buffers needed for message passing and I/O.

I 19: All compute nodes should consist of coherent shared memory nodes with at least 40 compute cores.

Check here that this requirement will be fulfilled: **[ ]**

I 20: At least 2 GByte usable by applications memory per MPI task for the **thin compute nodes** should be provided. In order to fulfill this requirement the Tenderer needs to account for the memory needed for other purposes such as OS, buffers etc. in the compute nodes.

Check here that this requirement will be fulfilled: **[ ]**

I 21: At least **128** nodes of SuperMUC-NG Phase 1 should be configured as (“fat”) **Large Shared Memory Compute Nodes** with at least 40 physical cores and a memory size of at least **768 GBytes**.

Check here that this requirement will be fulfilled: **[ ]**

I 22: The host processor architecture of the **fat compute nodes** should be a general purpose architecture[15].

Check here that this requirement will be fulfilled: **[ ]**

[15] The decision on whether to include accelerators in the fat nodes is left to the vendor. See also Section 3.2.5 below.

I 23: The Large Shared Memory Compute Nodes should be directly warm water cooled.

Check here that this requirement will be fulfilled: **[ ]**

T 4: Specific hardware support for innovative programming models is desired, such as support for transactional memory, support for fast thread synchronization, support for PGAS or support for communication, synchronization, and atomic operations.

If offered, a description should be given.

[Insert text here]

3.2.4 Optional: Many-Core Nodes

Additional stand-alone many-core compute nodes may be offered.

M 14: A concise description of the many-core nodes for Phase 1 must be given, if offered.

If applicable, the description must include the same items as for the general purpose compute nodes (see Section 3.2.2).

T 5: It is desired that the stand-alone many-core nodes contain at least 96 GBytes of memory and are equipped with a sizeable portion of high bandwidth memory.

[Insert text here]

3.2.5 Optional: Accelerated Nodes

Attached accelerators such as GPGPUs may be offered, integrated into suitable subsets of the general purpose and/or large memory compute nodes.

M 15: A concise description of the accelerator architecture for Phase 1 must be given, if offered.

Where applicable, the description must include the items as for the general purpose compute nodes (see Section 3.2.2). Additionally, the description must cover the following items:

- A general description of the accelerator technology
- Frequency, performance, bandwidths, and latencies between all subcomponents (intra-accelerator, accelerator-host, accelerator-accelerator, and accelerator-network)
- Details of the connection between accelerator and host (PCI, NVLINK, level of coherency, etc.)
- Number of offered accelerated nodes and number of accelerators in each node
- Means of performing communications between accelerators without unneeded copying of data, either within a node or across nodes
- ECC features
- Hardware sensors available for measuring accelerator temperatures, accelerator, DC power[16] consumption as well as performance -

M 16: The GPGPU-accelerated nodes must be equipped with at least 192 GBytes of DDR-based host memory on each node (i.e. not counting the high bandwidth memory on the GPU).

Check here that this requirement will be fulfilled: [ ]

[16] Power measurement specifications are discussed in Section 3.6.1.

I 24: The accelerated nodes should use the same host processors as the general purpose nodes. They should be integrated into the system using the same interconnect.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.2.6 Variability of the Compute Nodes

Reproducibility of key metrics from job to job or across different sets of nodes is important for reliable execution, accounting, and optimization of the workload. This refers not only to execution time but also to other items like electrical energy consumption or electrical power draw. Ideally, all processors have the same performance, electrical power, and thermal characteristics (including turbo mode).

Variation is defined as the absolute value of the difference of two measurements divided by their mean value. Verification is performed with a suitable subset of the benchmarks.

I 25: The Tenderer should describe its provisions for ensuring minimum variability of the compute nodes (e.g. quality sampling of processors, adjustment via BIOS or OS parameters, on-site calibration, etc.)

I 26: The Tenderer should describe the tradeoffs between runtime variation, energy and power variation, and means for controlling it.

I 27: For measurements that are performed under comparable conditions, the variation of **runtime** should not be more than 5% across different runs and across different sets of (comparable) compute nodes.

Check here that this requirement will be fulfilled: **[ ]**

I 28: For measurements that are performed under comparable conditions, the variation of **average electrical power draw** (=**electrical energy consumption divided by runtime**) should not be more than 7% across different runs and across different sets of (comparable) compute nodes.

The processor setup may be different from that of the previous runtime measurements.

Check here that this requirement will be fulfilled: **[ ]**

T 6: 3% average runtime variability is desired, even if this may be at (limited) expense of additional electrical power draw.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.2.7 Login and Service Nodes

Login nodes will be used for:

- User login and all types of general interactive user activity
- Submission of batch jobs
- Compilation
- User triggered archiving of results

Service nodes will be used for:

- Resource management and the batch queuing system
- Accounting data base(s)
- Power/Energy data base(s)
- System monitoring

The login nodes may differ from compute nodes but should contain the same processor architecture. The service nodes may differ from compute nodes and are permitted to contain a predecessor processor architecture. **Table 2** contains an overview of the requirements for both login and service nodes.

M 17: A standard Linux distribution must be supported for the login and service nodes.

Check here that this requirement will be fulfilled: **[ ]**

I 29: At least **4 login nodes** with at least **512 GB** main memory per node should be included in the offer for login and interactive work. These special nodes should include **local disks** for OS, scratch, swap, and paging.

The processor architectures used in the login nodes should be the same as that of the general purpose compute nodes.

Check here that this requirement will be fulfilled: **[ ]**

I 30: At least **4 additional login nodes** should have a high bandwidth connection to the LRZ backup and archive system. These nodes should be equipped with an additional dual-port 100 Gbit Ethernet network interface adapter and at least **512 GB** main memory.

Check here that this requirement will be fulfilled: **[ ]**

Beyond service nodes required for running basic system services, additional nodes capable of running the monitoring and database services required by the Tenderer and by LRZ (MySQL, persyst, Splunk, Icinga, etc.) will be needed.

I 31: At least **8 service nodes** with an aggregate main memory of **5 TByte** should be included in the offer for service purposes (see **Table 2**).

These special nodes should include redundant local disks (RAID) for OS, scratch, swap, and paging; they should be designed for continuous high I/O throughput (read and write of large numbers of files).

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

Type	**Number of units**	**Services which are run on these nodes**	**Access**	**Purpose**	**Type**	**Cores**	**RAM**	**Local Storage**
Login Nodes	4	Login	World-wide for all users.	Compile codes and submit jobs.	same as compute nodes	≥ 40	≥ 512 GByte	512 GByte SSD (for OS only)
	4	Login., Access to Archive	World-wide for all users.	Archive and retrieve data.	same as compute nodes, with fast connection to archive infrastructure	≥ 40	≥ 512 GByte	512 GByte SSD (for OS only)
Service Nodes	2	Accounting, Performance. and Power Databases	LRZ-internal for LRZ Staff.	Host databases for accounting data from the batch scheduling system, performance, and the DC power draw of the complete system. Configured as master and replica server.	Must be x86	≥ 32	≥1 TByte	16 x 1 TByte SSD RAID (Enterprise level SSD)
	2	Monitoring	LRZ-internal for System Admins	Host the web pages of the monitoring solution.	Must be x86	≥ 32	≥ 512 GByte	8 x 1 TByte SSD RAID (Enterprise level SSD)
	2	Batch System/Resource Management	LRZ-internal for System Admins	Host the central services for the resource management system and the OpenLDAP servers for user authentication.	TBD	≥ 40	≥ 512 GByte	8 x 1 TByte SSD RAID (Enterprise level SSD)
	2	System Management	LRZ-internal for System Admins	Host the central services for the system management solution.	TBD	≥ 32	≥ 512 GByte	8 x 1 TByte SSD RAID (Enterprise level SSD)

Table 2: Configuration of login and service nodes.

3.3 High-Performance Interconnect

3.3.1 General objectives

SuperMUC-NG must allow execution of jobs that use the full system. All data needed by a job must be accessible from any node. In particular, all parallel file systems must be accessible from the compute and login nodes. It should also be possible to access all parallel file systems from the service nodes. Apart from MPI messages and I/O transfers between nodes, it might also be necessary to route further communication related to the batch scheduling system through the high-performance interconnect.

A hierarchical interconnect may be provided where compute nodes are grouped into islands with fully non-blocking network topology. Depending on cost and technology, the number of network links between these islands can be reduced compared to a fully non-blocking network topology (pruned interconnect).

M 18: A concise description of the high-performance interconnect of Phase 1 must be provided.

The following items must be disclosed (at a minimum):

- topology
- technology
- material of cables (copper, optical, etc.)
- bandwidth and latency (between nodes and for all involved single components)
- support for RMA operations
- minimum and maximum time for a broadcast to reach all nodes
- limitations and resource usage (e.g. maximum number of compute nodes, message length, maximum number of outstanding messages, buffer sizes, etc.)
- scalability with respect to maximum number of compute nodes
- routing algorithms and whether dynamic rerouting is possible
- available performance and error counters
- sensors for AC as well as DC power draw and temperatures
- required management and administration hardware and software
- support for high-availability setups and means to improve reliability
- quality of service infrastructure (e.g. managing I/O bandwidth, separation of traffic)
- AC power draw

In addition, the appropriate fields in the tables in Section 4 must be filled in.

M 19: The high-performance interconnect must support efficient execution of applications that use the complete system.

Login and service nodes must be integrated into the high-performance interconnect.

The high performance file system (SCRATCH and WORK) must be integrated into the high-performance interconnect.

Check here that this requirement will be fulfilled: [ ]

I 32: The high-performance interconnect should support one- and two-sided communication as well as overlap of computation and communication.

Check here that this requirement will be fulfilled: [ ]

M 20: The high-performance interconnect must provide fault-detection and fault-isolation.

Check here that this requirement will be fulfilled: [ ]

I 33: The high-performance interconnect should support RDMA operations.

Check here that this requirement will be fulfilled: [ ]

I 34: Accelerators or many-core processors should be able to directly access the high-performance interconnect, if such devices or node types are supplied, respectively.

Please describe the interaction of these devices with the interconnect, if applicable.

Check here that this requirement will be fulfilled: **[ ]**

I 35: Apart from the native protocol, the interconnect should provide support for TCP/IP messaging.

Check here that this requirement will be fulfilled: [ ]

T 7: For the following desired features of the high-performance interconnect, please check the appropriate boxes and deliver a description (hardware and software) if applicable.

- Off-load message passing functionality to devices of the interconnect **[ ]**
- Support for PGAS languages such as CAF & UPC **[ ]**
- Support for (noncontiguous) gather-scatter operations **[ ]**
- [**additional features may be added** by the Tenderer and be checked] **[ ]**

T 8: It is desired that the interconnect balance, which is given by the aggregate theoretical link bandwidth (sum of all links, incoming plus outgoing) of a node divided by the aggregate theoretical memory bandwidth of a node (= maximum of DDR-type and HBM memory bandwidth), is larger than

1 : 40

Check here that this requirement will be fulfilled: **[ ]**

On SuperMUC-NG, it is intended to execute significantly larger jobs than on the current SuperMUC installation without incurring scaling limitations imposed by a blocking network.

T 9: It is desired that the size of a thin node island be at least 1024 nodes or has a peak performance of at least 1.5 PFlop/s.

Check here that this requirement will be fulfilled: [ ]

[Insert text here]

3.3.2 Interconnect within and between Islands

The bisection bandwidth (measured with the same number of nodes) between islands may be less than the bisection bandwidth within one island ("pruned interconnect").

M 21: Quantify the pruning factor between the islands.

If no pruning is used, insert the number "1".

Insert here: ______ **: 1** (intra:inter)

Details for the calculation must be provided.

T 10: It is desired that the MPI latency between two arbitrary nodes (running at least one MPI thread per node) within an island is less than 500 ns.

Check here that this requirement will be fulfilled: **[ ]**

T 11: It is desired that the MPI latency between two arbitrary nodes of different islands is less than 1000 ns.

Check here that this requirement will be fulfilled: **[ ]**

T 12: It is desired that the pruning factor of the interconnect bandwidth between islands is in the order of **4:1** (intra:inter).

Check here that this requirement will be fulfilled: [ ]

[Insert text here]

3.3.3 Interconnect between Phase 1 and Phase 2

Phase 1 and Phase 2 do not necessarily require a joint high performance interconnect. However, it should be assured that both Phases can access the filesystems HOME, PROJECT, WORK and SCRATCH with the committed I/O performance.

M 22: A conceptual description of the interconnect of Phase 2 and of the interconnect between Phase 1 and Phase 2 must be provided.

This description must cover the following technical aspects (at a minimum):

- Network technology and topology
- Performance characteristics (e.g. hardware, bandwidths, latencies, optional pruning factors, etc.)
- Capabilities such as dynamic routing

I 36: The access to the HOME, PROJECT, WORK, and SCRATCH file systems delivered with Phase 1 from both phases should be possible with the performance values committed for Phase 1.

Check here that this requirement will be fulfilled: **[ ]**

T 13: The interconnect between the two phases of SuperMUC-NG may be different from that within Phase 1 or Phase 2 respectively, but it is desired that it be interoperable.

If this is the case, running a parallel application across phase boundaries (e.g. HIGH PERFORMANCE LINPACK) should be possible in a transparent manner and with sufficient performance.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.3.4 Dynamic routing and congestion control

I 37: Dynamic network routing mechanisms such as error detection, congestion control and automatic re-routing of network packets in case of link failure or link congestion should be available.

If available, the technical solutions and characteristics of these mechanisms should be described.

Check here that this requirement will be fulfilled: **[ ]**

I 38: Monitoring software for the network traffic should be available. This software should display performance data, error counters, and current problems.

If available, the technical solutions and characteristics of these features should be described.

Check here that this requirement will be fulfilled: **[ ]**

T 14: It is desired to integrate the network monitoring with that of the other system components. If applicable, the solution should be described.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.3.5 Quality of Service of the High-Performance Interconnect

I 39: The high-performance interconnect should provide means to ensure “Quality of Service”, i.e. to separate and prioritize various classes of communication, such as e.g. I/O and MPI traffic. It should also be possible to mitigate the interference of various classes of communication.

If supplied, a description of the technical solution and characteristics as well as steps that can be taken to prevent mutual disturbance of different jobs should be provided. If the interconnect is also used for I/O, the Tenderer should also disclose the implications and describe how this could affect the performance of applications.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.3.6 Connection to external networks

LRZ will provide an Ethernet backbone infrastructure with 100 Gbit/s technology for the connection of SuperMUC-NG to the outside world. The entry points for the connection will be two central switches (not included in this procurement) positioned in the NSR0 and DAR1 rooms of the LRZ data centre, respectively.

We provide the following estimates for the cable lengths:

- between the middle of HRR to:
  - the switch in NSR0 is about 65 meters
  - the switch in DAR1 is about 50 meters
- between the middle of the wall between NSR0/1 to:
  - the switch in NSR0 is about 50 meters
  - the switch in DAR1 is about 40 meters

The exact cable lengths have to be determined during contract negotiations.

M 23: The connection technology to the LRZ network infrastructure must be described.

The appropriate fields for external networks listed in the table in Section 4 must be filled in.

M 24: Note – this requirement has been modified:
SuperMUC-NG Phase 1 must be connected to the LRZ Ethernet infrastructure with an aggregate Ethernet network link bandwidth of at least 400 Gbits per second.

All network components and cables necessary for the connection of the system to the LRZ Ethernet backbone must be included in the offer.

Check here that this requirement will be fulfilled: **[ ]**

I 39a Note – new requirement:

If Ethernet cable connections span different rooms (e.g. HRR to DAR1), the cabling should be implemented in a structured manner using a central patch panel in each room. In order to maximise investment protection and reusability, the inter-room cabling should be preferably implemented using a modular, reusable, plug-and-play data center cabling system, consisting of pre-configured, per-terminated MTP/MPO trunk cables with a variable number of fibres that connect into modular breakout modules, which can be mounted in rack mount enclosures. Fibre optics quality should be at least OM4. Optimised breakout modules, splitting for example two 24F MPO connections to six 12F MPO connections for usage for 40G or 100G connections (only 8 of 12 fibres required) are allowed.

