

Defining the requirements of your next HPC cluster

UCit – April 2025



Table of Contents

0) Why planning for your next HPC cluster is more challenging than ever	3
1) Analyzing your needs	4
1.1) A picture of the present... ..	4
1.1.1) Analyzing total cluster load	5
1.1.2) Analyzing workload distribution	6
1.1.3) What are your workloads running on?	8
1.1.4) Detecting wasted resources.....	9
1.1.5) Analyzing peaks, dips and congestion in workload demand	11
1.2) ...to plan for the future	14
2) Choosing the infrastructure: a workload-based approach	15
2.1) Sizing the cluster(s).....	15
2.1.1) Defining the total compute power needed	16
2.1.2) Should you have one large cluster or multiple clusters?	16
2.1.3) Where should the compute resources be located?	17
2.1.4) Planning for scalability	18
2.2) CPU vs GPU: is your balance right?	18
2.3) Cloud vs on-prem: optimizing flexibility and cost.....	19
2.4) What configuration for your cluster(s)?	19
2.4.1) Choosing the right node configuration and processor architecture	19
2.4.2) Defining the storage & filesystem strategy	20
2.4.3) Ensuring software stack & application compatibility	21
3) Optimizing costs & resource utilization	21
3.1) Improving resource allocation to reduce waste.....	22
3.2) Modeling costs over the next 3–5 years	23
4) Future-proofing: long-term planning & sustainability.....	24
4.1) Choosing a future-proof hardware strategy	24
4.2) Assessing energy efficiency & sustainability.....	24
4.3) Cost-efficient long-term operations	25
5) Conclusion: from analysis to action	26

0) Why planning for your next HPC cluster is more challenging than ever

Your old cluster is approaching its limits. GPUs are in demand, workloads are shifting, and AI's knocking at the door... how do you plan what's next?

Replacing an HPC cluster today isn't just about adding more power. It's about making the right choices in a fast-changing environment: balancing HPC and AI needs, defining the right balance between cloud and on-prem, selecting the right hardware amongst a myriad of options, and ensuring cost-efficiency... all while keeping an eye on sustainability. How can you future-proof your next investment?

You've probably realized this already: workloads are changing. Traditional HPC applications like simulations and modeling now run alongside AI and data analytics, requiring different types of compute power. Should your next cluster focus solely on CPUs, or should it include GPUs and accelerators optimized for AI?

And the hardware question doesn't stop there. Today the choice isn't just between Intel and AMD anymore: ARM is gaining traction, and specialized processors offer new possibilities. The right decision depends on compatibility, power efficiency, and how your workloads behave. Storage, networking, and job schedulers play a critical role in performance: they add complexity to the architectural equation.

Costs are another major factor. Many clusters suffer from inefficiencies, with idle resources leading to wasted computing power. At the same time, cloud computing offers scalability but comes with financial trade-offs. Should you keep everything on-prem, move to the cloud, or adopt a hybrid model?

Finally, sustainability has become a growing priority. With rising energy costs and environmental concerns, organizations need to measure and optimize their cluster's carbon footprint (now a legal obligation in many regions). Power consumption, cooling strategies, and workload distribution all impact long-term efficiency. How will you approach these choices?

In this article, we'll break down the key questions to ask when planning your next HPC cluster.

1) Analyzing your needs

1.1) A picture of the present...

Choosing your next HPC cluster starts with understanding how your current one is being used. By analyzing today's workloads, you can paint a relevant picture of the present situation: one that will help you align future infrastructure with evolving needs.

The best way to assess the current state of your cluster and its usage is through actual data. This is where cluster logs come in handy: they contain a wealth of insights about how your infrastructure is performing, how resources are allocated, and where inefficiencies might lie.

At UCit, we use OKA to ingest anonymized cluster logs and display them in easy-to-read dashboards. But you can also use other tools such as Grafana, Tableau, or even Excel to visualize the data and extract meaningful trends.

At this stage, the main questions we want to answer are:

- What is the overall load of the cluster today? → *This will help understanding the need for infrastructure adaptations overall.*
- Which workload types are dominant? (AI, simulations, data analytics, etc.) → *This will help evaluate how to distribute computing resources (CPU vs GPU, etc.).*
- Where are your workloads running? How are cluster resources allocated, and are they used efficiently? → *This will help identify optimizations in workload execution.*
- Are there any signs of an inefficient/waste use of resources from your users? → *This will help assess the potential waste of resources due to a lack of training or scheduler misconfiguration.*
- Do certain workloads have specific needs? → *This will help adapt resources for your future clusters, and help you choose which workload should run where.*
- Are there recurring workload peaks or underutilized periods? → *This will help determine the right mix of fixed on-premise resources vs. variable cloud resources.*

For each of these questions, there are probably a dozen ways to play with cluster data and reach a valuable insight. Let's break some of them down.

To do this, we'll take a typical example: say your organization runs different types of computations—fluid dynamics simulations (CFD), engineering analytics, and machine learning models for predictive maintenance.

- CFD simulations: R&D teams rely heavily on computational fluid dynamics to model aerodynamics for product design. These workloads are CPU-intensive, often requiring high-memory nodes with high-speed interconnects. They also generate large amounts of output data that need to be stored and post-processed.
- Engineering analytics: operational teams use data analytics to evaluate product performance, optimize manufacturing processes, and assess material fatigue. These workloads are less computationally heavy but require quick access to historical datasets.
- AI for predictive maintenance: the organization has started integrating machine learning models to predict equipment failures and optimize maintenance schedules. These AI workloads benefit from GPU acceleration but don't run continuously—they are triggered periodically based on new sensor data.

