Cost and performance modeling for Earth system data management and beyond

Conference or Workshop Item

Accepted Version

Updated

Lüttgau, J. and Kunkel, J. (2018) Cost and performance modeling for Earth system data management and beyond. In: HPC I/O in the Data Center Workshop, 24-28 June 2018, Frankfurt, Germany, pp. 23-35. (ISBN 9783030024642) Available at http://centaur.reading.ac.uk/79587/

Published version at: https://doi.org/10.1007/978-3-030-02465-9_2

Publisher: Springer

Cost and Performance Modeling for Earth System Data Management and Beyond

Jakob Lüttgau and Julian Kunkel

Deutsches Klimarechenzentrum, Hamburg 20146, Germany
{luettgau,kunkel}@dkrz.de

Abstract. Current and anticipated storage environments confront domain scientist and data center operators with usability, performance and cost challenges. The amount of data upcoming system will be required to handle is expected to grow exponentially, mainly due to increasing resolution and affordable compute power. Unfortunately, the relationship between cost and performance is not always well understood requiring considerable effort for educated procurement. Within the Centre of Excellence in Simulation of Weather and Climate in Europe (ESiWACE) models to better understand cost and performance of current and future systems are being explored. This paper presents models and methodology focusing on, but not limited to, data centers used in the context of climate and numerical weather prediction. The paper concludes with a case study of alternative deployment strategies and outlines the challenges anticipating their impact on cost and performance. By publishing these early results, we would like to make the case to work towards standard models and methodologies collaboratively as a community to create sufficient incentives for vendors to provide specifications in formats which are compatible to these modeling tools. In addition to that, we see application for such formalized models and information in I/O related middleware, which are expected to make automated but reasonable decisions in increasingly heterogeneous data centers.

Keywords: Storage, Data Management, Earth Systems, TCO, Cost

1 Introduction

As scientists are adapting their codes to take advantage of the next-generation exascale systems, the I/O bottleneck is becoming a major challenge[15][10][6] because storage systems struggle to absorb data at the same pace as it is generated. Large scale earth system simulations and workflows, as used in numerical weather prediction (NWP) and climate modeling, are especially I/O intensive. It is anticipated, that a growing proportion of the total budget for a supercomputer will be dedicated to storage systems. This raises the need to better understand the trade-offs associated with different technologies in short-, mid- and long-term perspectives. Technologies and applications are constantly influencing the research and development focus in one another.

The final publication is available at Springer via https://doi.org/10.1007/978-3-030-02465-9_2

1.1 Data Growth and Access Requirements

From a scientific perspective, multiple factors contribute to increasing data volumes and velocities, soon requiring systems which are able to routinely handle exabytes of data. On the one hand Moore's Law and distributed computing allow generating more data in total, for example, by using higher resolution models or increasing the number of members in ensemble simulations. On the other hand sensors and data loggers have become more affordable so that more observational data is being gathered. This also shows in an increasing number of active satellites used for remote sensing which also feature higher resolutions and more instruments. Besides these two obvious trends, data sets are expected to be used interdisciplinary on a more regular basis, so that additional users coming from other scientific domains will be requesting data[7]. Simulations will need to read and write more data when being coupled to use and produce the data products from and for other sciences. As a result estimates for global and local archive capacity requirements are rising[16]. For example, the Climate Model Intercomparison Projects (CMIP) required a total of 35TB to store all CMIP3 data, while CMIP5 already required 3108TB. CMIP6 and CMIP7 are expected to experience comparable increases to capacity requirements now in the exabytes.

1.2 Existing and Emerging Technologies

From a systems perspective, it is sometimes hard to make predictions due to the impact of economic factors, technological breakthroughs or natural disasters which have shown profound impact in the past[8]. None the less, it is necessary to factor in these trends into the choice of architecture and the design of cost effective data centers in the future.

As vendors are growing their production capacities, NAND-based storage technologies are expected to become affordable enough to be feasible as an alternative to disk, despite its limited write endurance. Unfortunately, NAND-based memory's areal capacity for single-level cells can not be further improved, which drives the development of multi-level cells, 3D NAND and high-bandwidth memory – but those, while providing high throughput come with latency penalties for small random I/O. Finally, non-volatile memory technologies for burst buffers and network-attached memory (NAM) are being researched for integration into the next-generation of supercomputers[5].