Check here that this requirement will be fulfilled: [ ]

I 40: If link aggregation is used to fulfill the requested Ethernet network bandwidth, the solution should fully support the Link Aggregation Control Protocol (LACP, IEEE 802.3ad), using a multi-chassis LACP trunk.

Check here that this requirement will be fulfilled: **[ ]**

It is essential to provide users access to the login nodes of SuperMUC-NG based on a reliable Ethernet.

I 41: All login nodes of both phases of SuperMUC-NG should be connected to the LRZ Ethernet backbone network infrastructure using native Ethernet technology with a fail-safe active-active connection of at least 50 GBit/s for each node using a pair of dedicated Ethernet switches (which must also be included in the offer).

The offered switch architecture should be described.

Check here that this requirement will be fulfilled: **[ ]**

I 41a Note – new requirement:

The switch architecture offered for the previous requirement should support configuration of VLANs.

Check here that this requirement will be fulfilled: [ ]

Because of their cumbersome administration and prior history with stability and performance issues, host-based gateway systems, which are equipped with adapters for the high-performance interconnect on the one hand and the external Ethernet on the other, should be avoided for the purpose of connecting SuperMUC-NG to external file systems.

I 42: In order to connect compute and login nodes to externally hosted file systems, appropriate gateway or bridging devices should be supplied. The technical solution and its characteristics should be described.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.4 Configuration of Phase 2 and the combined Phases 1+2

The conceptual description of the Phase 2 compute node architecture is covered in Section 3.2.2.

3.4.1 Extent of inhomogeneity of Phase 1 and Phase 2

SuperMUC-NG may be inhomogeneous within the first installation phase, for example

- nodes with different processor types, numbers of cores, or numbers of sockets may be used,
- different node types with respect to accelerators or processor architecture may be used.

SuperMUC-NG may be also inhomogeneous across the two installation phases, for example

- processors of different clock frequencies may be used for the two phases, respectively,
- nodes with different processor types, numbers of cores, or numbers of sockets may be used.

I 43: The processor architectures used in both installation Phases of SuperMUC-NG should be similar e.g., should have a compatible instruction set.

Check here that this requirement will be fulfilled: **[ ]**

I 44: Breaks in the application programming paradigms needed for efficient usage of SuperMUC-NG Phase 2 should be avoided.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.4.2 Performance of Phase 2

At this point, it is not intended to fix the exact values of peak performance, memory bandwidth, interconnect, or individual benchmarks for Phase 2 or for the combined Phases 1 and 2. This keeps the opportunity to negotiate the exact configuration in advance of the installation of Phase 2.

It should be clearly understood by the Tenderer that the final system should be similarly balanced as Phase 1. Without specifying exact values now, all components (particularly the interconnect) should be configured in conformance with this aim.

The performance of Phase 2 in comparison to the performance of Phase 1 will be evaluated as an **improvement ratio (IR)**.

$$IR = \frac{Aggregate\ Performance\ (Phase\ 2)}{Aggregate\ Performance\ (Phase\ 1)}$$

Thus, the performance of the combined Phases 1 and 2 is

$$Aggregate\ Performance\ (Phase1 + Phase\ 2) = (1 + IR) * Performance\ (Phase1)$$

The Aggregate Performance of Phase 1 will be measured during the acceptance test for Phase 1. Details for the ascertainment of the improvement ratio are given in the "*Decision Criteria and Benchmark Description SuperMUC-NG*" document.

M 25: The improvement ratio for the Phase 2 as defined in the document *Decision Criteria and Benchmark Description SuperMUC-NG, Section 1.14.2* must be specified:

The performance of Phase 2 will be ______ times that of Phase 1.

[Insert text here]

3.5 Storage

3.5.1 Overview

With the emergence of "big" data science over the last few years, requirements of users on HPC storage has become more and more diverse. In the past, it was sufficient for users to have a huge scratch area, a small home area, and a long term archive. These data areas needed only to be accessible on the HPC system itself for as long as an HPC project was active. Today, projects often want to generate data on the HPC systems and then share, post-process, and/or retain that data for future use on other systems, even after the formal end of an HPC project.

In order to support these new usage patterns, LRZ has recently introduced a **centralized, site-wide Data Science Storage (DSS)**; this is based on IBMs Spectrum Scale HPC Storage Cluster. DSS is currently only integrated into LRZ's HPC systems via the login nodes. However, with SuperMUC-NG, LRZ intends to extend the DSS Cluster and take its HPC integration to the next level.

In addition to the above shared storage resources, a system-wide scratch area will be needed which provides high and scalable throughput performance.

In the following, a high-level overview of the storage/file systems to be procured for SuperMUC-NG is provided:

- Data Science Storage systems

  The **HOME** file system will be used for the user home directories, and for supplying a system-wide software repository. The performance requirements are high IOPS and meta-data rate but moderate bandwidth. Data shall be protected by asynchronous replication to a secondary system and regular backups to the IBM Spectrum Protect based Backup system of LRZ.

  The **PROJECT** file system will be used for datasets, which need to be shared among other LRZ HPC systems, external parties, or retained on disk for analysis after expiration of the project on SuperMUC-NG. The performance requirements are high meta-data rate and high bandwidth. Data shall be protectable by regular backups to the IBM Spectrum Protect based Backup system of LRZ.

- High performance (parallel) file system(s)

  These consist of a file system or file system area called **SCRATCH** with no quotas but automated high watermark deletion and a file system or file system area called **WORK** with a per-project quota and no automated deletion. These file systems or storage areas will be used for "work-in-progress" data sets. The performance requirements are a moderate meta-data rate and a very high I/O bandwidth. Data will not be protected by regular backups or replication. However, users must be able to transfer data from this storage area into the Data Science Archive (which is not part of this procurement) or to the outside world.

The following table provides an overview of the intended file systems for Phase 1 and their characteristics; the number of "+" entries for each item indicates the relative performance requirement:

File System	Usable Capacity	Bandwidth	IOPS Rate	Accessibility	Protection
HOME	≥ 256 TiByte	+	+++	Site-wide	Snapshots, Replication, Backup
PROJECT	≥ 10 PiByte	++	++	Site-wide	Backup
SCRATCH WORK	≥ 50 PiByte	+++	++	System-wide	Explicit copy to archive or tape

3.5.2 Common Requirements

3.5.2.1 General

M 26: The offered storage and file system solutions must be described.

In addition, the appropriate fields in the table in Section 4 must be filled in.

The description must contain the following information (at a minimum):

- Type of storage devices
- Type of file systems
- Relevant features, such as hardware specifications, reliability, management software, interfaces, and functionalities for protection against data loss (RAID, redundancy and check summing, replication, snapshot features, etc.)
- AC power draw
- Hardware sensors available for measuring AC and DC power draw and temperatures
- Performance implications of rebuild operations as seen from file system level
- Upgrade / extension path and process from the capacity and performance requirements of Phase 1 to Phase 2
- Support structure: Who is responsible for handling first, second, and third level support for which component? Who is responsible for hardware and software enhancements of the proposed storage solutions?
- Handling of seamless degradation, recovery from failures, and impact of common failure scenarios.
- Failure domains

I 45: The design of the offered storage and file system solutions should follow a scalable building block approach.

The approach should be described.

Check here that this requirement will be fulfilled: **[ ]**

Because of privacy protection policy, LRZ normally does not permit to return defective drives to the storage vendor if the data on them is not encrypted.

I 46: All storage systems should be offered with a support contract which implements one of the following alternatives:

- keep defective drives at LRZ
- usage of self-encrypting drives
- contractual assurance that data on defective or retired drives will be reliably deleted.

Check here that one of the above requirement will be fulfilled: **[ ]**

I 47: All file systems should support POSIX access control lists (ACLs).

Check here that this requirement will be fulfilled: **[ ]**

T 15: It is desired that the Tenderer favors homogeneous **hardware** building blocks to implement both the DSS (HOME, PROJECT) and the High Performance File Systems (SCRATCH, WORK).

Check here that this requirement will be fulfilled: **[ ]**

It is LRZ’s experience that separate UPS equipment delivered as part of storage hardware does not cooperate well with the centre’s infrastructure.

T 16: It is desired that the used storage hardware does not rely on **external** high voltage battery backup units (BBUs).

Check here that this requirement will be fulfilled: **[ ]**

All electrically active components separately equipped with external high voltage BBUs need to be registered with the local campus fire brigade.

[Insert text here]

3.5.2.2 Reliability and Data Integrity

M 27: All active components in the storage systems must be redundant without any single point of failure.

Any single component failure must be automatically tolerated in a way that does not cause I/O errors or data loss on the accessing nodes or running applications, respectively.

Check here that this requirement will be fulfilled: **[ ]**

I 48: Any hardware failure in the storage systems and severe problems in the file systems should be automatically reported to the monitoring system.

Check here that this requirement will be fulfilled: **[ ]**

I 49: All active components in the storage systems should be hot swappable.

Check here that this requirement will be fulfilled: **[ ]**

T 17: It is desired that the storage systems are subdivided into different failure domains, where single-component failures in different failure domains do not cause I/O errors or data loss.

Check here that this requirement will be fulfilled: **[ ]**

M 28: All persistent storage (storage devices which are designed to store data permanently, e.g. disks or SSDs) in the storage systems must be either mirrored (e.g. RAID 1 or comparable mechanism) or must be protected against at least two simultaneous failures (e.g. using RAID 6 or comparable mechanisms).

Check here that this requirement will be fulfilled: **[ ]**

T 18: A “declustered” RAID approach or equivalent mechanism is desired to keep disk rebuild times and performance impact of rebuilds low.

Check here that this requirement will be fulfilled: **[ ]**

I 50: Automatic repair (recreation of redundancy) of failed persistent storage devices should happen without manual intervention (e.g. using hot-spares or spare capacity).

Check here that this requirement will be fulfilled: **[ ]**

M 29: The storage and file systems must be designed to tolerate power failures without data loss.

Check here that this requirement will be fulfilled: **[ ]**

M 30: All file systems must support a mechanism which allows a fast restart after system crashes or power failures. (e.g. journaling of meta-data and/or user data).

These mechanisms must be described.

Check here that this requirement will be fulfilled: **[ ]**

I 51: The storage systems should support storage device scrubbing (or alternative methods) which detects and corrects defective blocks.

Any abnormal events must be reported.

Check here that this requirement will be fulfilled: **[ ]**

I 52: Means to ensure end-to-end data integrity (from client to disk) should be available for the file systems. A segmented (e.g. client-to-server and server-to-disk) implementation is allowed, if it covers the whole path. Detected problems should be reported. Describe in detail how data integrity is checked from the client to the disk.

Check here that this requirement will be fulfilled: **[ ]**

I 53: File system consistency checking programs for all file systems used should be available. The check of a file system filled to at most 90% should not take more than 24 hours. Please describe the possible procedures in detail as well as performance of the file system checking mechanism in case a check is necessary.

Check here that this requirement will be fulfilled: **[ ]**

T 19: It is desired that a fully functional IBM Spectrum Protect (formerly Tivoli Storage Manager) client is available for the OS release used on the login nodes, service and management nodes of SuperMUC-NG.
The license costs for IBM Spectrum Protect are already covered by LRZ campus licenses.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.2.3 File System Access

I 54: All file systems should be accessible from all compute, login, service, and management nodes of SuperMUC-NG.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.3 Data Science Storage (DSS) style systems

LRZ runs a central HPC Storage Cluster based on **IBM Spectrum Scale** for sharing large scientific data within the LRZ HPC ecosystem (SuperMUC, Linux Cluster, Visualization Systems, login cloud, etc.) and via several gateway services (GridFTP, Globus Sharing, or even user driven VMs). The necessary service components are Spectrum Scale Cross Cluster Relationships and Spectrum Scale Cluster Export Services. The service as a whole is called **Data Science Storage (DSS**).

As described before, LRZ wants to implement the HOME and PROJECT file systems as an extension to its existing DSS Storage Cluster.

3.5.3.1 Common DSS Requirements

M 31: The DSS (HOME, PROJECT) must be accessible independent of the status of the compute nodes of SuperMUC-NG and of its high performance file systems.

Check here that this requirement will be fulfilled: **[ ]**

I 55: The HOME and PROJECT file systems should be implemented as an extension of LRZ’s existing centralized Spectrum Scale based DSS Storage Cluster. However, both file systems should be implemented as additional file systems.

Check here that this requirement will be fulfilled: [ ]

I 56: All DSS file systems should use the allocation method “scatter”.

Check here that this requirement will be fulfilled: **[ ]**

Currently LRZ holds the following Spectrum Scale Licenses:

- 512 Socket Licenses for Spectrum Scale Server Standard Edition
- 25576 Socket Licenses for Spectrum Scale Client Standard Edition

All licenses were purchased from IBM Germany. The Tenderer might be able to reuse these licenses once the current SuperMUC system is completely decommissioned.

M 32: The costs for the Spectrum Scale support during the targeted lifetime of SuperMUC-NG and the costs for potentially needed additional licenses or license trade-ups must be included in the offer.

Check here that this requirement will be fulfilled: **[ ]**

The current licenses for Spectrum Scale held by LRZ were acquired via IBM Germany. For SuperMUC-NG organizational difficulties resulting from involvement of different vendors should be avoided.

I 57: The Tenderer should describe how the first and second level support for Spectrum Scale will be delivered, particularly in cases where Spectrum Scale licenses delivered from multiple vendors are involved.

I 58: If the offered solution for DSS is based on an appliance approach (which provides an integrated stack of hardware, operating system and Spectrum Scale), a description should be provided how the release cycle of the integrated software stack relates to the new agile release cycles of Spectrum Scale. Describe in particular your release strategy for Spectrum Scale fix-packs, new minor versions and new major versions. Describe the mean delay time in weeks between the official availability of new Spectrum Scale releases and their deployment on the appliance solution. State whether there are any committable “not to exceed” delays.

[Insert text here]

3.5.3.2 DSS Storage Network Architecture

The Data Science Storage Cluster currently uses a 40 GigE Ethernet Backbone Network consisting of two 120 Port 40 GigE Switches located in DAR1 provided by LRZ[17].

M 33: The extension of the DSS Cluster must be connected to the DSS Ethernet Backbone Network.

Check here that this requirement will be fulfilled: **[ ]**

[17] It is not yet clear whether this can be upgraded to more recent technology in the time frame of this procurement. If so, part of the backbone will also be located in NSR0. Details should be attended to during the contract negotiations.

M 34: A reasonably sized gateway between the DSS Cluster, the 40 GigE Backbone Network and the High Performance Interconnect of SuperMUC-NG must be provided

or

the Spectrum Scale servers, which implement HOME and PROJECT, must be connected to the SuperMUC-NG High Performance Interconnect in addition to the connection to the DSS Backbone.

The second solution is preferred. The Tenderer must indicate which alternative is chosen.

Check here that this requirement will be fulfilled: **[ ]**

T 20: If the Spectrum Scale servers can be directly connected to the SuperMUC-NG high performance interconnect, it is desired that the connection between the servers and the high performance interconnect is sized suitable to allow the full bandwidth of HOME and PROJECT to be consumed by SuperMUC-NG.

In addition to this, it is desired that each Spectrum Scale server will be connected with two 40 GE links to the DSS Backbone Network.

Check here that this requirement will be fulfilled: **[ ]**

T 21: If the Spectrum Scale servers can be directly connected to the SuperMUC-NG high-performance interconnect, it is desired to use the Spectrum Scale Verbs RDMA interface for data transfer between the servers and SuperMUC-NG.