1.1.1) Analyzing total cluster load

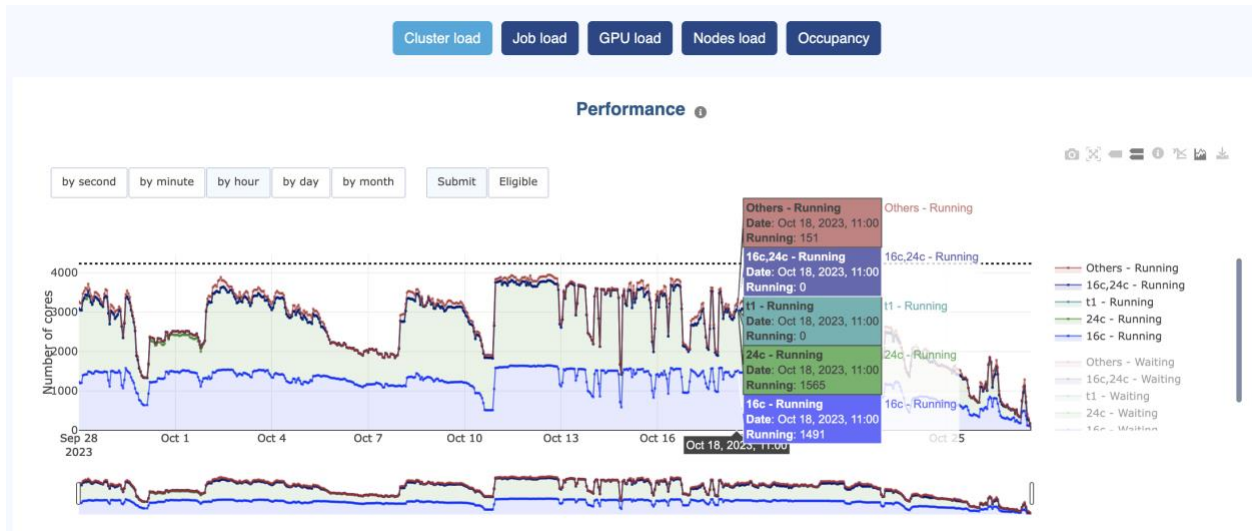
Before diving into a detailed workload analysis, it's important to step back and assess the overall load on your cluster. This gives you a high-level understanding of how much your infrastructure is being used, whether it meets current demand, and where potential bottlenecks or inefficiencies exist.

A global view of cluster activity helps answer key questions:

- How much of your total compute power is actively used?
- Are resources consistently maxed out, or are there idle periods?
- Are there recurring congestion points in your queues?
- How does cluster usage translate into costs, energy consumption, and carbon footprint?

With OKA's Load panel for example, you can track how compute resources are being used across different timeframes. This panel provides a clear breakdown of core utilization, GPU allocation, and queue congestion.

In the example below, we can see the core usage over time:



This type of view helps determine if the cluster is adequately used, and can also give us hints to know if it is adequately sized. If demand consistently exceeds capacity, additional nodes or workload rebalancing may be required. Conversely, if significant idle time exists, it may indicate over-provisioning or an opportunity to shift workloads dynamically to the cloud.

If peak loads consistently overwhelm the cluster, this justifies expanding resources—whether by adding on-prem nodes or bursting to the cloud. If demand fluctuates widely, a hybrid HPC approach could optimize efficiency and costs.

1.1.2) Analyzing workload distribution

The next question to ask is: what types of workloads are consuming your cluster's resources?

To answer this, you can break down the distribution of workloads across different job types in terms of:

- Number of jobs submitted → *Which workloads dominate in terms of job count?*
- Compute resources used (core-hours, GPU-hours, memory) → *Which workloads are the most resource-intensive?*
- Cost impact → *How much does each workload type contribute to the overall cost of running the cluster?*

Getting this information from your logs is very easy to do using OKA's Resources panel:



For example, let's assume your analysis reveals the following:

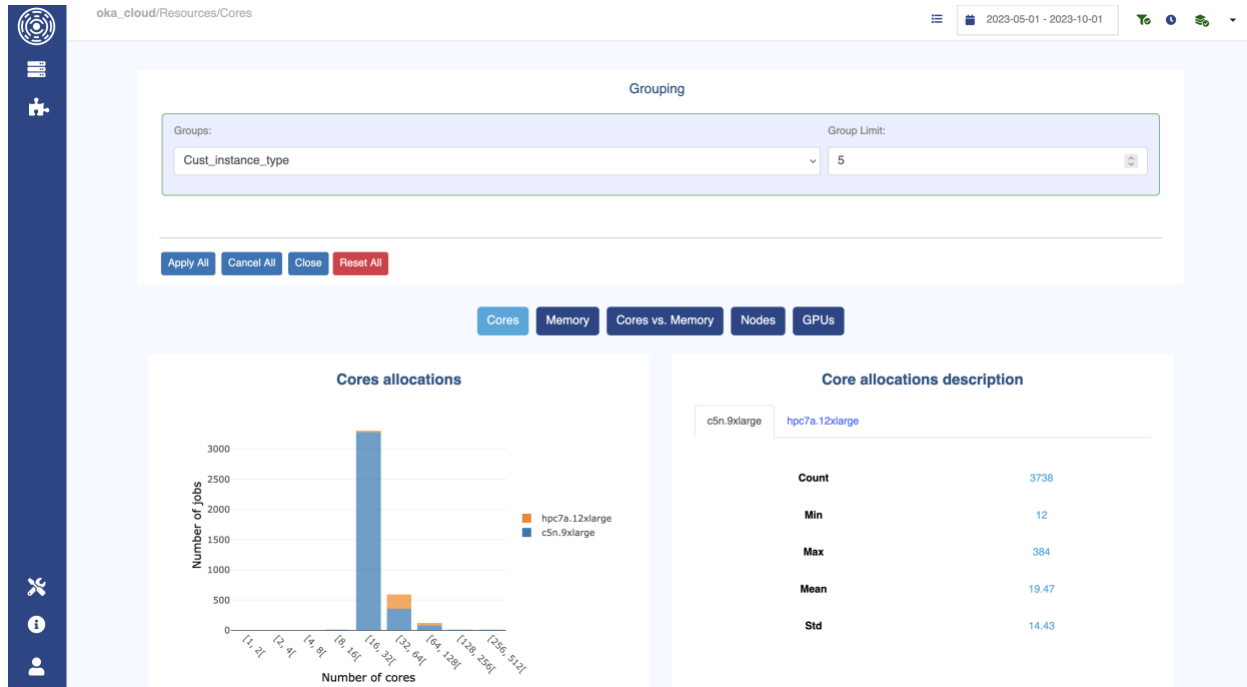
- 70% of jobs are traditional HPC simulations (CFD, FEA, molecular dynamics, etc.), running mostly on CPUs.
- 20% are engineering analytics, which use moderate CPU resources but require significant storage capacity.
- 10% are AI workloads, mostly deep learning training jobs running on GPUs.