As of 2018, the vast amount of online storage is provisioned using high performance, but expensive disks based storage systems. Object storage, instead of parallel file-systems, promises to offer a cost-effective alternative. Unfortunately, there is also a disconnect between how business and industry v. NWP and climate applications are using compute infrastructure, limiting direct benefit of commoditization for many HPC applications. Assuming more heterogeneous data centers, next-generation storage systems are likely requiring software stacks that play well with a variety of different interfaces to exploit storage technologies with new semantics.

Long-term archives and cold storage are typically realized using large automated tape library systems. The European Centre for Medium-Range Weather Forecasts (ECMWF) and the German Climate Computing Center (DKRZ) are among the institutions with the largest scientific archives world wide[9].

For NWP and climate users, upcoming infrastructure might be effected mainly by the following two modes of operation. Compute sites may specialize such that one site will focus on providing the required infrastructure to accommodate simulations, while another site might focus on infrastructure which is optimized for analysis tasks. A second model is collocation, where multiple services are consolidated into an external data center and potentially cloud providers. Both approaches can be observed within the climate and NWP communities and each approach comes with a number of benefits and drawbacks.

1.3 Addressing Domain Scientists and their Workflows

Specialized centers may benefit from simplifications and assumptions that can be made about the user base and their workloads. For example, it is possible to relax security considerations when no personal data needs to be handled, which in turn can provide performance advantages. Specialized data centers in climate and NWP are operated by DKRZ, ECMWF, UK Met and others. Multipurpose data centers on the other hand benefit from economies of scale and they can make use of workload sharing, though in practice this is often not possible because applications are not yet designed with this in mind. But cloud environments already make extensive use of workload sharing. Fortunately, operating a larger system usually does no require a proportional increase in staff. Future solutions will need to fit current workflows for at least a intermediate period, while adding a number of advanced features for adoption by application developers.

More tools for automation will be required to operate even larger systems because component failures are expected to be more common in exascale systems. But also to perform common optimization transparently or automatically. This might allow to reduce staff on the one hand, but also frees up experts to focus on non-routine problems. Automation is also a prerequisite to realize data handling policies and service level agreements at scale. Which, in turn, enables resource sharing and allows prioritizing critical or rewarding well behaving applications. Finally, storage systems need to be more customizeable, on the one hand to support user specific workflows and automation which are not provided by default by the storage systems, but also to work well in heterogeneous architectures.

The remainder of this paper is structured as follows. Related work is presented in Section 2. Section 3 introduces a hierarchical modeling approach. Section 4 discusses the coarse model in detail and how it can also be used to model resilience and performance. In Section 5 considerations for the most important parameters of compute and I/O nodes are discussed. Section 6 analyses cost for the current DKRZ system and the impact of alternative deployment strategies for relatively new or merely anticipated technologies. Section 7 briefly explains how I/O middleware in the future may exploit cost models to improve performance or to reduce cost. The results of the paper are summarized in Section 8.

2 Related Work

Related work can be grouped into approaches analyzing and modeling system characteristics and storage systems on the one hand, and standardization on the other. Multiple approaches model HPC systems by implementing discrete event simulation (DES) of queuing systems, as this allows for more complex models and time dependency. Modeling of computer systems has a long history[18]. In the last decade, the analysis of supercomputers and storage gained attention. In the CODES Project, multiple use cases for storage have been implemented[14], which are using a simulation framework the Rensselaer's Optimistic Simulation System (ROSS)[4]. A similar effort is the Structural Simulation Toolkit (SST)[2]. These efforts are mostly focused on finding new architectures to cope with exascale workloads and do not consider initial and operation cost specifically. In section Section 4 we also look at compatibility to a fine grained approach[13] which uses DES to determine cost and quality of service for hierarchical storage system including tape system in data centers. For the purpose of standardization and communication with users these models can be too fine-grained.

Modeling individual storage systems covers various characteristics like energy consumption, resilience, and performance. Llopis et al.[17] explored empiric means to determine power consumption for individual components with a focus on power consumption within the storage and I/O data paths. Various studies model resilience based on the distribution of data across storage devices and strategy; for example, in [21], resilience depending on RAID levels is investigated and visualized in 2D heatmaps depending on error rates of memory and storage. In [19], a formal method of investigating resilience depending on data replication strategy and hardware is described and explored on several use cases.