Check here that this requirement will be fulfilled: **[ ]**

T 22: If SuperMUC-NGs high-performance interconnect is Infiniband-based, it is desired that the Spectrum Scale Servers implementing HOME and PROJECT are connected to their own scalable and redundant Infiniband fabric which is then connected to the SuperMUC-NG Infiniband fabric via Infiniband routers.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.3.3 DSS Reliability and Data Integrity

For the DSS, data safety is extremely important. Therefore, data in DSS HOME should be replicated asynchronously.

The replication system will be physically placed in a computer room other than the primary system. However, the primary and secondary system will be connected to the same hardware redundant 40 GE Network Backbone. The replication target may either be implemented as a single system or two separate systems.

M 35: In addition to a traditional tape backup, HOME must be protected by **asynchronous replication** to a secondary system which is also part of the tender.

Check here that this requirement will be fulfilled: **[ ]**

T 23: It is desired to generate the backups from the replication using IBM Spectrum Protect.

Check here that this requirement will be fulfilled: **[ ]**

Loss of meta-data implies that the integrity of part of or the complete file system may be compromised, leading to effective data loss. As a best practice for avoiding this situation, the following requirement is imposed.

M 36: File system meta-data for HOME and PROJECT has to be protected by either 2-way RAID 1 mirrors in combination with a file system meta-data replication factor of two, or by implementing a 4-way mirror solution using a file system meta-data replication factor of one.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.3.4 DSS File System Export

I 59: In addition to the native Spectrum Scale protocol, the primary HOME and PROJECT file systems should be exported also via NFS and CIFS using a reasonably sized Spectrum Scale CES (Cluster Export Services) Cluster.
For sizing the CES nodes, expect that the maximum number of concurrent clients will be 500.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.3.5 DSS Performance

I 60: For NFS or CIFS exports of PROJECT via CES, the client read and write bandwidth of the CES Cluster should be at least

20 GiByte/s via NFS
4 GiByte/s via CIFS

using the IOZONE benchmark as defined in the *Decision and Benchmark Description SuperMUC-NG*.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.4 DSS HOME requirements

3.5.4.1 HOME implementation details

I 61: Dedicated storage solutions should be used for the primary and secondary HOME file system storage. This means that the storage hardware of the HOME file system will not be used by other DSS file systems.

Check here that this requirement will be fulfilled: **[ ]**

I 62: The primary HOME file system should be based on a tiered approach, using a combination of disk and SSD storage.

The SSD tier should be at least 20% of the overall capacity.

Check here that this requirement will be fulfilled: **[ ]**

I 63: The meta-data of the primary HOME file system should be located on SSDs. The meta-data of the secondary HOME file system can be located on disk. There must be enough meta-data capacity for storing at least 1,1 * 10^9 4K inodes on both the primary and secondary file systems.

Check here that this requirement will be fulfilled: **[ ]**

I 64: The Spectrum Scale block size for the primary HOME file system should be optimized with respect to the intended usage.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.4.2 HOME Capacity and Performance Requirements

I 65: The total usable size (as reported by the df command) of the **primary** HOME should be at least:

- 256 TiByte

The total usable size (as reported by the df command) of the **secondary** HOME should be at least:

- 256 TiByte

Check here that this requirement will be fulfilled: **[ ]**

M 37: The sequential read and write bandwidths for primary HOME using the IOR benchmark as defined in the document “*Decision Criteria and Benchmark Description SuperMUC-NG”* must be specified.

Provide performance values for the following workloads.

- Sequential Read/Write on the SSD tier (benchmark Test Type 1)
- Sequential Read/Write on the HDD tier (benchmark Test Type 1)

Data must be provided in the answer delivered for the “*Decision Criteria and Benchmark Description SuperMUC-NG”*.

M 38: The random I/O performance for the SSD tier of primary HOME using the FIO benchmark as defined in the document “*Decision Criteria and Benchmark Description SuperMUC-NG”* must be specified.

Provide performance values for the following workloads

- 4K Read/Write IOPS to the SSD Tier
- 4K Read/Write IOPS to the HDD Tier

Data must be provided in the answer delivered for the “*Decision Criteria and Benchmark Description SuperMUC-NG”.*

I 66: The replication performance between the primary and the secondary HOME should be at least:

5 GiByte/s

Check here that this requirement will be fulfilled: **[ ]**

I 67: Either reasonably sized NSD servers to run the Spectrum Scale mmbackup command on top of them or a reasonable number of mmbackup worker nodes should be provided to run both backup and restore of HOME with the following targeted performance:

2 GiByte/s Backup
2 GiByte/s Restore

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.6 DSS PROJECT File System Requirements

3.5.6.1 PROJECT Implementation Details

I 68: The **meta-data** of the PROJECT file system should be placed on SSDs. There should be enough meta-data capacity to store at least 2,7 * 10^9 4K inodes.

Check here that this requirement will be fulfilled: **[ ]**

I 69: The **meta-data** area of the PROJECT file system should be extendable. Please describe how this can be done and what would be the costs for doubling the available meta data capacity (including 5 years of support)

Check here that this requirement will be fulfilled: **[ ]**

I 70: The Spectrum Scale block size of the PROJECT file system for running the benchmarks should be 16 MByte.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.6.2 PROJECT File System Capacity and Performance Requirements

M 39: The total usable size of the PROJECT file system must be at least

10 PiByte

Check here that this requirement will be fulfilled: **[ ]**

M 40: The client read and write bandwidth of PROJECT file system using the IOR benchmark as defined in the document "*Decision Criteria and Benchmark Description SuperMUC-NG*" must be at least:

50 GiByte/s (Test Type 1)

Check here that this requirement will be fulfilled: **[ ]**

I 71: Either reasonably sized NSD servers to run the Spectrum Scale command on top of them or a reasonable number of mmbackup worker nodes should be provided to run both backup and restore of PROJECT with the following targeted performance:

10 GiByte/s Backup
10 GiByte/s Restore

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.7 High Performance Parallel File Systems

3.5.7.1 General

I 72: The offered high performance file system solution should consist of two separate file systems or file system areas.

- SCRATCH implements no quotas but automated high watermark deletion,
- WORK implements quota on a per-project and possibly per-user basis without any automated deletion.

Check here that this requirement will be fulfilled: [ ]

M 41: The scalability of the offered high performance file system solution(s) must be described in detail for each file system.

The following items must be included:

- maximum file system size
- maximum number of files and directories
- maximum number of files and directories per directory
- maximum file size
- maximum achievable aggregate I/O bandwidth
- maximum achievable meta-data operations per second (e.g. file creates per second)
- maximum number of client nodes which can be connected to one file system

M 42: It must be possible for LRZ to define the partitioning with respect to capacity or performance of each file system or file system area at installation time.

Check here that this requirement will be fulfilled: [ ]

M 43: The costs for the software support of any component of the high performance file system as well as its licences must be included in the offer.

Check here that this requirement will be fulfilled: **[ ]**

I 73: The maximum possible size of a **single file** in each high performance file system should be at least equal to the aggregate memory of the compute nodes. Please describe the potential side effects of a configuration which allows this.

Check here that this requirement will be fulfilled: **[ ]**

I 74: The mechanisms for parallel I/O on the high performance file systems should be described, especially for the use cases:

- multi-process access of small files
- multi-process access of large files
- single-process access of large files

I 75: The caching mechanisms of the high performance file systems for data and meta-data on clients and servers should be described, especially:

- How the file system caches interact with OS caches
- How to use or bypass the available cache tiers (if applicable)
- Any impacts of caches on I/O performance

I 76: The high performance file systems should support per-user, per-group, and per-directory-subtree quotas with respect to data size and the number of files. A reporting facility should be available.

Check for the availability:	size	files
- per POSIX-group quota	**[ ]**	**[ ]**
- per user quota	**[ ]**	**[ ]**
- per directory-subtree quota	**[ ]**	**[ ]**

T 24: It is desired that the SCRATCH file system provides efficient mechanisms to implement high watermark deletion. (e.g. parallel file system traversals). If such mechanisms are available, please describe them.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.7.2 Implementation Details

I 77: The meta-data of the Parallel High Performance File Systems should be placed on SSDs. There should be sufficient meta-data capacity to store $5.4 * 10^9$ files.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.7.3 Capacity and Performance Requirements

M 44: The total usable size of the High Performance File Systems of Phase 1 must be at least: 50 PiByte.

Check here that this requirement will be fulfilled: **[ ]**

M 45: The aggregate client read and write bandwidth of the high performance file systems using the IOR benchmark as defined in the document "*Decision Criteria and Benchmark Description SuperMUC-NG*" must be at least:

500 GiByte/s (Test Type 2: One file per MPI task, min{read, write})

Check here that this requirement will be fulfilled: **[ ]**

M 46: The aggregate client read and write bandwidth of the high performance file systems using the IOR benchmark

- MPIIO Shared File, one writer per MPI task, strided pattern (benchmark Test Type 3)
- HDF5 one file per node (benchmark Test Type 4)

as defined in the document *"Decision Criteria and Benchmark Description SuperMUC-NG"* must be specified.

Data must be provided in the answer delivered for the "*Decision Criteria and Benchmark Description SuperMUC-NG".*

I 78: The offered high performance file system solution should implement a transparent, intelligent, and cost effective mechanism to accelerate latency-bound IO patterns (e.g. small file IO). Describe the offered solution(s) in sufficient detail to enable assessment of the mechanisms.

Check here that this requirement will be fulfilled: **[ ]**

M 47: The meta-data performance (using production environment settings and conditions) by of the Bonnie++ MPI benchmark as described in the document "*Decision Criteria and Benchmark Description SuperMUC-NG*" must be specified.

Data must be provided in the answers for the "Decision Criteria and Benchmark Description SuperMUC-NG".

Fair sharing of I/O resources can help to balance bandwidth among different applications running simultaneously. In case that multiple applications are concurrently performing I/O to the file system, the aggregated I/O bandwidth, which can be utilized by a single application, has to be limited to a certain fraction of the total aggregated I/O bandwidth of the file system to avoid resource hogging. This limit should be a configurable parameter, which can be set by the system administrators. However, this requirement should not limit the usable aggregated I/O bandwidth for a single application in the case of non-concurrent I/O. Furthermore, the overhead imposed by such a feature should not heavily impact the overall performance of the file system.

T 25: A fair-share feature for use of I/O resources on the high performance file system is desired.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.5.8 Location of Storage Systems

The Data Science Storage systems will be installed in the computer room NSR0 of the LRZ (see also section 3.1.3).

M 48: For the DSS systems, the following must be assured:

- The storage racks must be installable in the NSR0 computer room, given its specifications. .
- The total power draw must be specified: ______ **kW**

Check here that this requirement will be fulfilled: **[ ]**

The SCRATCH and WORK storage systems can be installed in the HRR, NSR0 or NSR1 computer room (see again section 3.1.3), however there is the strong preference that it be placed in NSR0.

M 49: Note – this requirement has been clarified:
For all offered file systems, an installation concept must be delivered. It must be ensured that

- all infrastructural requirements on the installation location are specified (cooling infrastructure, floor space, weight and point load, number and type of electrical connectors)
- the offer includes all components (cabling, switches) needed for integration into the compute part of the system and (for DSS/HOME components) to the LRZ Ethernet backbone (cf. section 3.3.6)

- The total power draw must be specified: ______ **kW**

Check here that this requirement will be fulfilled: **[ ]**

I78a Note - new requirement:

The storage for the high performance parallel file systems should be installed in the NSR0 computer room, given its specifications.

The total power draw for this storage should be specified: ______ **kW**

Check here that this requirement will be fulfilled: [ ]

I78b Note - new requirement

If ethernet connections span different rooms, the cabling to the storage should be implemented in a structured manner, i.e. using a modular cabling system. See also I 39a (Section 3.3.6) for further technical details.

Check here that this requirement will be fulfilled: [ ]

[Insert text here]

3.5.9 Phase 2 Storage

For SuperMUC-NG Phase 2 it it will be necessary to extend capacity and performance of all existing file systems.

T 26: Note: this Target requirement is now mandatory

M49a:

The capacity and performance of the Phase 1 file systems must be extended in proportion to the additional compute power, as determined by the committed Improvement Ratio (section 3.4.2).

A concise draft for this extension must be delivered that describes it both quantitatively (additional capacities and bandwidths) and qualitatively (implementation strategy).

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.6 Technical Infrastructure for Energy Efficient Operation

3.6.1 Energy and Power Measurements

LRZ strives to reduce operational costs of its HPC operations by improving the energy efficiency of all IT and infrastructure components at all levels – from the compute node and compute rack level to power distribution and cooling infrastructures. Reliable measurements of each component’s energy consumption and AC as well as DC power are needed for realizing this goal. Table 3 contains the components to be measured along with their respective measurement precision and frequency.

SuperMUC-NG Hardware Components	Measurement accuracy AC / DC	Measurement frequency AC / DC
Compute Nodes	≤ 3% / ≤ 5%	0.0167 Hz / 10 Hz
Service, Cloud and Management Nodes	≤ 3% / ≤ 5%	0.0167 Hz / 10 Hz
Storage	≤ 3 % /	0.0167 Hz /
High Performance Network	≤ 3 % /	0.0167 Hz /
Ethernet Switches	≤ 3 % /	0.0167 Hz /

Table 3: Desired AC and DC measurement accuracies and measurement frequencies

M 50: The sensors for AC and DC power draw and energy consumption must be described, including

- List of components that are measured by each individual sensor
- Technical limitations such as error variation at different loads and temperatures, readout frequencies, etc.

M 51: Online measurements of the direct current (DC) **energy** consumption of individual compute nodes must be provided by means of monotonically increasing counters.

The accuracy and measurement frequency must be specified.

Check here that this requirement will be fulfilled: [ ]

I 79: The interval between counter overflows should be sufficiently long (> 1 week) and overflows should be detectable. The counters may be reset during reboot.

Check here that this requirement will be fulfilled: [ ]

I 80: Tools for reading out the node level energy and power counters should deliver energy values in units of Joules and power values in units of Watts. No additional calibration or transformation of values should be needed.

Check here that this requirement will be fulfilled: [ ]

I 81: All measurements at the node level should be accessible in-band by means of the Linux sysfs (or some other well-defined) user space interface and out-of-band over the service network via standardized and open protocols such as Redfish, PowerAPI, IPMI, or SNMP.

In-band and out-of-band readings should deliver consistent and accurate values.

Source code for the access API and a description of the protocol for the in-band measurements should be provided.

Check here that this requirement will be fulfilled: [ ]

I 82: In-band measurements of power and/or energy consumption of sub-components within a node should be available.

The Tenderer should deliver a list of the measurable sub-components that should include at least individual measurements of CPU, accelerator components if present, memory (including HBM), and network devices.

Check here that this requirement will be fulfilled: [ ]

M 52: Alternating Current (AC) power measurements devices must be provided down to the conversion between AC and DC.

Multiple components within a single rack may be measured by a single sensor, but different types of components must not be mixed (e.g. separate measurements for compute nodes and network switches are required).

Check here that this requirement will be fulfilled: [ ]

I 83: The number of components attached to one AC power meter should be the same throughout the system for each specific type of component.

Check here that this requirement will be fulfilled: [ ]

I 84: The readout frequency of AC energy and power measurements should be at least 0.0167 Hz.

The resolution of energy and power measurements should be at least 1 J and 1W, respectively. The error of energy and power measurements should be below 3%.

Check here that this requirement will be fulfilled: [ ]

T 27: It is desired that the readout frequencies of node level DC energy and power measurements are at least 10 Hz via all interfaces, with an error less than 5% for energy and power measurements.

Check here that this requirement will be fulfilled: [ ]

[Insert text here]

3.6.2 Reuse of Waste Heat

Reuse of system waste heat for the cooling of air or chilled water cooled system components will be positively evaluated.

T 28: Reuse of system waste heat to drive adsorption chillers is desired. If offered, the technical description of the implementation should be delivered and the cooling power that can be extracted should be specified.