This breakdown helps you see if your current cluster is well-aligned with actual usage. If AI workloads are increasing but you have limited GPU availability, this could indicate a need for more GPU-accelerated resources in your next cluster.

1.1.3) What are your workloads running on?

Once you understand what is running, the next step is to analyze how the scheduler handled workload placement. Are workloads optimally placed on the right hardware?

With OKA's Resources panel, you can categorize jobs based on the type of nodes or instances they ran on.



This allows you to:

- Identify if workloads are running on the appropriate compute nodes (e.g., GPU workloads on GPU nodes).
- See if any workloads are running on overpowered or underutilized hardware (e.g., simple data analysis jobs consuming high-memory compute nodes).
- We could also think of ways to detect whether workloads are spilling over to cloud instances and if those cloud resources are cost-justified.

For the sake of our example, the analysis may show that:

- Most CFD workloads run on high-core-count CPU nodes, but some large simulations are also using GPU nodes unnecessarily.
- AI training jobs run on GPUs, but demand often exceeds available GPU capacity, leading to jobs sitting in queues.
- A portion of batch analytics jobs are running on expensive compute nodes, when they could be handled by lower-tier instances.

This data allows you to optimize workload placement and identify whether your next cluster should maintain the same balance or shift toward different configurations.

1.1.4) Detecting wasted resources

A critical part of workload analysis is identifying wasted resources. Inefficiencies in job scheduling, poorly configured job submissions, and failed jobs can lead to significant waste.

For example, OKA's Consumers panel helps pinpoint inefficiencies by listing users with the highest number of timed out jobs:

The screenshot displays the 'Consumers' panel in the OKA interface. At the top, an 'Advanced filters' section shows a rule: 'State equal TIMEOUT'. Below this are buttons for 'Apply All', 'Cancel All', 'Close', and 'Reset All'. The main section is titled 'CONSUMERS' and includes a 'Values in category: UID' summary with the following statistics:

- Total number of values: 113
- Number of jobs: 30722
- Total consumed core-hours: 6035162

A table below shows the top consumers, sorted by UID. The table has columns for UID, Count, Core_hours, Allocated_cores, Allocated_GPUs, Execution_time, Waiting_time, Slowdown, Cost, Energy, and CO2.

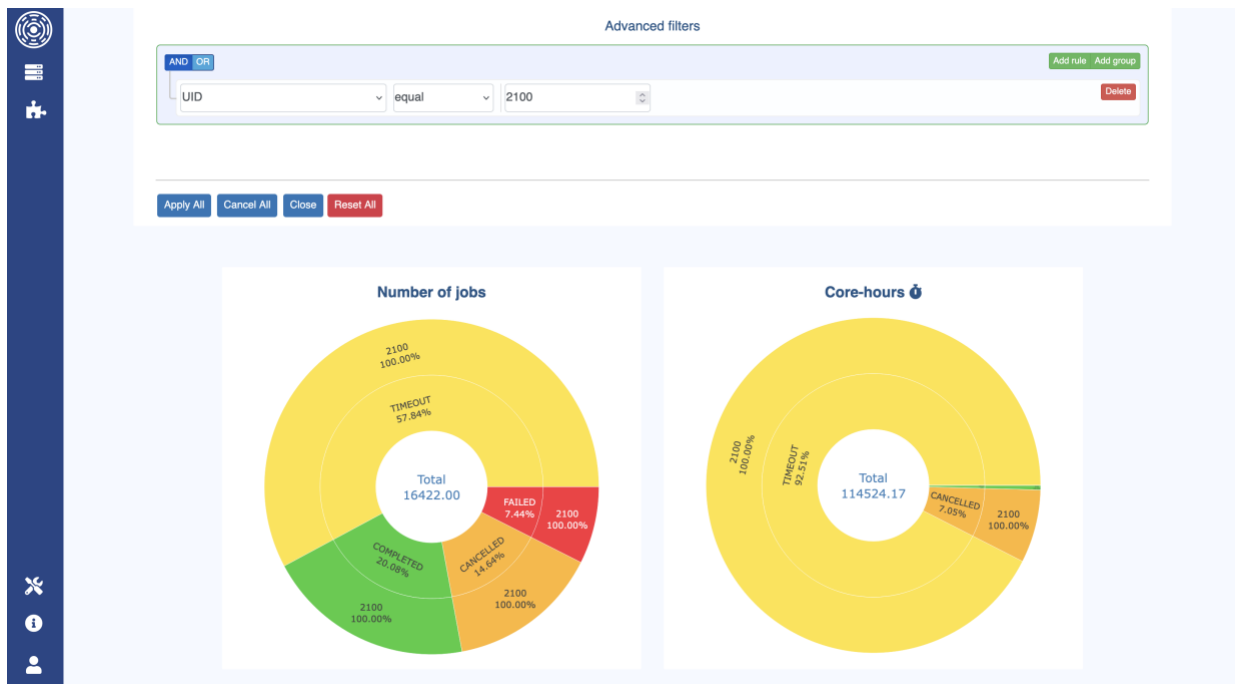
UID	Count	Core_hours	Allocated_cores	Allocated_GPUs	Execution_time	Waiting_time	Slowdown	Cost	Energy	CO2
2100	9498 (30.92%)	105951.3 (1.76%)	1	-	2h 45m 33s	5h 27m 42s	6.51	2119.03	1.37	0.85
2042	4135 (13.46%)	254660.4 (4.22%)	16	-	3h 51m 34s	36m 4s	1.16	5093.21	3.31	2.04
2002	3680 (11.98%)	102879.9 (1.70%)	2	-	16h 12m 42s	4h 58m 8s	1.72	2057.6	1.34	0.82
2211	2690 (8.76%)	1274.0 (0.02%)	1	-	28m 25s	5h 24m 12s	17.24	25.48	0.02	0.01
2023	2601 (8.47%)	339043.6 (5.62%)	2	-	2d 21h 56m 23s	22h 51m 49s	1.44	6780.88	4.41	2.71