The topology of storage systems and their characteristics is documented by most data centers and experts. While they can typically be understood by experts and serve the purpose of communication, the representations vary significantly in terms of abstraction, detail and style. An attempt to document the I/O path in a more standardized fashion has been made in [12]. A recent approach to collect the topology of data centers and their hardware characteristics is the Data Center List on the Virtual Institute for I/O[1]. It provides a template for different hardware components that can be filled. While this could standardize the descriptive nature of HPC systems, it does not allow to derive conclusions.

There have been various attempts to use and extend UML diagrams for performance prediction, mostly for use in computer aided software engineering. For example, UML activity diagrams are candidates to apply analysis techniques from PetriNets [20] and an assessment of parallel programs is described in [3]. The analysis of system performance is not covered as it depends on the use case.

Performance of parallel file systems have been subject to modeling, and for most file systems at least one attempt has been made. Examples are models for PVFS2 [1] and Lustre [22]. Both use a graphical representation and are based on

[1] `https://www.vi4io.org`

various system parameters as well as file system-specific configurable parameters like stripe size.

The work in this paper aims to provide abstractions that ease communication between experts and deriving conclusions while relying on a hardware model independent of the storage system and its tunable parameters. As such, the approach outlined in this paper aims to provide a performance estimate that is easy to understand by non-experts.

3 Cost Modeling

With all of the challenges and trends outlined in Section 1 and Section 2 it is apparent that there is a demand for better tools to conduct cost modeling of storage infrastructure in data centers. A common challenge to cost modeling is, that it is often only possible to make assumptions and best estimates. This may be due to changing release road-maps of vendors, due to unreliable technological breakthroughs but also because workloads might differ on a new systems as bottlenecks or user behavior are changing. Unfortunately, highly granular models quickly become not only overwhelmingly complex to maintain but also get prohibitively expensive to compute. Yet, for novel architectures it is not possible to turn to empiric data which would allow to simply measure the emerging behavior of a complex system. As it is not easy to find a balance here, a hierarchical approach is proposed. Starting from a coarse grained model, individual sub-components can use more detailed models where further insight is needed.

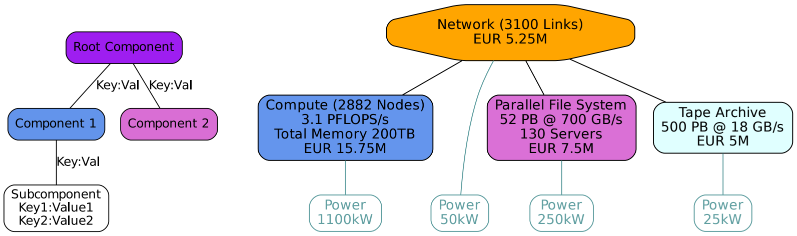
The coarse grained model would be covering relevant components and related metrics for a data center as well as an abstract workload description and optimization strategies related to system technology and workload organization. The goal of the model is to provide a heuristics to quickly determine promising combinations of data centre layouts and their associated costs. It is not our aim to provide a cent-accurate model. This level of the model ignores temporal and spatial factors of the workload runtime behavior.

For additional insight, parts of a coarse grained model can be refined with more detailed models. For example, by mimicing workload execution using DES and workload traces in combination with the actual data center topology, taking temporal behavior into account (see Section 2 and [13]). This way one can narrow down on uncertain areas in the general model. Typically, one might implement a fine grained model for promising coarse grained model or for an existing system to get further insight for optimization.

Ideally, using these models it should be possible to provide a workload mix, e.g., the behavior of multiple typical user and estimate the inherent costs, performance and required fault tolerance. While this paper focuses on data center cost, performance and cost considerations are intimately related to each other. It allows to explore different data center designs in respect to storage given a fixed budget or required features for hardware and software.

4 Coarse Grained Model

This section introduces the coarse grained modeling considerations in more detail. In the coarse grained model a graph of components is assumed which models also how components relate to each other. The graph of components is the foundation to compute different emerging properties. Graphs for an abstract case and a cost example are shown in Figure 1. Component 1 and 2 have dependencies to a root component, and the Subcomponent to Component 1. Edges and components can hold key value pairs which describe the characteristics and which also allows to add custom annotations. The cost example features cost information for the major components and also includes specification details such as performance or annual power consumption. This approach is flexible to model system characteristics beyond cost such as resilience and performance.



(a) Abstract Model (b) Simple Cost Model for a Data Center (DKRZ)

Fig. 1: Example for relationships and characteristics for system components.

4.1 Resilience Model

To model resilience, components need to feature failure metrics, for example, derived using empirical methods. Error propagation of failures follows a directed graph. This also allows to account for cascades as well as mitigation measures.