[Insert text here]

3.7 Reliability and Fault Tolerance

SuperMUC-NG should be designed for high Reliability, Availability, and Serviceability (RAS) and should contain features for fault tolerance and resilience.

3.7.1 Mean time to Failure and Lifetime

The failure rates of individual components may be higher as long as the overall system adapts, and any impacted jobs or services are restarted without user or administrator action.

Because it is expected that the Phase 1 system might be operated for 6-7 years, the Tenderer is encouraged to deliver components with a life expectancy of at least 7 years for performing typical HPC workloads.

I 85: The mean time to application failure on the fully loaded system due to a system hardware fault requiring user or administrator action should be estimated. An application that is automatically rerun by the batch scheduler is also considered failed.

An estimate and rationale for the expected mean time to failure for the overall system (including the I/O subsystems) and for a single island should be supplied.

The estimated mean time to failure for the complete system is ________ hours.

The estimated mean time to failure for an island of the system is ________ hours.

[Insert text here]

3.7.2 Power Cycling

I 86: All system components should be designed to tolerate power cycling including intentional and unintentional switch-off of power or power failures at least at a rate of once per week on average, without affecting the warranty.

Check here that this requirement will be fulfilled: **[ ]**

I 87: The Tenderer should provide means for orderly power-up and power-down of the complete system. The process and tools used for this procedure should be described.

Check here that this requirement will be fulfilled: [ ]

[Insert text here]

3.7.3 Power Ramps

Power ramps might have severe impact on the facility infrastructure or system components.

I 88: If available, means for controlling power ramp-up and ramp-down of the system should be described.

[Insert text here]

3.7.4 Detection of Hardware Faults

M 53: Facilities for the online detection, monitoring, and reporting of hardware errors (e.g. faulty memory modules, processors, fans, network links, and switches) must be provided. Particularly describe:

- The facilities and processes for diagnosis and error correction
- Whether spare parts are kept on site and how a high availability of the total system will be realized

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.7.5 Log File Aggregation with SPLUNK

LRZ uses SPLUNK[18] to search, monitor, analyze, and visualize machine data and log files.

I 89: The log files of all major system components (compute nodes, login nodes, management nodes, network, storage, etc.) should be reliably forwarded and collected in a SPLUNK instance for analyzing and tracking system and machine problems. The necessary license extension for a maximum ingest rate of 25 GByte/day should be part of the offer.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.7.6 Fault Isolation

I 90: A seamless degradation of the system should be feasible in case of a failure of a single hardware component, such as a network switch, network link, compute node or I/O node.

Check here that this requirement will be fulfilled: **[ ]**

[18] http://www.splunk.com

I 91: If applicable, the Tenderer should concisely describe:

- How the operation of the system can continue with lower performance in case of failure of a component (a compute node, an I/O node, a network switch, etc.).
- How the repaired components are returned to operation and in which cases that will be possible without an interruption of normal system operation.
- In which cases an interruption cannot be avoided.
- Which components may cause an interruption of the complete system or larger proportions of it e.g. central switches.
- Strategies for problem resolution and the provisioning of spare parts.

[Insert text here]

3.7.7 On Site Stock Keeping of Replacement Parts

I 92: The Tenderer should keep a sufficiently large number of spare components on-site and replenish components in a timely manner. If applicable, the Tenderer should provide details of the planned on site system spare parts stock.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.8 Operating System

System components for resource administration and batch administration as well as other components that are essential for system operation are considered as parts of the operating system.

3.8.1 Standards

M 54: The operating system (OS) must provide a complete, supported and stable UNIX-like environment that is appropriate for production usage in a supercomputing system. It must allow flexible administration and control of jobs in interactive and batch mode.

- The system must be delivered with a 64-bit operating system using a 64-bit kernel.
- It must be configurable to comply with standard security guidelines.
- It must support: IPv4, IPv6, TCP/IP, file protocols.

The type and vendor of the deployed operating system must be disclosed.

Check here that this requirement will be fulfilled: **[ ]**

I 93: If a light-weight OS or micro-kernels are used on the compute nodes, the Tenderer should disclose the capabilities and the differences to a full featured OS.

I 94: The operating system should be based on Linux and should be compatible with the X/Open Standard POSIX 1003 (ISO/IEC 9945). The OS licenses and maintenance fees should be part of the proposal.

Check here that this requirement will be fulfilled: **[ ]**

I 95: It should be possible to use IPv6 for all SuperMUC-NG services which have to be reached from external networks or which must connect to external services (login, ntp, DNS, LDAP, file protocols, etc.).

Check here that this requirement will be fulfilled: **[ ]**

I 96: Not later than 3 months before the support for an official OS release installed on SuperMUC-NG ends, the Tenderer should provide certified versions of dependent system software components for the new target OS release.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.8.2 Diskless Nodes

It is intended to operate the compute nodes without disks, since there is no intent to use swap space on the compute nodes, and also because of energy efficiency.

I 97: The compute nodes should be operated without disks.

If available, the diskless provisioning and operation of the compute nodes must be described.

Check here that this requirement will be fulfilled: [ ]

[Insert text here]

3.8.3 Checkpoint/Restart

As a provision against unplanned interrupts, the availability of checkpoint/restart facilities may support efficient job processing.

T 29: Concisely describe the possibilities for checkpointing including information about this feature, if available.

- possibilities of system wide, preventive checkpointing
- multi-threaded or MPI applications or both
- integration with the batch-system
- restart on different nodes than those the checkpoint was written from
- limitations.

[Insert text here]

3.8.4 Suspend/Resume

Suspending a job means interrupting execution of a job and keeping the job and its environment in virtual memory (e.g. the paging space, if available), so that all processor resources used by this job are released and can be used by other jobs. Suspension is especially useful for short term scheduling of jobs on a filled machine or for scheduling a large job by suspending a few smaller ones.

T 30: If available, describe the capabilities of job suspend/resume,

[Insert text here]

3.8.5 Detection and Handling of Resource Overuse

I 98: Describe means for the detection and handling of node resource overuse, particularly out-of-memory conditions.

[Insert text here]

3.9 Resource Management and Batch Scheduling

A highly scalable resource manager and batch scheduling system is crucial for the smooth operation of SuperMUC-NG. It must be tolerant to system failures, including failure of the node executing its control functions. It must be highly scalable, and it must be ready to manage new challenges like energy consumption, power draw, or complex workflows.

The term "Resource Management System" includes resource manager as well as the queueing and scheduling system.

3.9.1 General Objectives

M 55: The resource management system, its components, capabilities, and restrictions must be described.

The description must include how resources and devices (e.g. GPUs) can be managed from both the scheduler and application viewpoint.

M 56: The resource management system must be able to classify batch jobs into different job classes.

Check here that this requirement will be fulfilled: [ ]

M 57: Depending on the chosen job policies, it must be possible to charge for used resources on a per-user, per-project, per account, and on a per-job basis.

The resource management system must provide a database back end to report on:

- user, project, account, job identifier
- date and time spent in particular job states
- allocated compute resources (cores, nodes, memory, I/O, I/O bandwidth, etc.)
- energy consumed by the job
- DC power draw (at least: average, min, and max during job execution)
- exit status

Check here that this requirement will be fulfilled: **[ ]**

M 58: The resource management system must be able to handle jobs up to the size of the whole system.

This also includes reliable removal of all processes started by the job when it is terminated. Furthermore, resources on any accelerators of the system must be reliably assigned and removed at the beginning and end of the job, respectively.

Check here that this requirement will be fulfilled: **[ ]**

M 59: The resource management system must be able to calculate job priorities in a configurable manner, taking job size and type, time, and other factors into account. It must be able to enforce a fair share policy.

Check here that this requirement will be fulfilled: **[ ]**

M 60: The resource management system must support the reservation and allocation of nodes for interactive usage. .

Check here that this requirement will be fulfilled: **[ ]**

M 61: The resource management system must be able to handle dependencies between jobs.

Check here that this requirement will be fulfilled: **[ ]**

I 99: The resource management system should provide functionalities for data staging[19].

Check here that this requirement will be fulfilled: **[ ]**

I 100: The resource management system should provide a fail-safe design, i.e. it should be able to fail over without noticeable interruption of services.

Check here that this requirement will be fulfilled: **[ ]**

I 101: Provisions for notifications about faults on allocated resources should be provided.

A description on how developers will be able to use this information to build fault tolerant applications should be provided. Interaction with fault tolerant MPI is of particular interest.

Check here that this requirement will be fulfilled: **[ ]**

T 31: The preferred resource management system to be delivered with SuperMUC-NG is SLURM

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.9.2 Specific Job Handling

For parameter or replica exchange studies users frequently want to start several, sometimes hundreds, concurrent instances of applications **within one single job** while using the dedicated resources (termed “subjobs” in the following).

Users also want appropriate startup and control mechanisms for filling their allocated nodes optimally. This may include, but is not limited to means for

- specifying the size of a subjob,
- waiting for the completion of specific subjobs,
- reacting on the completion of specific subjobs,
- aborting specific subjobs,
- re-scheduling the execution order and/or backfilling,
- and self-healing of failed subjobs.

[19] A method for handling the data flow across the different storage tiers.

Subjobs require more sophisticated handling than simple job arrays which have all the same characteristics (script, resource requirements) and can be identified by an environment variable.

I 102: The resource management system should provide a way for submission and execution of large numbers of similar jobs (job arrays).

Check here that this requirement will be fulfilled: **[ ]**

I 103: The resource management system should provide means to manage various (possibly parallel) applications and workflows (subjobs) within a single job instance.

If applicable, a description of the capabilities, including the control mechanisms, should be given.

Check here that this requirement will be fulfilled: **[ ]**

I 104: The resource management system should be able to efficiently schedule a very large number of small jobs, at least in the order of twice the number of nodes of the system.

The achievable submission rate should be specified (submitted jobs/second).

Check here that this requirement will be fulfilled: **[ ]**

I 105: The resource management system should be capable of executing a scheduling cycle on a list of a mixture of both large and small queued jobs in a reasonably short time (of the order of a few minutes).

Check here that this requirement will be fulfilled: **[ ]**

I 106: The resource management system should support both backfill scheduling and preemption.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.9.3 Energy and Topology Awareness

I 107: The resource management system should provide means to control power and energy usage, including but not limited to

- energy and power capping
- power ramp-up or ramp down (see also 3.7.3)

Check here that this requirement will be fulfilled: **[ ]**

I 108: The resource management system should be energy-aware i.e. it should support switching unused components to energy-saving mode (e.g. deep sleep mode) as well as the application dependent optimization of processor frequencies.

A description should be provided, including the power draw in deep sleep mode.

Check here that this requirement will be fulfilled: **[ ]**

I 109: The resource management system should be aware of the topology of the interconnect, i.e. be able to perform advantageous placement of jobs to nodes and islands and be able to interoperate with the MPI-implementation's and the OS facilities to perform placement of tasks and threads on each compute node.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.9.4 Allocation of nodes and access to the nodes

I 110: The resource management system should enable users to request resources such as

- number of cores and/or nodes (optionally including accelerators),
- amount of memory,
- specific nodes and islands,
- file systems (e.g., HOME, SCRATCH and WORK),
- and licenses, software, specific files, etc.

Jobs should not be started if the specified resources are not available.

Information about the allocated resources should be accessible from running user applications.

Check here that this requirement will be fulfilled: **[ ]**

I 111: Means to correlate information about jobs and resources with information about faults or inefficiencies should be provided.

A description of these means should be given.

Check here that this requirement will be fulfilled: **[ ]**

I 112: The resource management system should support the control of resource limits at the node level, e.g. memory usage.

Check here that this requirement will be fulfilled: **[ ]**

I 113: The resource management system should not need additional access permissions (such as ssh connections) to start jobs and applications on the compute nodes. The MPI implementation must be able to cope with this restriction.

Check here that this requirement will be fulfilled: **[ ]**

I 114: Means to gather OS information of nodes running a batch job should be available for the users.

Check here that this requirement will be fulfilled: **[ ]**

T 32: It is desired that the resource management system be able to dynamically allocate additional resources during the execution of a job (e.g. allocate additional nodes).

A description of which resources can be handled should be given.

Check here that this requirement will be fulfilled: **[ ]**

T 33: Means for selectively denying or allowing interactive access of users to compute nodes during the execution of their jobs are desired. Interactive user sessions and user access permissions must be securely removed from the node at the end of a job.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.9.5 Reporting

I 115: The resource management system should provide reports about the resource usage.

If applicable, the Tenderer should describe the resource usage features which can be reported by the resource management system (e.g., time, nodes, cores, memory, IO, energy, power, performance in GFlop/s, IOPs, min/max/avg, high water marks, etc.)

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.9.6 Additional Requirements

I 116: The resource management system should provide diagnostics on the state of the scheduler; these diagnostics should be suitable for import into the monitoring system.

Check here that this requirement will be fulfilled: **[ ]**

I 117: Overload or failure of the scheduling system including their causes should be reported.

Check here that this requirement will be fulfilled: **[ ]**

I 118: The resource management system should be capable of supporting Grid software, particularly Globus and UNICORE.[20]

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

[20] Interfacing with UNICORE can rely on support of the UNICORE developers.

3.10 System Management

A stable operation of a system of the size of SuperMUC-NG requires a management infrastructure that supports reliable and efficient administration of the complete system. The configuration management software, the management software for the high-performance interconnect, the resource management system, the monitoring solution, OpenLDAP servers, as well as the hardware required for running these software packages, are considered to belong to this management infrastructure.

Management access to any components of SuperMUC-NG (including any network switches, baseboard management controllers of any host, the root account on any node, etc.) needs to be secured with appropriate ("best practice") security measures. System administration and monitoring needs to be possible out-of-band in a dedicated network, which in the following will be called "management network".

To prepare the installation of both phases of SuperMUC-NG and any maintenances, which will become necessary during the lifetime of SuperMUC-NG, a test system must be delivered and installed at LRZ which includes instances of relevant system components in hardware and software.

3.10.1 System Administration

I 119: At least 6 months before the target date for delivery of the compute and login nodes of either phase, the Tenderer should provide a detailed System Administration Guide.

It should cover the following topics (at a minimum):

- documentation of all involved hardware components
- documentation of all involved software packages
- a design description of the high-performance interconnect
- a design description of the parallel file system(s)
- a design description of the resource management system
- a design description of the system installation and configuration management
- a design description of the monitoring solution
- a description of how to report hardware and software problems to the Tenderer

Wherever possible, the documentation should follow established standards, e.g. IEEE-1016. If applicable, specify which standard the Tenderer follows.

This System Administration Guide must be updated as necessary during performance of the contract and will be subject to approval by LRZ.

Check here that this requirement will be fulfilled: **[ ]**

If the Tenderer cannot fulfill this requirement, an indication of a realistic timeframe should be supplied in the text field at the end of chapter 3.10.1.

The management infrastructure should be set up properly, considering reasonable high-availability setups and IT-security considerations. Per experience, this takes a considerable amount of time and should be done without pressure of time significantly before the installation of the complete main system is performed.

I 120: Based on the target date of the delivery of login and compute nodes of either phase, the corresponding management infrastructure of the system, including all hardware and all software components, should be delivered at least 3 months in advance. If necessary, management servers may be equipped with the predecessor architecture of the compute and login nodes.

Check here that this requirement will be fulfilled: **[ ]**

I 121: Any involved management solution for a given area of activity (e.g. configuration management, management of the high-performance interconnect, resource management, etc.) should scale to the full system size. Each activity, including any that affects the whole system, should be executable from one designated management server. Where appropriate, hierarchical solutions may be employed.

A description of the management infrastructure should be delivered.

Check here that this requirement will be fulfilled: **[ ]**

T 34: It is desired that the central management servers are set up in a high-availability configuration. This in particular includes:

- central configuration servers
- central management servers for the high-performance interconnect
- central servers for the resource management system
- central monitoring servers
- OpenLDAP servers

Check here that this requirement will be fulfilled: **[ ]**

T 35: It is desired that the configuration management solution allows for an automatic synchronization of installation steps across several hosts.