Timed out jobs are particularly costly because they consume compute resources without delivering results.

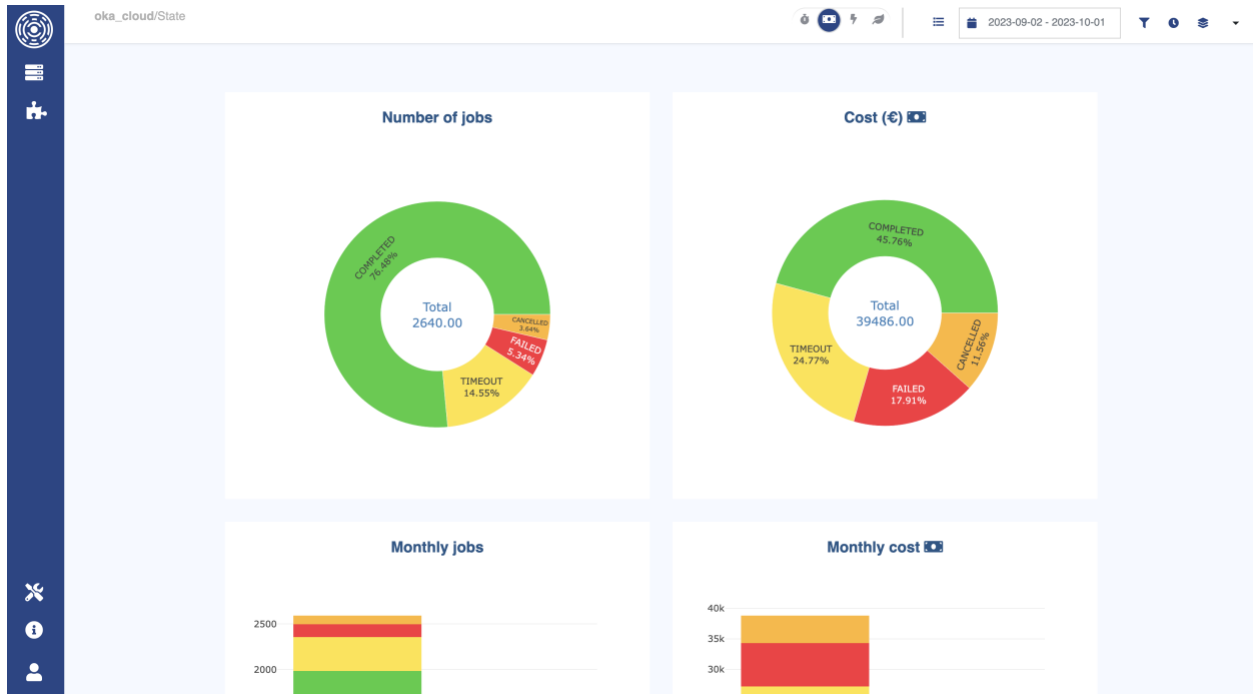
You can go further by analyzing the distribution of job states:

- Completed jobs: delivered results successfully.
- Failed jobs: consumed resources but failed before completion.
- Cancelled jobs: stopped before consuming full resources.
- Timed out jobs: reached their runtime limit without finishing, wasting the allocated resources.

In OKA's State panel, you can display the total workload distribution in both number of jobs and core-hours:



Toggleing on the “Cost” switch can reveal a completely different story: despite “non-completed” jobs representing only 24% of the total amount of jobs, they cost your organization more than 54% of the total spending:



By making good use of the Cost toggle switch, you can quantify the impact of these wasted jobs in real monetary terms. Our example suggests a need for better job scheduling policies, user training, or improved job runtime estimates.