A simple example to model resilience is shown in Figure 2. A failing data center power supply may result in subsequent switch failures, but selected subsystems may be kept operational for 20 minutes due to the presence of an uninterrupted power supply (UPS). Mitigation strategies usually depend on redundancy, for example, replication of data usually adds costs but may either reduce or improve performance as can be the case for RAID systems. Besides topological relationships, annotations allow to associate components with common error measures such as mean-time between failures (MTBF), or mean-time to recovery (MTTR). To allow calculating reliability metrics of parent components, error metrics for sub-components should be independent. Mitigation strategies can become rather complex, and it is not always obvious if a architecture decisions mainly serves as a resilience or a performance feature. Similarly, an uninterrupted power supply might be deployed to ensure high availability, but could also only serve to shutdown a system into a consistent state.

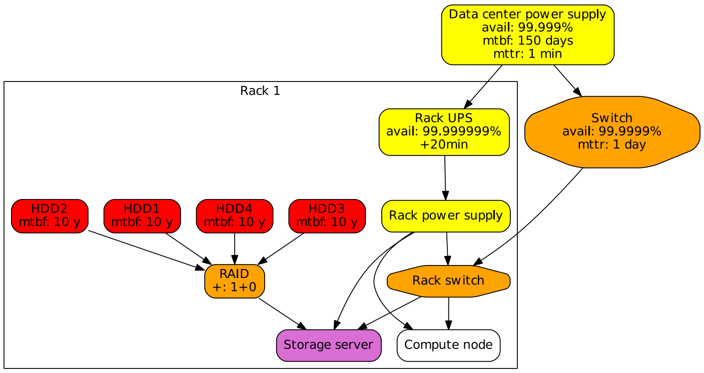


Fig. 2: Component dependency graph to model resilience.

4.2 Performance Model

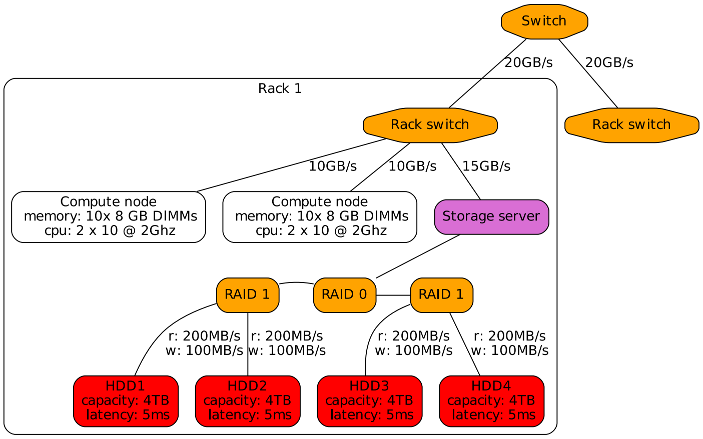


Fig. 3: Component dependency graph to model performance.

The performance model also uses a graph based description, again using annotation for relevant components featuring throughput and latencies. Using the hardware graph it is easy to determine theoretic peak performance for individual components. But also in more complex scenarios, it is possible to gauge the performance that can be obtained from using components in parallel.

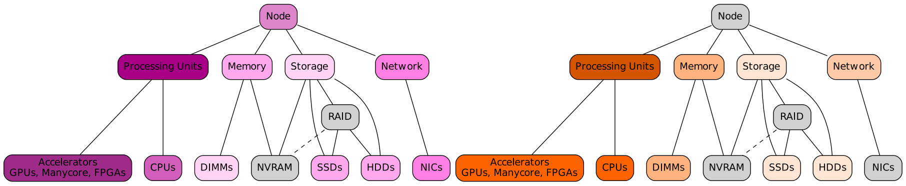
Figure 3 shows an example graph with performance metrics for compute nodes, network and storage media added. For example, it is easy to see that node to node communication can not exceed bandwidths of 10GB/s, while the storage server will be happy to handle incoming bandwidths of up to 15GB/s. Unlike networks, storage media require a more granular approach to account for different transfer rates for read or writes. The throughput for paths to the network may be simply defined by $max(edge0_{throughput}, ..., edgeN_{throughput})$. Similarly the latency is the accumulation of latencies attached edges and nodes on the path $\sum_{item}^{path} item_{latency}$.