Check here that this requirement will be fulfilled: **[ ]**

T 36: In case the management solution employs a hierarchical approach, it is desired that it allows for a multiple number of hierarchy levels.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.10.2 Test System

Concerning type and number of the individual components, the test system may be as minimal as possible, but must be as complex as required, such that it offers a reasonable testing ground for the system administrators to prepare the installation of the main system as well as hard- and software maintenances. It must allow to duplicate important configurations of the main system.

M 62: For both phases of SuperMUC-NG, the Tenderer must supply a test system.

This test system must feature copies of relevant system components in hardware and software. The test system must be delivered in advance of the main system. The test system setup must be described.

Check here that this requirement will be fulfilled: **[ ]**

I 122: The test system for each phase should be delivered several months in advance of the main system (login and compute nodes) of that phase.

All hardware and software components constituting the management infrastructure of the test system of a phase should be equipped with the identical architecture of their counterparts in the target system of that phase.

The test systems may be delivered with preliminary compute and login nodes, which are equipped with the predecessor architecture. On delivery of the main system these preliminary nodes must be substituted with compute and login nodes featuring the target architecture.

Together with the test system, the vendor provides a description of the design of the main system, which allows for a blueprint installation of the test system.

The time frame of delivery (in months before installation of the main system) should be specified for phase 1:

Delivery date in months before delivery of production system: _______

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.10.3 Monitoring

M 63: A graphical system monitoring application must be provided by the Tenderer.

This system monitoring application must scale to the full system size.

It should allow for zooming from a high-level overview into a detailed display.The system monitoring should display information about all relevant components of SuperMUC-NG. In particular, this should include information about:

- the status of all hosts (hardware sensors, OS status including required daemons, etc.)
- the status of the high-performance interconnect (e.g. switches, defective cables)
- the status of all file systems (hardware, required daemons, fill levels, trends, performance, etc.)
- the status of the cooling infrastructure (e.g. water temperatures, heat transfer, water volume flow)

Check here that this requirement will be fulfilled: **[ ]**

For all hardware components that are delivered in large quantities, processing of hardware failures (reporting, ordering, and replacement) needs to proceed as efficiently as possible; the aim is to reduce the time between detection of a failure and the replacement of the failed part and the efficient management of the expected replacement turnover. Therefore, a suitable process needs to be predefined and implemented in a way that automatizes it as much as possible.

M 64: Hardware failures of common components must automatically be detected and reported to the Tenderer’s support infrastructure.

In particular, this must include the following components:

- compute, login and management nodes (including CPUs, DIMMs, NICs)
- storage devices
- network switches and cabling
- direct liquid cooling infrastructure, where applicable

Check here that this requirement will be fulfilled: **[ ]**

T 37: It is desired that the monitoring solution is either based on or interoperable with Icinga.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.10.4 IT Security

In the following requirements the term “active components” comprises:

- any central management server (servers dedicated to configuration management, high-performance-network management, resource management, as well as OpenLDAP servers, monitoring servers, file system servers, etc.),
- all switches and routers in the management network,
- all switches and routers in the high-performance network,
- all power-distribution units,
- potentially existing disk controllers in storage systems,
- the board management controllers of any compute or login node and all the above servers.

M 65: A dedicated network for system administration and monitoring must be available (“Out-of-band management”).

Management access to any active component of SuperMUC-NG must only be possible via this management network.

Check here that this requirement will be fulfilled: **[ ]**

The goal of the previous requirement is to prevent unauthorized access to and potentially destructive abuse of active components of SuperMUC-NG.

For any support staff of the Tenderer and its subcontractors access to the management network will only be possible through a dedicated gateway node.

T 38: It is desired that access to any node in the management network which is directly accessible from the outside world (e.g. the above dedicated gateway node and any directly accessible management server) is secured with a two-factor authentication scheme.

If offered, the implementation of the two-factor authentication must be described.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.10.5 System Restarts and Upgrades

The times specified by the Tenderer for the following items will be checked after the bring-up of the system.

I 123: The duration of the following tasks should be specified:

- a complete system reboot
- a reboot and re-integration of an island into the whole system
- a reboot and re-integration of a single node into the whole system

Time required for a complete system reboot:

_______________ minutes

Time required for a reboot and re-integration of an island into the whole system:

_______________ minutes

Time required the reboot and re-integration of a node into the whole system:

_______________ minutes

I 124: The time (interruption of regular system operation) required for an operating system upgrade or complete installation of a new operating system on all nodes should be specified:

_______________ hours

[Insert text here]

3.11 Support, Documentation and Training

3.11.1 Problem Solving Mechanisms

The Tenderer must commit to diagnose and resolve hardware and software problems efficiently.

M 66: The Tenderer must describe its incident and problem solving mechanisms including the number of persons involved in this process and how the responsibilities are handled.

I 125: The Tenderer should describe its prioritization scheme and the committed response times.

[Insert text here]

3.11.2 System-Related Support

Typical activities of the system support personnel include, but are not limited to:

- Installation and maintenance of all system software components provided by the Tenderer
- Troubleshooting of system incidents
- Assistance in all system administration related topics
- Assistance in configuration and policy definition and enforcement of the batch queuing system
- OS and system performance tuning
- Installation and performance tuning of file systems
- Configuration and tuning of system management software
- Training of LRZ system administrators

M 67: For the duration of operation of the system, the Tenderer must provide **on-site** personnel responsible for system-related support, critical failure analysis, and maintenance management of the system.

On-site availability of at least **one** skilled persons from 8 am till 6 pm during normal working days is required[21].

On-call availability of at least one skilled person is required from 8 am till 6 pm during weekends and official Bavarian legal holidays.

Check here if the requirements are fulfilled: **[ ]**

[21] An intermission interval for the Tenderer personnel of up to 2 hours is permitted within the 8 am to 6 pm time window.

I 126: During the first year of operation[22], the Tenderer should provide **additional on-site** personnel (at least 1 FTE is expected) which, together with LRZ staff, oversees configuring and putting the system into user operation.

These personnel should have expert knowledge of the system and should maintain the necessary contacts to the Tenderer.

The Tender should specify the number of additional personnel below.

Number of additional on-site personnel in first year operation: ____________
(in Full Time Equivalents)

T 39: Please describe additional possibilities you could offer to LRZ with respect to system administration and tuning.

Office space for Tenderer's personnel will be provided by LRZ; some additional provisions may be needed during the installation process.

T 40: Please specify the (maximum) number of staff members that will need office space during the installation phase of SuperMUC-NG: **________ persons**.

[Insert text here]

3.11.3 User and Application Support

A supercomputer requires staff-intensive user and application support; LRZ here depends on the manufacturer's assistance.

Typical activities of the support personnel include, but are not limited to:

- Handling of user related incidents which cannot be dealt with by LRZ support staff
- Optimization of user codes in case of severe performance problems
- Assistance in optimization and porting of third party software, libraries, and tools
- Assistance on definition of complex workflows, including data handling, grid usage and visualization, pre- and post-processing of data
- Training of users on the Tenderer specific topics like batch system or tools

M 68: For the duration of operation of the system, the Tenderer must provide application support or software engineering personnel[23] (at least one full time equivalent) with expert knowledge about the system architecture and programming environment that assist with optimization-related problems of users.

Check here if the requirements will be fulfilled: **[ ]**

[22] Starting after acceptance of Phase 1

[23] This support may be realized partly off-site

T 41: It is desired that the Tenderer provides additional support for obtaining optimal performance (especially for systems that imply a large shift in programming paradigm compared to the current one) and scalability as well as for porting and optimization of specific software packages.

These personnel need not be on-site at all time, but must be readily available. Please describe the extent of such support for Phase 1.

Number of additional person days for additional support: ______

Check here that this request will be fulfilled: [ ]

[Insert text here]

3.11.4 Documentation

I 127: All the documentation for the system and its software products should be available for LRZ staff. The documentation should be viewable and searchable on-site even without access to the internet.

Check here that this requirement will be fulfilled: **[ ]**

I 128: Documentation for the end user should be electronically accessible[24].

Check here that this request will be fulfilled: **[ ]**

[Insert text here]

3.11.5 Introduction to the System

I 129: Training for the LRZ system administrators as well as an introduction for selected users and LRZ staff to usage of the system should be provided.

Check here that this request will be fulfilled: **[ ]**

[Insert text here]

[24] Access restrictions may be applied.

3.12 Software

The offered software stack must be reliable and scalable. Even for jobs which use the whole system, severe bottlenecks must not occur and system-side resource usage should not prevent a job from executing.

For every software package mentioned in this section, please give an indication, if only third-party manufacturers can provide it.

3.12.1 Programming Models

I 130: A description of the programming models that are needed to obtain suitable performance levels (particularly, if accelerated nodes or many-core nodes are included in the offer) should be supplied.

[Insert text here]

3.12.2 Major Software Components

Major software components are those that are essential for the scientific programmer to develop and deploy large scale simulations on the SuperMUC-NG system. In particular, the compilers, performance libraries, MPI implementations, debuggers as well as tuning, profiling, and verification tools fall into this category.

I 131: If major software components are delivered by a third party, specify what level of support is available for each delivered package and whether LRZ can be provided with direct access to the support structure of the software provider.

I 132: Specify the Tenderer's scope of responsibility and warranty for the major software components.

I 133: A suitable licensing scheme including the license fees for the major software components should be part of the offer and be included in the maintenance costs.[25]

Please describe the licensing schemes.

[Insert text here]

[25] LRZ already has campus licenses for some compilers, libraries and tools which may be used or extended.

M 69: The Tenderer must fill in the requested baseline descriptions of the major software components, including their licensing scheme and volume and the assignment of support responsibility for each component in Table 4**.**

Some lines in Table 4 may require replication; further lines can be added by the Tenderer if upcoming technology necessitates a new entry.

Component	Purpose	Licensing and Volume	Support deliverer, responsibility and support level
Operating system	Basic node operation services		
Interconnect drivers	For the high-performance interconnect		
Accelerator drivers	Enables use of accelerator		
MPI implementation	Distributed memory parallel programming interface		
Compilers	For base languages Fortran, C, C++		
Basic optimized performance libraries	Linear Algebra, FFT, etc.		
Debugger	Failure analysis		
Profiling and tracing	Performance/Bottleneck analysis		
Verification tools	Program correctness checking		
[Additional entries by Tenderer]			

Table 4: Baseline description of major software components.

3.12.3 MPI: Message Passing Interface

The MPI implementation must be scalable. Even for very large applications, the resource usage (e.g. buffer memory) must not be excessive and start-up times as well as connection times must be moderate. Process management for start-up and for termination must be reliable and predictable.

M 70: The offer must include at least one scalable MPI implementation which is capable of efficiently running jobs up to the size of the whole system without excessive system-side resource usage.

Check here whether this requirement will be fulfilled: **[ ]**

M 71: The offer must include a complete and conforming implementation of the MPI 3.1 standard and any errata that relate to this version. The implementation must also contain explicit interfaces for all calls in the Fortran MPI module.

Check here whether this requirement will be fulfilled: **[ ]**

M 72: If GPUs are offered, the MPI implementation must be capable of directly handling communication buffers that reside on the GPU without explicitly copying data to the host memory (*GPU-awareness*).

Check here whether this requirement will be fulfilled: **[ ]**

M 73: A thread-safe MPI implementation (MPI_THREAD_MULTIPLE) must be provided.

Check here whether this requirement will be fulfilled: **[ ]**

I 134: The Tenderer should describe the MPI implementation(s), particularly:

- Which implementation flavor it is based on (e.g. MPICH, OpenMPI)
- The scaling behavior in general (e.g. maximum supported node and task count)
- Resource usage and its scaling behavior (e.g. whether there is a complexity order with respect to node or task count)
- Tools available for tuning and/or automation of algorithm selection for collective communication
- Implementation and restrictions for the all-to-all types of collective communication
- Offloading functionalities, if applicable.
- If GPUs are offered, the following should be described:
  - How MPI handles data buffers residing on GPUs,
  - How multiple MPI processes can share one GPU and multiple GPUs
  - How the resource sharing of multiple kernels simultaneously executing on the GPUs is handled, and what performance issues may arise,
  - How multiple MPI tasks running on the CPUs can access memory on the GPUs (e.g., how MPI Shared Memory Model introduced with MPI-3 can be handled), and what performance issues may arise
  - How computation and communication can be overlapped

I 135: The Tenderer should disclose information about the fraction of the memory **typically** needed for internal buffers for message passing (MPI): The scaling behavior of the buffer size should be specified as a function of communication partners and several examples for typical regular patterns (e.g. 3-D nearest neighbor, all-to-all, the MPI settings as used for the benchmark runs, etc.) should be provided. Furthermore, the Tenderer should discuss the tradeoff between performance and buffer size.

Amount of memory per core ________________________ Mbytes

Amount of memory per node ________________________ Mbytes

The amount of MPI internal buffering needed for the execution of the HPL benchmark run as a pure MPI program on the whole Phase 1 system should be disclosed.

Amount of memory per core ________________________ Mbytes

Amount of memory per node ________________________ Mbytes

I 136: The maximum startup time for a pure MPI application spanning the whole system should be specified.

(The startup time is defined as the time between the start of mpiexec and the completion of the execution of MPI_INIT i.e. it includes the time for the spawning of the MPI processes and the time to establish the connections between the processes.)

Maximum Startup time: ______________ Seconds

I 137: The MPI implementation should support the MPI_F08 Fortran interface module, with the following values for specific constants:

- MPI_SUBARRAYS_SUPPORTED with value .TRUE.
- MPI_ASYNC_PROTECTS_NONBLOCKING with value .TRUE.

Check here whether this requirement will be fulfilled: **[ ]**

T 42: It is desired that development kits (including front ends e.g. mpif90, mpicc etc.) to the MPI subsystem are provided for alternative compilers (e.g. PGI, GCC).

Check here whether this requirement will be fulfilled: **[ ]**

M 74: It must be possible to use the **high performance parallel file systems (SCRATCH and WORK)** with high efficiency and performance for MPI-IO.

A description of the MPI hints available for tuning the MPI-IO layer must be supplied.

Check here whether this requirement will be fulfilled: **[ ]**

I 138: It should be possible to use all shared file systems of SuperMUC-NG for MPI-IO.

Check here whether this requirement will be fulfilled: **[ ]**

I 139: The MPI implementation should include a mechanism for deadlock detection. If included, a concise description should be given.

Check here whether this requirement will be fulfilled: **[ ]**

Fault tolerance functionality in MPI, i.e. the ability of an MPI program to continue if a subset of the tasks in MPI_COMM_WORLD have failed due to a system or hardware problem, is a feature that is currently in development by the MPI-Forum within the scope of its MPI 4.0 effort. The target date for finalizing the MPI 4.0 standard currently is 2018. This feature will enable advanced users to explore the possibilities for fault-tolerant program development, enabling them in the long run to significantly reduce cycle losses caused by single node failures and increase throughput for large and long-running jobs.

T 43: It is desired that the MPI implementation supplies the fault tolerance feature targeted for MPI-4.0 within one year after the MPI-4.0 standard is released.

Check here whether this requirement will be fulfilled: **[ ]**

[Insert text here]

3.12.4 Other Communication Libraries

I 140: At least the following communication libraries should be available and should make efficient use of the interconnect hardware and its drivers:

- An implementation of GASPI (e.g. GPI)
- Global Arrays
- GASnet

Check here whether these requirements will be fulfilled: **[ ]**

Typically, a low level communication library is supplied that provides support for various parallel programming models such as MPI, Global Arrays, OpenSHMEM, GASPI, PGAS, and others.

I 141: The concepts underlying the low-level communication library should be described, if available.

T 44: Describe which other message passing implementations or communication libraries are available (OpenMPI, MPICH, etc.)