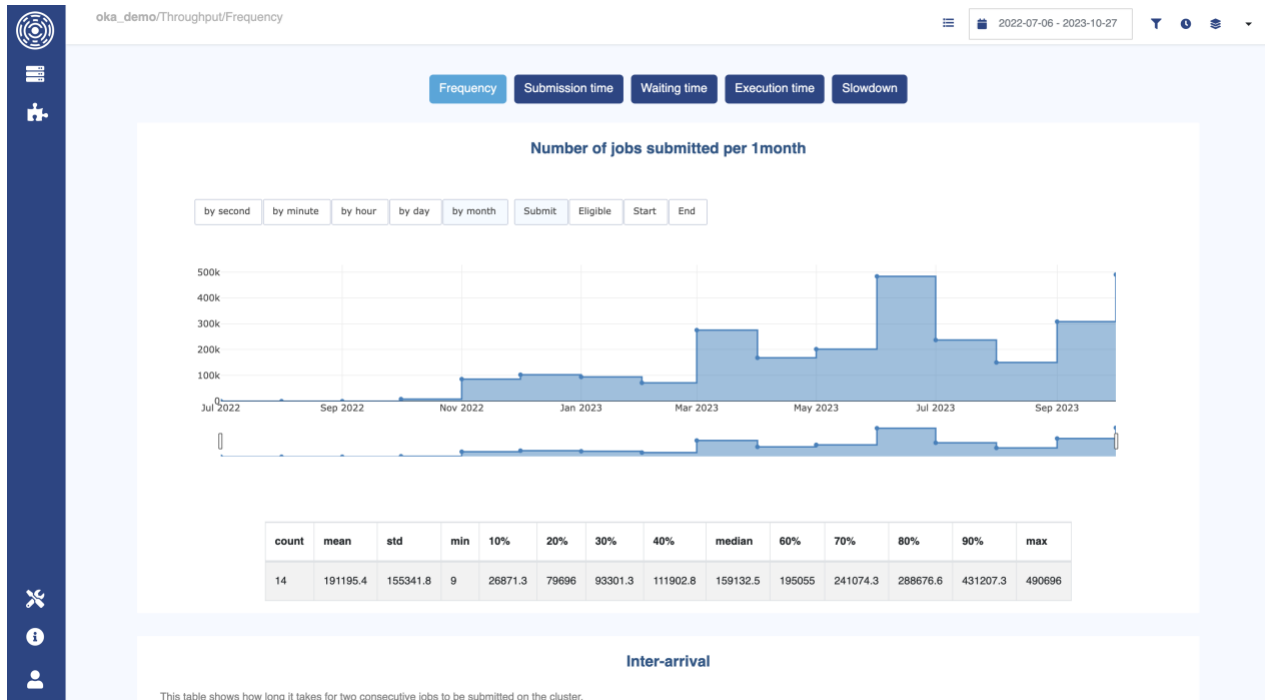
1.1.5) Analyzing peaks, dips and congestion in workload demand

Understanding when workloads peak and dip helps determine if your cluster is properly sized. If demand fluctuates significantly, a mix of fixed and cloud resources might be necessary.

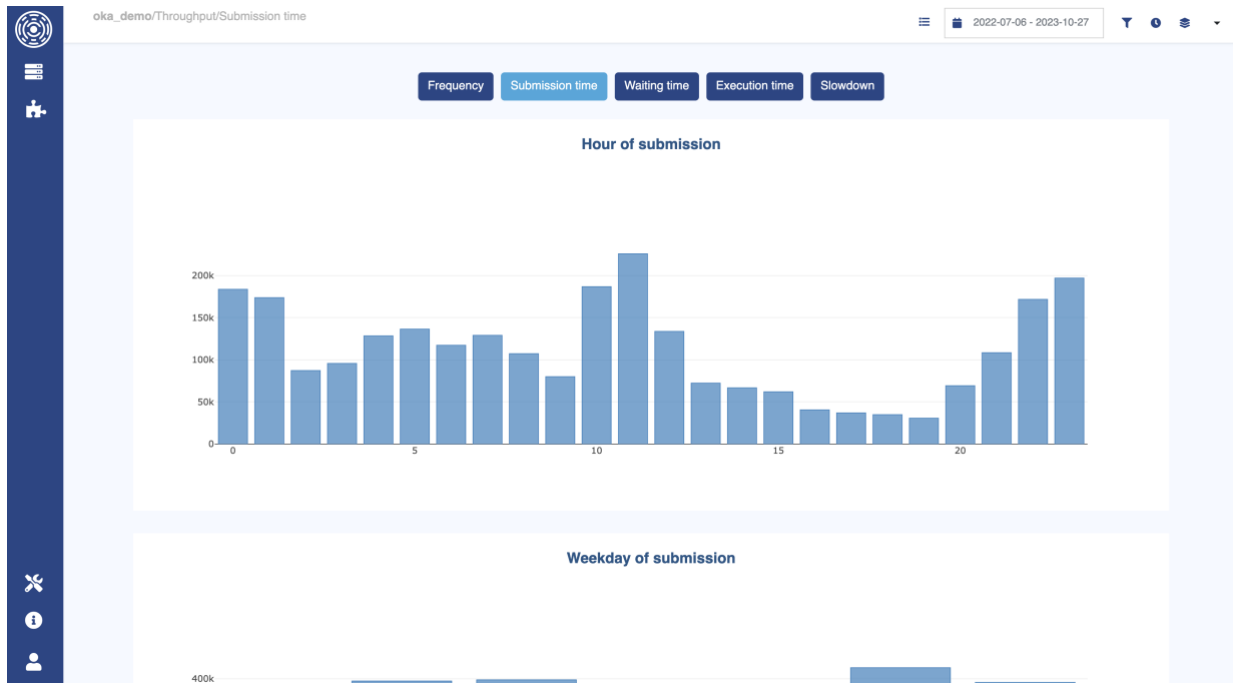
Using OKA’s Load and Throughput panels, you can:

- Pinpoint peak activity periods: identify when compute demand is highest and whether resources are sufficient.
- Analyze queue times: see how long jobs wait before execution, indicating potential bottlenecks.
- Assess overall resource utilization: determine if the cluster is underutilized at certain times.

Here is the cluster's throughput across a long period of time:



And here is the same view, per hour of the day:



For example, your analysis might show:

- Heavy simulation workloads during weekdays, with underutilization on weekends.
- Same pattern for certain hours of the day.
- Frequent job queuing due to resource contention, leading to inefficiencies.
- AI training jobs creating occasional bursts in GPU demand, exceeding available resources.

Additionally, assessing cluster congestion is essential, helping assess whether resources are consistently maxed out. If congestion levels are frequently high, this could justify additional hardware investment or better job scheduling policies.

OKA's Congestion panel provides a comparative look at the cluster's load:



On this instant view, each dot represents a day, and you can see how the cluster's load is distributed in the various load quadrants (Optimal, Acceptable, Congestion, Contention).