Some components such as disks are commonly used in RAID arrays or another combination. It is not always useful to model this complexity explicitly even though the theoretical dynamics are well understood and can be abstracted as described in [12]. In a RAID 1+0 group, for example, the performance of the RAID group is more relevant than the performance of the individual HDDs.

5 Model Considerations for Common Subcomponents

As the approach outlined in Section 3 assumes components and sub-components, this section discusses the most important building blocks for cluster systems in more detail. In particular compute nodes Section 5.1 and I/O nodes Section 5.2 affect the cost and the performance of a system. For both a breakdown by sub-components is provided to illustrate impact on cost and power consumption.

5.1 Compute Nodes

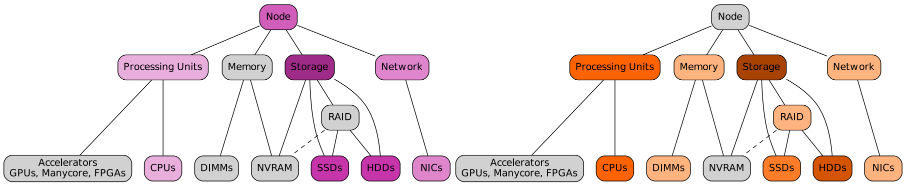


(a) cost (b) power

Fig. 4: Cost and power footprint by component for compute nodes. A darker shade represents a larger share relative to the total node configurations.

Compute nodes come in various configurations depending on the tasks most commonly performed. For climate simulation, nodes usually require substantial amount of compute power and memory in combination with a low latency network. Other use cases, such as visualization or increasingly big data and machine learning applications, may be less dependent on synchronous communication but make use of accelerators. Figure 4 illustrates the initial cost as well as the power consumption per sub-component. CPUs, GPUs and potentially FPGAs determine how fast data data can be processed or generated. They also account for most of the power consumption of the node. If there is spare processing power, it may be be invested into data reduction or more intelligent data handling. The system memory affects the problem size that can be held for quick access on the node. More main memory allows to improve storage performance by use of caching. However, memory is usually contended, for exmaple by network components that require buffers. Nodes may feature storage, which is local to the node. Usually, node local storage is considered too slow for large amounts of data in comparison to PFS/Object Storage. There is potential for this to change as node local NVRAM and burst buffers become more affordable. Network interface cards of a node determine how fast nodes may communicate with each other, but also how fast data can be drained away from the compute nodes, e.g., when writing snapshots. The network also affects how quickly a compute node, which needs to load parts of datasets or shared libraries first, can start to perform useful work.

5.2 I/O Nodes



(a) cost (b) power

Fig. 5: Cost and power footprint by component for I/O nodes. A darker shade represents a larger share relative to the total node configurations.

Scalable storage systems often feature different types of I/O nodes, each subsystem specialized to handle a different class of requests. Figure 5 illustrates the main contributors to initial cost as well as the power consumption in I/O nodes. In I/O nodes processors determine the number of requests that can be handled by a single node. Increasingly, GPUs may be used for data reduction and other in-transit transformations. I/O nodes typically feature considerate amounts of memory, for use as a quick cache layer, but most importantly they accommodate a large amount of disks or SSDs, with data of objects or files strided across multiple devices for fault tolerance and higher performance. In many cases, the storage devices are bundled in so called JBODs which then are connected to the I/O nodes, while an I/O node itself has no storage devices. I/O nodes in HPC systems, commonly use advanced interconnects in fail-over configurations. Depending on the application this maybe high-bandwidth for data storage or low-latency interconnects for metadata access.

Metadata Handlers and Targets: Parallel file systems provide dedicated metadata targets optimized to perform many I/O operations. Metadata servers commonly utilize different storage media than data targets. For example, they often have faster and more expensive solid state disks, and may be candidates for storage class memory (SCM).

Data Handlers and Targets: Data targets are configured for capacity and high throughput. Data targets may feature a lot of memory for caching but the cost is dominated by the amount of hard drives. For HPC systems, usually larger reads and writes are observed, for database systems also the data targets might profit substantially from the usage of SSDs. RAID controllers can also be a cost factor but software-based RAID is becoming very popular for being more flexible.

6 Cost Study for alternative Deployments

This section discusses alternative architecture deployments for upcoming data centers. As described in Section 1 research institutions tend to organize compute and storage capabilities either by specializing at different locations or by collocating and consolidating different services into a single site, as is the case with DKRZ. In this paper only the summary of a larger case study is presented, a more detailed report[11] is available separately.