[Insert text here]

3.12.5 Compilers

M 75: A least one set of supported, optimizing compilers including a Fortran, a C, and a C++ compiler must be provided. The C compiler must be able to serve as a companion processor for the Fortran compiler.

A sufficient number of licenses including support must be included in the offer.

Check here whether this requirement will be fulfilled: **[ ]**

The following requirements refer to this "mainline" set of compilers.

On large HPC systems, Fortran is still an important language with respect to percentage of code bases as well as executed cycles. The additional features that are provided by the current standard (and go beyond Fortran 95) permit programming on a more abstract level by using object-oriented concepts, allow to exploit dependency inversion for improved design by using submodules, support interoperation with the C language family, and allow for simplified parallel programming by using coarrays. A significant fraction of applications uses C or C++ as implementation language, relying on the respective language standards. Recent additions to the C++ language semantics contribute to efficiency and scalability of programs.

I 142: The Fortran compiler should constitute a full implementation of the language specification ISO/IEC 1539-1: 2010 ("Fortran 2008"), including the Corrigenda 1-4.

Check here whether this requirement will be fulfilled: **[ ]**

I 143: The Fortran compiler should supply the "extended interoperability with C" features as specified in ISO/IEC TS 29113, to be published as part of the upcoming Fortran 2015 standard. This feature is needed for a high quality implementation of the MPI 3.1 Fortran interface.

Check here whether this requirement will be fulfilled: **[ ]**

I 144: The C compiler should constitute a full implementation of the standard ISO/IEC 9899:2011 („C11"):

Check here whether this requirement will be fulfilled: **[ ]**

I 145: The C++ compiler should constitute a full implementation of the standard ISO/IEC 14882:2014 („C++14"), including the C++ standard library.

Check here whether this requirement will be fulfilled: **[ ]**

OpenMP is a standardized method for threading and vectorization of suitable parts of HPC codes. In combination with MPI, it provides the basis for efficient hybrid programming schemes. More recent versions of the standard also support programming of attached accelerator devices.

M 76: The Fortran, C, and C++ compilers must support the OpenMP 4.5 standard.

Check here whether this requirement will be fulfilled: **[ ]**

I 146: The Fortran, C, and C++ compilers should support the additional offloading features of the "most recent OpenMP standard at the time of the Phase 1 acceptance" within two years after its publication.

Check here whether this requirement will be fulfilled: **[ ]**

T 45: It is desired that C++17 is made available within two years after publication of that standard. The Tender should describe any limitations not meeting the full standard.

Check here whether this requirement will be fulfilled: **[ ]**

T 46: It is desired that OpenMP 5.0 is available within two years after its release.

Check here whether this requirement will be fulfilled: **[ ]**

OpenACC enables portable exploitation of an accelerator's computational capabilities via a directive-driven approach. The features currently defined in OpenMP do not fully cover the needed semantics for doing this. As a result, many third-party codes are already using OpenACC. In combination with MPI, OpenACC provides an alternative basis for efficient hybrid programming schemes.

I 147: If the offer contains computational nodes with accelerators, the Fortran, C, and C++ compilers should support the OpenACC standard on at least the level of version 2.5.

Check here whether this requirement will be fulfilled: **[ ]**

Additional compiler features, some of which are not formally standardized, are either helpful during the development and tuning process or needed for legacy codes.

I 148: The compilers should support:

- Parallelization, vectorization and optimization reports [ ]
- Cray Pointer Functionality [ ]
- POSIX Threads [ ]
- Interoperability with GCC (GNU compiler collection) [ ]
- GCC extensions within C and C++ [ ]
- Detection and handling of floating point exceptions [ ]
- Compiler based instrumentation (like GCCs "-finstrument-functions") [ ]

Please check each requirement above individually.

Recently, a technical specification ISO/IEC TS 18508 ("additional parallel features in Fortran") was released that defines features significantly improving the usability, composability, and scalability of the coarray programming model. This technical specification will be integrated in to the upcoming Fortran 2015 standard.

I 149: The interoperability (including name mangling) of the offered compilers should be described.

T 47: It is desired that the Fortran compiler supplies a full implementation of the technical specification ISO/IEC TS 18508 within one year after the acceptance of Phase 1.

Check here whether this requirement will be fulfilled: **[ ]**

T 48: Check here whether the TS 18508 implementation will support continued execution in the face of failed images.

Check here whether this requirement will be fulfilled: **[ ]**

The upcoming Fortran 2015 standard, beyond integrating ISO/IEC TS 29113 and TS 18508, provides additional useful semantics in the language that improve convenience and safety of programming.

T 49: It is desired that the mainline Fortran compiler supplies a full implementation of the future Fortran 2015 standard, within two years after publication of that standard.

Check here whether this requirement will be fulfilled: **[ ]**

[Insert text here]

3.12.6 Other Compilers, Libraries, Components and Development Tools

I 150: It should be assured that the GNU compiler collection (GCC) can be built for the system or is provided by the Tenderer, and that the generated code can be executed with reasonable performance.

Check here that this requirement will be fulfilled: **[ ]**

I 151: Emerging compilers based on PGAS concepts should be available.

Check individually for available implementations:

- UPC (Unified Parallel C) **[ ]**
- UPC++ (Unified Parallel C++) **[ ]**
- coarray C++ **[ ]**
- [insert additional ones, if available] **[ ]**

I 152: If the offer contains computational nodes with accelerators, the Fortran, C, and C++ compilers should support the CUDA programming paradigm at least on the level of CUDA 8.0

Check here whether this requirement will be fulfilled: **[ ]**

I 153: An OpenCL implementation supporting OpenCL 2.2 or higher should be available. Check individually for available implementations:

- On host CPUs **[ ]**
- On heterogeneous architecture (if the offer contains accelerators) **[ ]**

T 50: It is desired that the statistical R programming framework and its extensions, particularly those for parallel execution, are available.

Check here whether this request will be fulfilled: **[ ]**

T 51: It is desired that other compilers and compiler-like tools are available for the system (e.g. Java, PGI, LLVM). A list including the scope (general purpose or accelerated nodes) may be supplied.

Check here whether this request will be fulfilled: **[ ]**

[Insert text here]

3.12.7 Task and Thread Location and Pinning

I 154: Mechanisms for the placement of tasks and threads should be provided. Mechanisms for both querying and controlling the placement from within the user's program should also be provided. This applies for MPI, OpenMP, other shared memory paradigms, or combinations thereof. If available, please give a short description or reference.

Check here that this requirement will be fulfilled: **[ ]**

The Portable Hardware Locality (hwloc, https://www.open-mpi.org/projects/hwloc/) software package provides a portable abstraction (across OS versions, architectures, etc.) of the hierarchical topology of the architectures.

I 155: The Portable Hardware Locality software package should be available for the compute node and attached devices.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.12.8 Stack Traces and Lightweight Core Files

Traditional core files tend to be large and can consume too much disk space, particularly for large parallel applications. Lightweight core files contain, primarily, stack traces (listings of function calls that led to the error).

I 156: Compilers and runtime should support aggregated stack traces (redundant information is removed) for parallel applications.

A lightweight core file format should be available; which core file format is used should be selectable at runtime.

A concise description for the feature should be supplied.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.12.9 Libraries

M 77: At least the following highly efficient and optimized numerical libraries must be available and supported by the Tenderer:

- A scientific library including FFT routines, random number generators, linear algebra, etc.
- Optimized BLAS and LAPACK (serial and shared memory parallel version)
- ScaLAPACK, BLACS
- FFTW (serial and parallel versions)[26]

Versions of these libraries must exist that interoperate with applications that utilize OpenMP.

Check here whether these requirements will be fulfilled: **[ ]**

I 157: At least the following numerical libraries should be available and supported by the Tenderer:

- PETsc
- Trilinos
- GNU Scientific Library.

Versions of these libraries should exist that interoperate with applications that utilize OpenMP.

Check here whether these requirements will be fulfilled: **[ ]**

I 158: In the case when different compiler suites available on the system use different name mangling schemes, the Tenderer should supply all necessary versions of all libraries to permit seamless usage by these compilers (i.e. without requiring additional compiler switches that change name mangling).

Check here that this request will be fulfilled: [ ]

T 52: It is desired that the NAG Fortran Libraries (serial and shared memory parallel versions) are available.

Check here that this request will be fulfilled: **[ ]**

I 159: Different applications will use the I/O system in different ways. Depending on the specific I/O patterns, tuning for either bandwidth or metadata operation rate may be necessary. The tuning measures may be implemented in the file system’s I/O layer (e.g. buffering), or may be possible through calls to a specific I/O library.

The tools available for tuning I/O should be described.

[26] A proprietary implementation must support the FFTW interface

M 78: At least the following I/O libraries must be available and be supported:

- HDF5 (both serial, OpenMP and MPI-integrated variants)
- NetCDF (both serial, OpenMP and MPI-integrated variants)

Check here whether this requirement will be fulfilled: **[ ]**

T 53: It is desired that the Sionlib library (http://www.fz-juelich.de/ias/jsc/EN/Expertise/Support/Software/SIONlib/_node.html) be available on the system

Check here whether this requirement will be fulfilled: [ ]

T 54: It is desired that the ADIOS library be available on the system

Check here whether this requirement will be fulfilled: [ ]

T 55: A documentation of methods for tuning I/O via a file system specific facility, if available, is desired.

I 160: The system should include an optimized Python implementation including MPI4py, NumPy and SciPy.

If offered, the extend of optimization on accelerated nodes must be described

Check here whether this requirement will be fulfilled: **[ ]**

[Insert text here]

3.12.10Scope of Support for Scientific Libraries

M 79: The Tenderer must describe the availability and the degree of optimization (optionally for Host and Accelerator) for the numerical libraries listed in M 77: and I 157:.

The appropriate fields in Table 5 must be checked.

[Insert text here]

BLAS Level 1, 2, 3 and LAPACK:	General purpose nodes	If applicable: many-core	If applicable: accelerator
Fully supported by Tenderer; Tenderer handles support requests	[ ]	[ ]	[ ]
Fully supported by code owner; code owner accepts and handles support requests for offered architecture	[ ]	[ ]	[ ]
Available in source code; no specific support for the offered architecture	[ ]	[ ]	[ ]
Available in source code, build process supports the offered architecture	[ ]	[ ]	[ ]
Extensively optimized by Tenderer (e.g. handcoding, intrinsics, assembler)	[ ]	[ ]	[ ]
Extensively optimized by code owner (e.g. handcoding, intrinsics, assembler)	[ ]	[ ]	[ ]
Only Partially optimized for the offered architecture; other parts are unoptimized but functionality is provided	[ ]	[ ]	[ ]
Tenderer has contract with code owner for making the software available	[ ]	[ ]	[ ]
Sparse BLAS	General purpose nodes	If applicable: many-core	If applicable: accelerator
Fully supported by Tenderer; Tenderer handles support requests	[ ]	[ ]	[ ]
Fully supported by code owner; code owner accepts and handles support requests for offered architecture	[ ]	[ ]	[ ]
Available in source code; no specific support for the offered architecture	[ ]	[ ]	[ ]
Available in source code, build process supports the offered architecture	[ ]	[ ]	[ ]
Extensively optimized by Tenderer (e.g. handcoding, intrinsics, assembler)	[ ]	[ ]	[ ]
Extensively optimized by code owner (e.g. handcoding, intrinsics, assembler)	[ ]	[ ]	[ ]
Only Partially optimized for the offered architecture; other parts are unoptimized but functionality is provided	[ ]	[ ]	[ ]
Tenderer has contract with code owner for making the software available	[ ]	[ ]	[ ]

Petsc/Slepsc	General purpose nodes	If applicable: many-core	If applicable: accelerator
Fully supported by Tenderer; Tenderer handles support requests	[ ]	[ ]	[ ]
Fully supported by code owner; code owner accepts and handles support requests for offered architecture	[ ]	[ ]	[ ]
Available in source code; no specific support for the offered architecture	[ ]	[ ]	[ ]
Available in source code, build process supports the offered architecture	[ ]	[ ]	[ ]
Extensively optimized by Tenderer (e.g. hand-coding, intrinsics, assembler)	[ ]	[ ]	[ ]
Extensively optimized by code owner (e.g. hand-coding, intrinsics, assembler)	[ ]	[ ]	[ ]
Only Partially optimized for the offered architecture; other parts are unoptimized but functionality is provided	[ ]	[ ]	[ ]
Tenderer has contract with code owner for making the software available	[ ]	[ ]	[ ]
Trilinos	General purpose nodes	If applicable: many-core	If applicable: accelerator
Fully supported by Tenderer; Tenderer handles support requests	[ ]	[ ]	[ ]
Fully supported by code owner; code owner accepts and handles support requests for offered architecture	[ ]	[ ]	[ ]
Available in source code; no specific support for the offered architecture	[ ]	[ ]	[ ]
Available in source code, build process supports the offered architecture	[ ]	[ ]	[ ]
Extensively optimized by Tenderer (e.g. hand-coding, intrinsics, assembler)	[ ]	[ ]	[ ]
Extensively optimized by code owner (e.g. hand-coding, intrinsics, assembler)	[ ]	[ ]	[ ]
Only Partially optimized for the offered architecture; other parts are unoptimized but functionality is provided	[ ]	[ ]	[ ]
Tenderer has contract with code owner for making the software available	[ ]	[ ]	[ ]

FFTW	General purpose nodes	If applicable: many-core	If applicable: accelerator
Fully supported by Tenderer; Tenderer handles support requests	[ ]	[ ]	[ ]
Fully supported by code owner; code owner accepts and handles support requests for offered architecture	[ ]	[ ]	[ ]
Available in source code; no specific support for the offered architecture	[ ]	[ ]	[ ]
Available in source code, build process supports the offered architecture	[ ]	[ ]	[ ]
Extensively optimized by Tenderer (e.g. hand-coding, intrinsics, assembler)	[ ]	[ ]	[ ]
Extensively optimized by code owner (e.g. hand-coding, intrinsics, assembler)	[ ]	[ ]	[ ]
Only Partially optimized for the offered architecture; other parts are unoptimized but functionality is provided	[ ]	[ ]	[ ]
Tenderer has contract with code owner for making the software available	[ ]	[ ]	[ ]
Eigen-Solvers (shortly describe the package)	General purpose nodes	If applicable: many-core	If applicable: accelerator
Fully supported by Tenderer; Tenderer handles support requests	[ ]	[ ]	[ ]
Fully supported by code owner; code owner accepts and handles support requests for offered architecture	[ ]	[ ]	[ ]
Available in source code; no specific support for the offered architecture	[ ]	[ ]	[ ]
Available in source code, build process supports the offered architecture	[ ]	[ ]	[ ]
Extensively optimized by Tenderer (e.g. hand-coding, intrinsics, assembler)	[ ]	[ ]	[ ]
Extensively optimized by code owner (e.g. hand-coding, intrinsics, assembler)	[ ]	[ ]	[ ]
Only Partially optimized for the offered architecture; other parts are unoptimized but functionality is provided	[ ]	[ ]	[ ]
Tenderer has contract with code owner for making the software available	[ ]	[ ]	[ ]

Table 5: Degree of optimization

3.12.11Programming Environment and Tools

A large part of the effort of preparing large scale simulations is spent by program development, debugging and tuning. To make this process more efficient, a consistent and seamless programming environment is considered very useful.

T 56: An integrated programming environment that allows to develop, start, debug and optimize parallel programs, and to visualize performance data, is desired (e.g. via integration into Eclipse) and should be concisely described.

Check here whether this request will be fulfilled: **[ ]**

I 161: Means for users should be available to determine and define the placement of tasks as well as threads via environment variables or API calls. If the offer contains accelerators, the means should include an easy way to control resource assignment on the accelerator.

Check here that this requirement will be fulfilled: **[ ]**

T 57: It is desired that the MPI implementation as well as the tools for performance analysis support the Structured Trace Format (STF) and/or Open Trace Format (OTF) trace file formats.