Congestion arises when a cluster underutilizes its resources, leaving jobs waiting in queues despite available computing capacity. It simply means that you still have resources that could be used for those jobs in queue but for whatever reason, the cluster does not allocate these jobs to free resources.

On the contrary, **Contention** occurs when the cluster operates at peak capacity, yet fails to meet the surging demand for resources, causing queues to pile up.

This other view shows the evolution of the cluster state over time:



For example, by analyzing different time frames, we can observe whether the cluster is spending increasing amounts of time in a congestion state. This could indicate that resource demand typology is mismatched with availability, leading to longer queue times and reduced throughput: suggesting that a redefinition of existing resource typology is needed.

Additionally, contention may occur when demand typology aligns well with existing resource, but too little of these resources are available: suggesting that an expansion of existing resources is needed.

If congestion and/or contention trends are worsening over months or years, it may signal the need to expand capacity, adjust job scheduling policies, reconsider current mix of resource types available, or optimize workload distribution between on-premises and cloud resources.

1.2) ...to plan for the future

Now that you understand the past better, let's build the future.

By analyzing real usage data and projecting future needs, you can build an infrastructure that is not only more powerful but also better suited to your actual workloads.

2) Choosing the infrastructure: a workload-based approach

With this macro analysis of the present in hand, the next step is to align workloads with infrastructure choices. Should your new cluster maintain the same balance of resources, or should it shift based on evolving needs?

Let's go back to our example:

- CFD simulations: these workloads are CPU-intensive, requiring high-memory nodes with strong interconnects. Their compute demand is relatively stable but generates large amounts of data that must be stored and post-processed.
- Engineering analytics: these jobs require moderate computing power but significant storage and fast access to historical data.
- AI for predictive maintenance: these workloads benefit from GPU acceleration but don't run continuously. They create bursts in demand, particularly when retraining models or processing new sensor data.

Now, let's break down how these insights guide your infrastructure decisions.

2.1) Sizing the cluster(s)

When deciding on the size and structure of an HPC cluster, it's crucial to balance performance, scalability, and cost. A miscalculated approach can lead to wasted resources, excessive waiting times, or an infrastructure that fails to keep up with growing demands.

To define the right cluster size, you need to evaluate:

- Overall compute power needed (based on past and projected workloads).
- Single vs multiple clusters (should you consolidate or separate workloads?).
- Geographical distribution (local vs remote vs cloud resources).
- Scalability over time (how flexible does the infrastructure need to be?).

2.1.1) Defining the total compute power needed

The first step is to assess the total amount of computing resources required to meet workload demands. This means calculating:

- CPU & GPU requirements: how many core-hours and GPU-hours do your workloads consume today? Do you need to accommodate growth?
- Memory & storage needs: what are the RAM and disk space requirements for different job types?
- Network & interconnect speeds: how critical is low-latency communication between nodes?

Looking at historical job data (e.g., job logs, usage patterns) can help identify:

- Baseline usage: the minimum required compute capacity to support day-to-day operations.
- Peak demand: the maximum resources used during the busiest periods (important to avoid bottlenecks).
- Trends over time: growth in job submissions, increased use of AI, or other shifts in workload demands.
- OKA can help provide these insights by analyzing past cluster activity and estimating future needs.

2.1.2) Should you have one large cluster or multiple clusters?

Organizations often debate whether to consolidate all workloads into a single, shared cluster or to split resources across multiple specialized clusters. The right choice depends on:

- Diversity of workloads: if workloads have highly different needs (e.g., CFD vs AI vs data analytics), separate clusters may be more efficient.
- User policies: some organizations prefer to separate internal R&D from production workloads.
- Resource allocation & fairness: in a shared cluster, job scheduling must be carefully optimized to avoid one group monopolizing resources.

Example scenarios:

- A single cluster: best for homogeneous workloads, centralized management, and shared high-speed interconnects.
- Multiple clusters: recommended when workloads have distinct compute/storage needs, or when different departments require separate policies, or when using Cloud compute resources.
- A hybrid model: a large central cluster for general workloads, with smaller clusters (or cloud bursting) for specific use cases.

2.1.3) Where should the compute resources be located?

HPC clusters can be fully on-premises, distributed across multiple sites, or leveraging cloud resources.

Key factors to consider:

- Latency-sensitive workloads (like real-time simulations) benefit from on-prem clusters with high-speed networking.
- Data sovereignty & security requirements may limit cloud options for certain organizations.
- Multi-site deployments can provide redundancy and disaster recovery capabilities.

Common strategies include:

- A single, centralized on-prem cluster: ideal when high-speed interconnects are needed.
- Regional clusters: useful when different teams operate in separate geographic locations.
- Hybrid HPC deployments: a base on-prem cluster with cloud bursting for peak loads.
- All in cloud: a full-on deployment of all workloads onto cloud resources.

2.1.4) Planning for scalability

Clusters need to be scalable to accommodate future growth, new applications, or sudden spikes in workload.

- Modular expansion: design infrastructure so nodes can be added as needed.
- Flexible job scheduling: optimize queue policies to maximize utilization.
- Cloud integration: keep the option open to extend compute power when required.

By making data-driven decisions based on past usage patterns, you can right-size your HPC infrastructure while ensuring flexibility for the future.

2.2) CPU vs GPU: is your balance right?

One of the biggest questions in HPC today is how to distribute resources between CPUs and GPUs. The workload analysis may suggest:

- **CFD simulations remain CPU-heavy**, meaning you will still need a strong CPU foundation in your next cluster. Investing in high-core-count CPU nodes with high memory bandwidth will be crucial.
- **AI workloads, are growing** (they won't remain a minority for long) and we noticed frequent GPU queuing. This suggests that expanding GPU capacity could be valuable, not just for AI but also for potential future hybrid workloads (e.g., AI-driven CFD).
- **Analytics jobs don't necessarily require expensive compute nodes** and could be offloaded to lower-cost hardware or cloud instances to free up resources for compute-intensive tasks.