Check here that this request will be fulfilled: **[ ]**

I 162: For performance and communication analysis of message passing programs, a tool with functionality equivalent to Score-P with Scalasca, Vampir (www.score-p.org) or the Intel Cluster / Intel Parallel Studio Tools should be available. Means for instrumenting program code at compile/link time should be provided.

Check here whether this requirement will be fulfilled: **[ ]**

A prerequisite for performance analysis is the availability of an infrastructure for accessing hardware performance counters.

M 80: Access to the hardware performance counters (including counters for energy consumption) from user space must be available and be supported.

Check here whether this requirement will be fulfilled: **[ ]**

T 58: It is desired that the Tenderer provides support for a port of the LIKWID performance analysis tool (see https://github.com/RRZE-HPC/likwid).

Check here whether this requirement will be fulfilled: **[ ]**

The performance application programming interface (PAPI, see http://icl.cs.utk.edu/papi) offers a reasonably standardized and easy-to-use programming interface.

M 81: The operating system must provide support for PAPI.

Check here whether this requirement will be fulfilled: **[ ]**

T 59: Means for automatic calling of user-specified functions at subroutine entry and exit (user hook functions for insertion of user specified calls) are desired.

Check here that this requirement will be fulfilled: **[ ]**

I 163: A tool for instrumentation and analysis of an application's I/O patterns (e.g. Darshan) should be available.

It should support the mainline MPI implementation for all its base languages.

Check here whether this requirement will be fulfilled: **[ ]**

An efficient workflow for scientific code development on large HPC systems requires a mature set of tools for isolating correctness problems

M 82: A scalable parallel debugger with a GUI must be available.

Its scalability features must be described. Both the number of MPI tasks supported by the debugger and the license must permit debugging of a program that uses 1024 to 4096[27] license tokens. Each token represents an MPI task. Debugging capabilities for larger jobs for a fixed number of days per year must be included.

The software license and support for this debugger for the complete operation duration of Phase 1 must be included in the offer. If the offer contains accelerated compute nodes, the same debugger must be able to debug accelerated code.

Check here that this requirement will be fulfilled: **[ ]**

I 164: The Totalview or Allinea DDT debugger should be provided, including memory debugger and support for accelerators or many core systems, if offered. The license and maintenance costs should be included in the offer.

Check here that this request will be fulfilled: **[ ]**

T 60: It is desired that a port of the GNU debugger (GDB) be available for the system, including accelerator support if the offer uses accelerators.

Check here that this request will be fulfilled: [ ]

A specific area of correctness and performance checking is for parallel programming models inside a shared memory node.

[27] LRZ expects that the actually offered number of tokens will depend on the node architecture.

I 165: The Tenderer should provide and support tools that permit the programmer to perform correctness checking and performance analysis of threaded code written for execution on a shared memory system. A description of the tools should be supplied. If an accelerated node architecture is offered, tools for the correctness checking and performance analysis of offloaded code sections should be included.

Check here that this request will be fulfilled: [ ]

The goal of **UNITE** (UNiform Integrated Tool Environment) is to provide a robust, portable, and integrated environment for the debugging and performance analysis of parallel MPI, OpenMP, and hybrid MPI/OpenMP programs on high-performance compute clusters. It consists of a set of well-accepted portable, mostly open-source tools. Detailed information is available at https://apps.fz-juelich.de/unite/index.php/Main_Page

T 61: It is desired that the Tenderer describes to which extent the UNITE installation process is supported for the offered architecture and which of the UNITE packages are available for the proposed system.

T 62: It is desired that the tools listed in Table 6 be available on the proposed system.

Check the appropriate fields in Table 6.

Tool Name	**Description**	**Available for general purpose processors ? If yes, give current version number.**	**If applicable: Available for many-core processors ? If yes, give current version number.**	**If applicable: Available for (Host + Accelerator) ? If yes, give current version number.**
Allinea MAP, Performance Reports	Performance analyzer			
HPCtoolkit	Performance Analysis from Rice Univ.			
Intel tools	VTune Amplifier, Trace Analyzer, Inspector, Advisor			
ompP	OpenMP profiler from NERSC			
Periscope	Scalable automatic performance analysis from TU Munich			
Scalasca	Performance analysis			
Score-P	Performance analysis			
Vampir	Performance Analysis (TU Dresden)			
Further tools	**To be added by Tenderer**			

Table 6: Performance analysis tools

[Insert text here]

3.12.12 Availability of Applications and Libraries

Currently, LRZ provides a large spectrum of applications and libraries on its SuperMUC system as well as its Linux-Cluster.

T 63: It is desired that the applications listed in
Table 7 are available on the compute nodes of SuperMUC-NG to provide the user with a seamless migration path.

Check the appropriate fields in Table 7.

[Insert text here]

Package	**Description**	**Available for General purpose processors ? If yes, give current version number.**	**If applicable: Available for many-core processors ? If yes, give current version number.**	**If applicable: Available for (Host + Accelerator) ? If yes, give current version number.**
Abinit	electronic states in solids			
Amber	molecular dynamics			
Ansys	Finite Elements			
Blast	Genomics			
Boost	C++ utilities			
CFX (Ansys)	computational fluid dynamics			
Charmm	molecular dynamics			
Clc bio	genome bench			
Comsol	Multi-physics			
CP2K	atomistic and molecular simulations			
CPMD	Car-Parrinello Molecular Dynamics			
Desmond	quantum chemistry package			
Fire, AVL Fire	CFD			
Fluent	computational fluid dynamics			
Gamess	quantum chemistry package			
Gaussian	quantum chemistry package			
Globus	Grid Software			
Gromacs	molecular dynamics			
Lammps	large-scale molecular dynamics			
Marc	general purpose finite element code			
Materials studio/ CASTEP	Quantum and catalysis software			
Magma	Linear Algebra			
Matlab	computer algebra			
Metis, Parmetis	Graph partitioning			
Mumps	Parallel sparse solver			
MSC Nastran	Structural mechanics			
Namd	molecular dynamics			
Nwchem	quantum chemistry package			
Octave	computer algebra			
Openfoam	computational fluid dynamics			

Plasma	Linear Algebra			
Plumed	Gromacs pluggin			
Powerflow	computational fluid dynamics			
Qespresso	electronic states in solids			
R	statistics and data visualization			
Schrodinger	quantum chemistry package			
Siesta	quantum chemistry package			
Specfem3d	geophysics			
Tecplot	FEM visualisation			
TAU	TAU parallel performance toolset			
Turbomole	quantum chemistry package			
Vasp	quantum chemistry package			
Wannier90	solid state code			
Wien2k	quantum chemistry package			
yt project	analysis and visualization toolkit for volumetric data			
Other packages	**Further relevant HPC packages may be inserted by Tenderer**			

Table 7: Availability of applications

3.12.13 Virtualization of the Software Environment

Rapid deployment of applications can be supported by making use of virtualization concepts, i.e. packing an application and its dependencies into a virtual container. If the application was built on a system with binary compatible processor architecture and similar interconnect network hardware, this guarantees to some extent that the application can be executed without recompilation on SuperMUC-NG. Hence, the concept of compute node virtualization will enable users to use applications, tools and libraries from other HPC sites or Linux distributions on SuperMUC-NG. The concept of HPC node virtualization will also give users an easy opportunity to tailor and adapt the run-time environment to the needs of their applications.

Virtualized or container-based environments into which users can load their own operating system images are, for example, docker, shifter, singularity, or proot. The preferred solution is based on singularity. Each approach has tradeoffs of ease of use against security of deployment that should be justified for the offered choice.

I 166: Solutions for virtualization of the software environment should be available.

If available, a description should be given and it should be described how they are integrated into the system and the resource management/batch system.

Check here that this request will be fulfilled: **[ ]**

[

I 167: The virtualization/container environments should run in user space (without requiring privileged access to the system or allowing a user to escalate privilege) and provide a layer on top of the local operating system without the need of changing the operating system image of the compute or login nodes.

Possible security implications should be discussed.

Check here that this request will be fulfilled: **[ ]**

I 168: It should be possible to use the low level drivers of the interconnect and possible accelerators in the virtualization/container environments.

Check here that this request will be fulfilled: **[ ]**

[Insert text here]

3.13 Migration Path

I 169: The potential migration path from the current SuperMUC system to SuperMUC-NG for user applications should be described, including but not limited to:

- changes in the programming paradigm
- changes of source code
- optimization and tuning
- scalability issues
- dependencies on OS, libraries, glibc etc.
- third party and ISV software
- potential performance gains
- sustainability and maintainability of the resulting code base

I 169a: Note – new requirement:

The expected migration path between Phase 1 and Phase 2 of SuperMUC-NG should be described, including as many of the details spelled out in the previous requirement as currently possible.

[Insert text here]

3.14 Cloud Resources and Remote Visualization

3.14.1 Cloud Concept

It is intended that SuperMUC-NG includes a cloud resource to provide services to users that cannot be offered otherwise. These services contain the ability to run a user-defined software environment such as databases or web frontends to supercomputing applications. Such environments shall be based on x86 images.

The desired functions of the cloud include testing, pre- and post-processing of data, remote visualization, and strong user separation. The users should be able to configure their Virtual Machines (VMs) according to their needs, and without interfering with other users' VMs and software. From within these VMs users might submit jobs to the (non-virtualized) SuperMUC-NG compute nodes, and communicate with an executing job in user-land via network sockets.

Within the cloud, regular login facilities will likely be available, i.e. some (possibly virtualized) login nodes might provide shared user login to all users on the same login node for those users who do not require their own environment and thus want to avoid the extra effort of setting up their own VMs.

The requirements specified in the following sections oblige the Tenderer to deliver hardware and its support for the cloud component. Furthermore, the Tenderer must deliver a concept and possibly some software infrastructure that permits interfacing the cloud component to the compute hardware of SuperMUC-NG.

LRZ will be responsible for installing and maintaining the cloud software stack; LRZ will inform the Tenderer about the decision on the specifics of the software stack to be deployed. It is expected that part of the work to be done for interfacing between the compute part and the cloud component will be implemented in the framework of a collaboration between Tenderer and LRZ.

3.14.2 Cloud Hardware

The hardware components listed in this section will be preferably installed in either LRZ computer room NSR0 or NSR1 (see section 3.1.3). Warm water cooled systems might also be installed in HRR.

The hardware resources to be procured are listed in the following Table 8.

Number of units	Type	Cores	RAM (GByte)	Local Storage	Ethernet bandwidth (GBit/s per node)
32	Standard 2 socket cloud node	≥ 40	≥ 192	diskless	100
32	2 socket Cloud nodes with one GPU each	≥ 40	≥ 768	diskless	100
1	Huge memory node	≥ 80	≥ 6000	diskless	200

Table 8: Cloud nodes

M 83: Note – this requirement was modified.

The hardware listed in Table 8 must be delivered as part of the cloud component. All nodes must be based on the same generation of an x86 processor architecture. The per-node HPL (double precision) performance of a 2 socket node of the offered processor architecture must **meet or exceed 2.0 TFlop/s**.

Check here that this requirement will be fulfilled: **[ ]**

M 84: The internal cloud network must be primarily based on 100 GbE Ethernet technology (preferably 100GBaseSR4).

This implies that switches and cabling with sufficient port counts must be delivered to integrate the hardware components as well as providing uplinks and cabling for integration with both the LRZ backbone and the SuperMUC-NG gateway solution (Section 3.14.3). 100 GbE must be implemented for those connections. A non-blocking switch solution for cloud nodes is not required.

- The number of uplinks to the LRZ backbone must support at least 200 GBit/s throughput.
- The number of uplinks to the SuperMUC-NG gateway must support at least 200 GBit/s throughput.

The Ethernet switches and the delivered cabling technology must be described.

Check here that this requirement will be fulfilled: **[ ]**

I 169b Note – new requirement:

If ethernet connections span different rooms, the cabling to the cloud should be implemented in a structured manner, i.e. using a modular cabling system. See also I 39a (Section 3.3.6) for technical details.

Check here that this requirement will be fulfilled: [ ]

It is desired that the ethernet infrastructure for the cloud can be easily integrated into the LRZ network.

T 63a Note – new requirement:

The preferred vendor for the switch component that provides the uplinks to the LRZ backbone switches is HP.

Check here that this requirement will be fulfilled: [ ]

[Insert text here]

3.14.3 SuperMUC-NG Gateway to the Cloud

M 85: The Tenderer must provide a gateway that provides a transport interface between the high performance interconnect of SuperMUC-NG and the Ethernet-based cloud components.

This gateway must support an aggregate Ethernet bandwidth of at least 200 GBit/s.

The offered gateway solution must be described.

Check here that this requirement will be fulfilled: **[ ]**

The preferred solution for the gateway is either a single hardware component, or a high-reliability appliance.

I 170: The gateway solution should provide high availability in the face of failure of a single hardware component. It is permissible that the available bandwidth degrades in case of such a failure.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.14.4 Interfacing of SuperMUC-NG with the Cloud

I 171: A facility should be provided and described that permits an export of all file systems available on SuperMUC-NG (e.g., via NFS) through the gateways described in Section 3.14.3, on a per-user basis.

Check here that this requirement will be fulfilled: **[ ]**

I 172: A mechanism (e.g., a dynamic firewall solution) should be provided and described that permits temporarily setting up network connections between virtual machines (VMs) executing on the cloud and other parts of the infrastructure through an automatic provisioning process.

Check here that this requirement will be fulfilled: **[ ]**

Establishing the provisioning process could be part of a collaboration effort. The scenarios supported include (but are limited to):

- Supplying the network ports needed for NFS mounts of parallel file systems via the gateways described in Section 3.14.3 (root-level),
- Supplying the network ports needed for NFS mounts of HOME or PROJECT storage (root-level),
- Supplying the network ports needed for accessing a job submission service (probably user-level),
- Supplying the network ports needed for access to a specific application executing on a subset of the compute nodes from a VM e.g., license management, visualization component of computational steering, remote debugging/program analysis (user-level only)

None of the expected scenarios require RDMA accesses between cloud VMs and other parts of the infrastructure.

T 64: It is desired that job submission to SuperMUC-NG should be possible from a VM in the cloud, through a web interface or a web API. If such a facility is available, it should be described.

Check here that this requirement will be fulfilled: [ ]

Implementing a feature complete solution for job submission from a VM in the cloud could be part of a collaboration effort.

[Insert text here]

3.14.5 Grid Software

I 173: Globus and UNICORE should be available for the targeted platform.

Check here that this requirement will be fulfilled: **[ ]**

[Insert text here]

3.14.6 Remote Visualization

The HPC cloud includes nodes with graphics pipes (GPUs) for remote visualization (see Section 3.14.2). The GPUs can be used for rendering the graphics output, which is then transported over the network using a tunnel from the cloud nodes to the output devices. Data from the compute nodes is available via the common high performance parallel file system(s).

I 174: The Tenderer should provide a concept for remote visualization. The concept should include information about the availability of corresponding graphics drivers and remote visualization software.

[Insert text here]

3.15 Maintenance

M 86: A 365 x 10 hours on-site **hardware** maintenance contract fulfilling the following condition must be offered:
During peak operation times (from 8 am till 6 pm), the availability of technicians and parts on-site within 4 hours after a problem report must be guaranteed for critical hardware errors (errors which have a high impact on the SuperMUC-NG service quality).

Check here if the requirements are fulfilled: **[ ]**

M 87: A 365 x 10 hours **software** maintenance contract fulfilling the following condition must be offered:

Response to critical software errors (errors which have a high impact on the SuperMUC-NG service quality) must happen within 4 hours.

Check here if the requirements are fulfilled: **[ ]**

I 175: The total cost for hardware and software maintenance for a 6-year term of operation of Phase 1 should be specified in the following table:

Total cost for HW maintenance Phase 1:	___________ €
Total cost for SW maintenance Phase 1:	___________ €

I 175a: Note – new requirement.

The total cost for hardware and software maintenance for a 4-year term of operation of Phase 2 should be specified in the following table:

Total cost for HW maintenance Phase 2:	___________ €
Total cost for SW maintenance Phase 2:	___________ €

I 175b: Note – original text was converted to a new requirement:

Further details about which parts of the system and its software are considered critical, the guaranteed repair times for both critical and non-critical components, and possible delivery times for spare parts should be supplied.

These details will be regulated in the maintenance contract.

I 176: The annual cost for hardware and software maintenance for an extension of operation beyond the planned 6-year term should be specified in the following table:

Annual cost for HW maintenance for year 7:	___________ €

Annual cost for SW maintenance for year 7:	___________ €

[Insert text here]

3.16 Market Penetration

I 177: The Tenderer should describe how many systems similar to the proposed one are installed or are expected to be installed in the period 2018/19 and the size of these systems.

“Similar” here means: Processors with the same or nearly same instruction set, accelerator solutions with the same baseline architecture, a high performance interconnect based on compatible technology or with comparable performance characteristics.

I 178: The Tenderer should indicate the smallest commercially available configuration with the same system architecture and programming environment which can be deployed locally in a science department for application development.

[Insert text here]

3.17 Price and Scaling

Although it is not required to disclose the price of different components of the system, it might be useful to know the relative costs to scale the system during the negotiation process.

T 65: It is desired that the Tenderer fill in Table 9 with the relative cost of system components.

Item	**Relative Cost (Percent)**
Nodes	%
Cloud Nodes	%
Interconnect	%
High performance parallel File System	%
DSS File system (incl. additional licenses)	%
Software (OS, libraries, tools provided by Tenderer)	%
Sum:	**100 %**

Table 9: Relative cost of system components

Beyond the present procurement, deployment of additional hardware might become necessary during the lifetime of the system. Either LRZ itself or other academic or research institutions may appear as the owner of such a system (“Housing”). Such systems would be installed and maintained in the LRZ facilities, though possibly separately from SuperMUC-NG.

T 66: This requirement was extended and converted to the level “(I)mportant” – see below.

I 178a:

The Tenderer should specify conditions and options (including volume pricing) for procuring additional compute, cloud, storage and network/cabling hardware of the same or similar type as that to be delivered with this procurement (options for follow-up orders or a framework agreement).

[Insert text here]

3.18 Collaboration with the Tenderer

T 67: A collaboration with the Tenderer is desired.

Please describe the scope, extent, and organizational structure of such a collaboration.

[Insert text here]

3.19 Benchmarks

M 88: A document with the results for the benchmarks as stipulated in the *Decision Criteria and Benchmark Description SuperMUC-NG* must be handed over to LRZ.

4 Tables of Key System Parameters

Entries for Phase 2 are non-binding but should provide credible estimates of the proposed offer. Note that, formally, it is permitted to fill in the tables with values of “unknown”, but this will imply a qualitative devaluation in the area of “quality and credibility of the offer”. It should definitely be avoided to not specify key values for the system performance.

4.1 System Reliability

Item and Unit	**Phase 1**
Hardware Mean Time to Failure for the entire system (hours)	

4.2 Test System

Item and Unit	Test System
Footprint of the system incl. maintenance areas: length x width (m x m)	
Dimension of system racks: width x depth x height (m x m x m)	
Total weight (kg)	
Maximum rack weight per square meter (kg/m^2)	
Maximum Point Load (N)	
Type of cooling (chilled water, warm water, air)	
Ambient temperature (range) (°C)	
Relative humidity (range) (%)	
Number of nodes	
Type of processor	
High-performance interconnect (type)	

4.3 Infrastructure

In the table below, the terms “peak” and “expected” power draw appear. These are defined as follows.

The **peak electrical power draw** of the whole system is defined as the sum of the following power draw values:

- Power draw of the system excluding the network components when running a single node HPL benchmark at maximum processor frequency on all compute, service, management, cloud and login nodes and, if applicable, attached accelerators
- Maximum power which can be theoretically consumed by all storage and network components of the system.

For any specific subset of compute nodes, the peak power draw is that of running the single node HPL benchmark at maximum processor frequency on that subset, including attached accelerators if applicable.

The **expected electrical power draw** is the power draw under typical operating conditions e.g., fully loading the system with as equal as possible shares of the application benchmarks and executing these applications with energy optimized processor frequencies. Turbo mode must be disabled for all processing elements.

Item and Unit	**Phase 1**	**Phase 2 estimates**
Footprint of the system incl. maintenance areas: length x width (m x m)		
Number of racks		
Dimension of system racks: width x depth x height (m x m x m)		
Total weight (kg)		
Maximum rack weight per square meter (kg/m^2)		
Maximum point load (N)		
Ambient temperature (range) (°C)		
Relative humidity (range) (%)		
Air purity of computer room (μg/m^3)		
Coolant type of disk storage (e.g. water, air)		
Maximum tolerable supply temperature of cooling Loop 2 (°C)		
Allowed inlet temperature range of compute node coolant (°C)		
Expected outlet temperature range of compute node coolant (°C)		
Allowed inlet temperature range of disk storage coolant (°C)		
Expected outlet temperature range of disk storage coolant (°C)		
Idle power draw of total system (kW)		
Expected electrical power draw of total system (kW)		
Peak electrical power draw of total system (kW)		
Expected electrical power draw of thin nodes (kW)		

Peak electrical power draw of thin nodes (kW)		
Expected electrical power draw of fat nodes (kW)		
Peak electrical power draw of fat nodes (kW)		
Expected electrical power draw of accelerated nodes (kW), if offered		
Peak electrical power draw of accelerated nodes (kW), if offered		
Expected electrical power draw of disk storage (kW)		
Peak electrical power draw of disk storage (kW)		
Expected electrical power draw of communication network (kW)		
Peak electrical power draw of communication network (kW)		
Expected heat emission of total system into air (kW)		
Expected heat emission of total system into water loop 1[28] (kW)		
Expected heat emission of total system into water loop 2 (kW)		
Peak heat emission of total system into air (kW)		
Peak heat emission of total system into water loop 1[28] (kW)		
Peak heat emission of total system into water loop 2 (kW)		
Expected heat emission of disk storage into air (kW)		
Expected heat emission of disk storage into water loop 1[28] (kW)		
Peak heat emission of disk storage into air (kW)		
Peak heat emission of disk storage into water loop 1[28] (kW)		
Expected heat emission of network components into air (kW)		
Expected heat emission of network components into water loop 1[28] (kW)		
Peak heat emission of network components into air (kW)		
Peak heat emission of network components into water loop 1[28] (kW)		
External connection voltage e.g. as required by power distribution units (V)		
Number of electrical phases		
Frequency of electrical current (Hz)		

[28] Please only specify the heat which will be emitted via indirect water cooling of the indicated component in this field (i.e. do not add the heat which will be emitted into air)

4.4 Compute Node Architecture

4.4.1 General Purpose Compute Nodes

For the thin and large shared memory nodes of the offered system, the following table must be filled in. If a subset of the thin nodes will be equipped with accelerators, the table of Section 4.4.3 must be filled in. The general purpose processor type and all information not relating to accelerators must be entered in the table below. The same procedure applies for the large shared memory nodes.

Item and Unit	**Phase 1 thin nodes**	**Phase 1 large shared memory nodes**
Processor type		
Total number of nodes of this type		
Number of islands with this node type		
Number of processors (sockets) per node		
Number of cores per processor		
Number of hardware threads per core		
Nominal frequency (GHz)		
Peak frequency (GHz)		
Floating point operations per core per clock (multiply/add, no div or sqrt, Fused Multiply-Add counts as 2)		
Floating point instructions per core per clock (Fused Multiply-Add counts as 1, vector instructions count as 1 etc.)		
Thread Count per physical core necessary to achieve full floating point operation and instruction rates		
Peak number of instructions per second of one core at nominal frequency (GInst/s)		
Peak floating point performance of one core at nominal frequency (GFlop/s)		
Peak floating point performance of **one node** that can be sustained by the node using higher frequencies than nominal (GFlop/s).		
If applicable: Number of vector registers per core		
If applicable: Length of vector registers (in units of bits)		
Caches		
Number of physical cores sharing one L1 data cache		
Number of physical cores sharing one L2 data cache		
Number of physical cores sharing one L3 data cache		
Size of (single) L1 data cache (kByte)		
Size of (single) L2 data cache (kByte)		
Size of (single) L3 data cache (MByte)		
Bandwidth of (single) L1 data cache (GByte/s)		
Bandwidth of (single) L2 data cache (GByte/s)		

Bandwidth of (single) L3 data cache (GByte/s)		
Latency of L1 data cache (core clock cycles)		
Latency of L2 data cache (core clock cycles)		
Latency of L3 data cache (core clock cycles)		
DDR-based Memory		
Memory type (vendor, model)		
Memory technology		
Memory frequency (MHz)		
Single memory module/chip capacity (GByte)		
Total memory capacity of a node (GByte)		
Maximum configurable size of memory of one node (GByte)		
Memory latency at nominal core frequency (ns)		
Memory latency at peak core frequency (ns)		
Memory bandwidth per node at nominal core frequency (GByte/s)		
Memory bandwidth per node at peak core frequency (GByte/s)		
Interconnect Integration of Node		
Adapter type (vendor, model)		
Number of adapters per node		
Number of ports (physical links) per adapter		
Number of active ports per adapter		
Physical link speed (GByte/s) per direction		
Node to switch hardware latency (ns)		

4.4.2 Optional: Many-Core Nodes

Filling in the following table is only required if the Tenderer intends to deliver many-core thin nodes as part of the system.

Item and Unit	**Phase 1**
Processor type	
Total number of nodes of this type	
Number of islands with this node type	
Number of processors per node	
Number of cores per processor	
Number of hardware threads per core	
Nominal frequency (GHz)	
Peak frequency (GHz)	
Mean time between failures of a single node (days)	
Floating point operations per core per clock (multiply/add, no div or sqrt, Fused Multiply-Add counts as 2)	
Floating point instructions per core per clock (Fused Multiply-Add counts as 1, vector instructions count as 1 etc.)	
Thread Count per physical core necessary to achieve full floating point operation and instruction rates	
Peak number of instructions per second of one core at nominal frequency (GInst/s)	
Peak floating point performance of one core at nominal frequency (GFlop/s)	
Peak floating point performance of one node that can be sustained by the node using higher frequencies than nominal. (GFlop/s).	
If applicable: Number of vector registers per core	
If applicable: Length of vector registers (in units of bits)	
Caches	
Number of physical cores sharing one L1 data cache	
Number of physical cores sharing one L2 data cache	
Number of physical cores sharing one L3 data cache	
Size of (single) L1 data cache (kByte)	
Size of (single) L2 data cache (kByte)	
Size of (single) L3 data cache (MByte)	
Bandwidth of (single) L1 data cache (GByte/s)	
Bandwidth of (single) L2 data cache (GByte/s)	
Bandwidth of (single) L3 data cache (GByte/s)	
Latency of L1 data cache (core clock cycles)	
Latency of L2 data cache (core clock cycles)	

Latency of L3 data cache (core clock cycles)	
DDR-based Memory	
Memory type (vendor, model)	
Memory technology	
Memory frequency (MHz)	
Single memory module/chip capacity (GByte)	
Total memory capacity of a node (GByte)	
Maximum configurable size of memory of one node (GByte)	
Memory latency at nominal core frequency (ns)	
Memory latency at peak core frequency (ns)	
Memory bandwidth per node at nominal core frequency (GByte/s)	
Memory bandwidth per node at peak core frequency (GByte/s)	
Interconnect Integration of Node	
Adapter type (vendor, model)	
Number of adapters per node	
Number of ports (physical links) per adapter	
Number of active ports per adapter	
Physical link speed (GByte/s) per direction	
Node to switch hardware latency (ns)	

4.4.3 Optional: Accelerators (Accelerated Nodes)

Filling in the following table is only required if the Tenderer intends to equip part of the compute nodes with accelerator devices.

Item and Unit	**Phase 1** **thin shared memory nodes**	**Phase 1 large shared memory nodes (if applicable)**
Accelerator type (vendor, model)		
Host node type (thin, fat) into which accelerator is integrated		
Integration technology used (e.g. PCIe, NVLink)		
Number of accelerated nodes		
Number of accelerator devices per host node		
Single precision (IEEE) peak performance of an accelerator device with ECC active (GFlop/s)		
Double precision (IEEE) peak performance of an accelerator device with ECC active (GFlop/s)		
Local memory of an accelerator device (GByte)		
Bandwidth to local memory of an accelerator device (GByte/s)		
Number of Streaming Multiprocessors (SM) per accelerator device		
Number of Streaming Processors (SP) per SM		
Number of Special Function Units (SFU) per SP		
Latency for host ↔ accelerator data transfers (µs)		
Bandwidth for host ↔ accelerator data transfers (per accelerator device, GByte/s)		
Latency for accelerator ↔ accelerator data transfers inside a node (µs, only if more than one device is present in a node)		
Bandwidth for accelerator ↔ accelerator data transfers inside a node (GB/s, only if more than one device is present in a node)		

4.4.4 Optional: Specialized Memory Architectures

Filling in the following table is only required if the Tenderer intends to equip part of the compute nodes with special memory architectures. If other technologies than High Bandwidth Memory are offered the table must be suitably replicated.

Item and Unit	**Phase 1**
High Bandwidth Memory (HBM)	
Memory technology and HBM type	
Memory frequency (GHz)	
Node types equipped with HBM (all, or list specific types)	
Capacity per node (GByte)	
Maximum configurable size of HBM per node (GByte)	
Bandwidth per node (GByte/s) at nominal core frequency	
Latency (ns) at nominal core frequency	
Bandwidth per node (GByte/s) at peak core frequency	
Latency (ns) at peak core frequency	

4.5 High Performance System Interconnect

Item and Unit	Phase 1
Type of interconnect (technology, vendor)	
Network topology	
Number of switches in first level hierarchy	
Number of downlink ports of switches in first level hierarchy	
Number of uplink ports of switches in first level hierarchy	
If applicable, number of switches in second level hierarchy	
If applicable, number of downlink ports of switches in second level hierarchy	
If applicable, number of uplink ports of switches in second level hierarchy	
Theoretical bisection bandwidth of the entire system (TByte/s)	
Maximum time for a broadcast (1 Byte length) to reach all nodes (µs)	
Maximum MPI send-receive (1 Byte length) latency between two arbitrary communication partners (nodes) (µs)	
Minimum MPI send-receive (1 Byte length) latency between two arbitrary communication partners (nodes) (µs)	

4.6 IO Subsystem

Item and Unit	**Phase 1**
Aggregated bandwidth of DSS CES NFS export cluster (GiBytes/s)	
Aggregated bandwidth of DSS CES CIFS export cluster (GiBytes/s)	
Capacity of the DSS primary HOME SSD Tier (TiByte)	
Number of 4K inodes that can be stored in the DSS HOME meta-data area	
Usable capacity of primary HOME (TiByte)	
Usable capacity of secondary HOME (TiByte)	
Aggregated bandwidth of the primary HOME SSD tier (GiBytes/s)	
Aggregated bandwidth of the primary HOME HDD tier (GiBytes/s)	
Aggregated 4K random read IOPS from HOME SSD tier	
Aggregated 4K random write IOPS to HOME SSD tier	
Aggregated 4K random read IOPS from HOME HDD tier	
Aggregated 4K random write IOPS to HOME HDD tier	
Replication performance between primary and secondary HOME (GiBytes/s)	
Number of 4K inodes that can be stored in the DSS PROJECT meta-data area	
Usable capacity of DSS PROJECT (PiByte)	
Aggregate bandwidth of DSS PROJECT (GiBytes/s)	
Usable capacity of the High Performance Parallel File Systems (PiByte)	
Aggregated bandwidth of the High Performance Parallel File Systems (GiBytes/s)	
Usable capacity of caches for accelerating latency-bound IO patterns (TiByte)	

END OF DOCUMENT