If AI adoption continues to increase, you may need to explore specialized accelerators (e.g., TPUs for AI inference) or hybrid nodes that support both CPU and GPU workloads efficiently.

2.3) Cloud vs on-prem: optimizing flexibility and cost

A hybrid approach (balancing cloud and on-prem resources) can help match infrastructure to workload patterns. In our current example, we see:

- CFD workloads run at a steady pace and benefit from high-speed interconnects, making on-prem the preferred option. However, peak periods (e.g., when multiple large simulations are submitted) could justify occasional cloud bursting.
- AI training jobs generate spikes in GPU demand, leading to contention. Instead of overprovisioning on-prem GPUs, you could allocate burst capacity to the cloud, ensuring flexibility without overspending on hardware that remains idle between peaks.
- Engineering analytics jobs require large storage but not high-performance computing, making them ideal candidates for cloud-based storage and processing. This avoids overloading high-performance nodes with I/O-heavy but low-compute workloads.

These insights help define a hybrid architecture that optimizes cost, performance, and scalability: keeping latency-sensitive and high-performance workloads on-prem while leveraging the cloud for burst capacity and scalable storage solutions.

2.4) What configuration for your cluster(s)?

Once the size, location, and purpose of the HPC cluster are determined, the next step is defining its technical configuration. The right choices will directly impact performance, efficiency, and long-term adaptability.

2.4.1) Choosing the right node configuration and processor architecture

HPC workloads vary greatly in their compute, memory, and interconnect needs, making node selection one of the most critical decisions.

- CPU vs GPU balance: some workloads are best suited for multi-core CPUs, while others (AI, molecular dynamics, etc.) benefit from GPU acceleration.
- Processor architecture: while x86 (Intel/AMD) has been dominant, ARM-based processors (such as AWS Graviton or NVIDIA Grace) are gaining traction due to power efficiency and cost-effectiveness.
- Memory per node: large simulations or genomics workloads require high-memory nodes, while parallel workloads may be more CPU/GPU bound.

- Interconnects: low-latency, high-bandwidth connections (InfiniBand, NVLink) are crucial for tightly coupled simulations but unnecessary for loosely coupled workloads.

Example scenario:

- CFD simulations often require high-core-count CPU nodes with fast interconnects.
- AI training workloads benefit from GPU-accelerated nodes.
- Data analytics tasks may only require commodity CPU nodes with fast storage.

By analyzing workload patterns, you can customize node configurations to fit your real needs rather than overspending on one-size-fits-all hardware.

2.4.2) Defining the storage & filesystem strategy

Storage performance is just as important as compute power in HPC environments. Selecting the right storage architecture and filesystem ensures that simulations, AI training, and data analysis workflows are not bottlenecked by slow I/O.

- High-speed local scratch storage: fast, temporary storage (e.g., NVMe SSDs) for active workloads.
- Parallel filesystems: distributed storage (e.g., Lustre, BeeGFS, GPFS) optimized for large-scale HPC workloads.
- Object storage: cost-effective storage (e.g., S3, Ceph) for long-term data retention or hybrid cloud workflows.
- Tiered storage approach: mixing fast storage for active jobs with archival storage for older datasets.

Example scenario:

- A CFD engineer submits a job that writes large checkpoint files every few minutes → Needs a parallel file system.
- An AI researcher pulls large datasets for training from long-term storage → Needs object storage with caching.
- A team working on climate simulations shares multi-petabyte data → Needs a scalable parallel filesystem.
- HPC admins should map workload I/O patterns to the right storage tiers to avoid performance bottlenecks.

2.4.3) Ensuring software stack & application compatibility

Hardware selection must be aligned with the software ecosystem to ensure smooth transitions, avoid performance drops, and maximize portability across architectures.

- Application compatibility: ensure key applications support the selected processor architecture (x86, ARM, GPUs).
- Compilers & libraries: check whether existing workloads need specific compilers (Intel, GCC, LLVM) or optimized libraries (CUDA, ROCm, OpenMP).
- Job schedulers: SLURM, PBS Pro, or other scheduling tools must be optimized for the cluster's configuration.
- Containerization & portability: using Singularity or Docker ensures software portability, especially when adopting hybrid cloud models.

Example scenario:

- A research team using legacy CFD software may require x86 nodes for compatibility.
- AI developers using PyTorch or TensorFlow can benefit from ARM-based GPUs like NVIDIA Grace Hopper.
- A multi-user environment may require containerized applications to ensure reproducibility across on-prem and cloud clusters.

By evaluating software requirements early in the planning process, you can avoid costly compatibility issues and ensure their applications make full use of new hardware.

3) Optimizing costs & resource utilization

Building a new HPC cluster is a significant investment, but the real cost extends beyond the initial hardware purchase. Poor resource allocation, inefficiencies in job scheduling, and underutilized hardware can quickly drive up operational expenses. Before finalizing a new cluster configuration, it's crucial to analyze where waste occurs and ensure that resources are sized and allocated properly.

3.1) Improving resource allocation to reduce waste

Regarding the inefficiencies we detected in part 1), the next step is to ensure that resources are properly allocated based on workload needs.

A few practical ways to reduce waste include:

Right-sizing job requests

Many HPC users request more compute time, memory, or GPUs than their job actually needs, leading to inflated queue times and underutilized resources. Analyzing historical job logs can help optimize job submissions and enforce policies to prevent excessive overprovisioning.

Reserving the right hardware for the right job

- Assigning CFD simulations to high-core-count CPU nodes ensures they don't consume GPU resources unnecessarily.
- Allocating AI training workloads exclusively to GPU nodes prevents inefficient use of general-purpose compute nodes.
- Ensuring analytics jobs run on lower-tier compute resources avoids wasting high-performance hardware.

Eliminating bottlenecks in job scheduling

If jobs experience long queue times due to resource shortages, it may be necessary to adjust scheduling priorities, implement preemptible jobs, or introduce resource-based fair-share scheduling to improve efficiency.

Leveraging cloud bursting for peak workloads

If the cluster experiences occasional peaks in demand, spinning up additional cloud nodes can help avoid purchasing unnecessary on-prem hardware that sits idle during off-peak periods.

Promoting user awareness of peaks is also a very relevant direction to consider, especially when clear job submission patterns are visible throughout workweeks, project cycles, etc.

By implementing these workload-aware resource allocation strategies, you can significantly reduce waste and maximize the value of their HPC investments.

3.2) Modeling costs over the next 3–5 years

Beyond immediate resource optimization, planning for long-term costs is essential. A well-structured cost model should factor in:

Infrastructure ownership costs

- Hardware acquisition and maintenance
- Cooling, power consumption, and facility costs
- Staffing and administration

Operational costs

- Software licensing fees (commercial solvers, AI frameworks)
- Storage expansion and data transfer costs
- Network upgrades for increased workloads
- Maintenance costs

Cloud vs on-prem cost balance

For variable workloads like AI training and burstable CFD jobs, estimating cloud costs vs. on-prem capacity can help determine the best long-term strategy. If cloud reliance becomes excessive, bringing more capacity in-house may be more cost-effective.

Scaling considerations

If workloads are growing rapidly, planning for modular expansion—adding nodes or storage as needed—ensures that the cluster remains cost-efficient without requiring a full rebuild.

Example:

- If AI workloads are projected to double within 3 years, investing in additional GPU nodes today may be more cost-effective than frequent cloud expenditures.
- If CFD simulations are growing, investing in high-memory CPU nodes might be a priority.
- If analytics workloads are becoming more storage-intensive, scaling parallel filesystems should be factored into cost projections.

By systematically analyzing inefficiencies, optimizing resource allocation, and modeling future costs, HPC teams can ensure that their next cluster delivers maximum performance at the lowest possible cost—without compromising scalability or flexibility.

4) Future-proofing: long-term planning & sustainability

HPC infrastructure is a long-term investment, but the pace of technological change means that what works today might not be the best fit in a few years. Ensuring that a new cluster remains relevant over its lifespan requires future-proofing—building flexibility into hardware, workload management, and sustainability strategies.

4.1) Choosing a future-proof hardware strategy

One of the biggest risks when planning a new cluster is locking into hardware that may become outdated or suboptimal before the system reaches the end of its lifecycle. This applies not only to compute but also to storage, networking, and accelerators.

A few key considerations when selecting hardware:

CPU and GPU roadmap alignment

With major shifts in processor architectures (Intel vs. AMD vs. ARM), organizations need to ensure they are investing in hardware with long-term software ecosystem support. If AI workloads are expected to grow, choosing a GPU vendor with strong long-term driver and framework support is essential.

Scalability and modular expansion

Rather than building one large monolithic system, designing a cluster with modular, scalable components allows for incremental upgrades. Can additional GPU nodes be added without redesigning the entire system? Can the storage solution scale with increasing dataset sizes?

Interoperability and hybrid HPC readiness

Many organizations are moving toward hybrid HPC environments. Ensuring compatibility with cloud HPC instances, multi-vendor architectures, and shared job scheduling frameworks prevents the cluster from becoming a technical dead-end.

4.2) Assessing energy efficiency & sustainability

Sustainability is no longer just an afterthought—it has become a critical decision factor for many HPC buyers. In some regions, regulatory requirements now mandate reporting and reducing energy consumption, making energy-efficient HPC a necessity.

Key areas to evaluate:

Current cluster power consumption

Measuring energy usage per job, per node, and per workload type helps identify areas for improvement. OKA's analysis features can provide insights into which workloads are consuming the most power and whether resources are being used optimally.

Cooling and infrastructure efficiency

Traditional air cooling is reaching its limits for high-density clusters. Liquid cooling, immersion cooling, and energy reuse strategies (such as repurposing waste heat) are increasingly important for keeping power costs under control.

Cloud sustainability trade-offs

While cloud providers offer HPC scalability, they also consume large amounts of energy. Choosing cloud regions powered by renewable energy and evaluating cloud compute efficiency vs. on-prem efficiency can help balance sustainability goals.

Optimizing job scheduling for power efficiency

By implementing power-aware scheduling, clusters can allocate jobs to nodes that maximize performance per watt—reducing overall energy consumption while maintaining throughput. Other strategies can be implemented as well: node electrical shutdowns, reduction of GPU/CPU operating frequencies, etc.

4.3) Cost-efficient long-term operations

Beyond initial capital investment, the real cost of HPC comes from ongoing operations—power, maintenance, software licensing, and cloud expenditures.

To maintain long-term efficiency:

Regularly reassess workload placement

If AI workloads grow and GPUs become a bottleneck, adjusting the CPU-to-GPU ratio may be necessary. If cloud spending becomes excessive, migrating some workloads back on-prem could be cost-effective.

Flexible licensing models

Commercial software can be a significant cost factor. Choosing scalable, usage-based licensing can prevent overpaying for underutilized software.

Automated monitoring and forecasting

Using OKA's long-term trends analysis, organizations can predict workload evolution and proactively scale resources before performance bottlenecks emerge.

By choosing adaptable hardware, optimizing energy efficiency, and planning for long-term cost control, you can build an HPC infrastructure that remains efficient, scalable, and sustainable well into the future.

5) Conclusion: from analysis to action

Choosing your next HPC cluster is never just about hardware. It's about understanding your workloads, anticipating change, and building something that can evolve with you. Whether you're focused on performance, cost control, energy impact—or all three—data-driven planning is your best ally. With tools like [OKA](#) and the right approach, you can turn infrastructure planning into a smart, forward-looking investment.

Do you want help assessing your current cluster? Let's talk—this is [what we do](#).