The DKRZ system (Mistral) operates 3.300 compute nodes in a 3.1 petaflop compute cluster attached to a 52 Petabyte disk based PFS distributed across over 10.000 disks. Besides online storage there is also an archive with capacity of up to 500 petabyte on tape (more for next-generation LTO tapes). In the current deployment I/O related investments are 7.5M Euro for two Lustre based PFS, 5.25M Euro for the network and 5M Euro for the archive. Figure 1 (b) illustrates this setup using the graph approach introduced in Section 4. Together with Table 1 this preserves the relationship between components while allowing to use additional visual cues. With these numbers as a baseline for each subsystem and a total budget of roughly 39M Euro, it maybe interesting to see what half the storage budget might achieve given current technology. Table 1 compares two alternative deployment scenarios and offers a breakdown of cost, performance and power consumption including the factor with respect to the actually deployed system to gauge the impact on each subsystem for different metrics.

The first scenario, not diverting from the topology of the actual deployment, explores the potential trade-off when altering the ratio of offline storage to online storage. The general motivation for this scenario being that timely data staging from tape to PFS might preserve quality of service at lower cost. This would, however, require transparent automated staging mechanisms which integrate with batch scheduling system like Slurm or workload specifications. The example calculation does not take licensing costs into account, and somewhat optimistically assumes performance and power consumption scale linearly. Finally, any remaining budget is used to procure additional compute capabilities, which is responsible for the higher overall power consumption of this scenario.

In a second scenario we use object storage instead of a parallel file system. Assuming state of the art hardware this has a performance penalty, as the throughput performance drops to about a third of the original system. Yet, using only half the budget it is possible to provide about the same capacity and also to conserve energy. Again, as the remaining budget is spent on additional compute, the overall power consumption of the system is higher than the original system.

	Mistral	Scale-down PFS, spent leftovers on compute		Switch to Object Storage, leftovers spent on compute	
Characteristics	Value	Factor	New value	Factor	New Value
Performance	3.1 PF/s	1.17	3.6 PF/s	1.19	3.7 PF/s
Nodes	2882	1.17	3370	1.19	3430
Node performance	1.0 TF/s				
System memory	200 TB	1.17	234 TB	1.19	238 TB
Network links	3100	1.12	3450	1.15	3565
Storage capacity	52 PB	0.5	26 PB	0.9	47 PB
Storage throughput	700 GB/s	0.5	350 GB/s	0.375	262 GB/s
Storage servers	130	0.5	65	0.75	98
Disk drives	10600	0.5	5300	0.74	7800
Archive capacity	500 PB				
Archive throughput	18 GB/s				
Compute costs	15.75 M EUR	1.17	19.53 M EUR	1.24	19.53 M EUR
Network costs	5.25 M EUR	1.10	6.04 M EUR	0.98	5.15 M EUR
Storage costs	7.5 M EUR	0.5	3.75 M EUR	0.5	3.75 M EUR
Archive costs	5 M EUR				
Building costs	5 M EUR				
Investment	38.5 M EUR		38.41 EUR		38.43 M EUR
Compute power	1100 kW	1.19	1290 kW	1.10	1309 kW
Network power	50 kW				
Storage power	250 kW	0.5	125 kW	0.75	188 kW
Archive power	25 kW				
Power consumption	1.20 MW		1.49 MW		1.57 MW

Table 1: Summary of the expected impact of two alternative deployment scenarios. Comparison of Mistral as installed, a deployment with a reduced disk system and a deployment using object storage instead of a file system.

Burst buffers promise to compensate for some of the lost throughput performance of the previous two scenarios. Unfortunately, only a number of experimental commercial products for burst buffers are available at the moment and price estimates may be subject to non-disclosure agreements. Non-volatile memory maybe integrated into compute nodes, which can be attractive for data locality, e.g. in case of node failures, which potentially takes load off the network. Alternatively burst buffers could be integrated similar to network attached memory (NAM), which allows to dynamically allocate remote memory which provides a high degree of flexibility.

Given the recent rise of cloud technologies and popularity of object storage, some might anticipate a possible displacement of parallel file systems and tape archive in future data centers. The integration of clouds, can reduce some burdens on application developers and add flexibility to applications, but current rates charged by cloud service providers do not justify moving away from on-premise deployments.

7 Application in Cost-Aware I/O Middleware

Besides better understanding the relationship of cost, performance and resilience when procuring and designing new systems, these models can also support I/O middleware to make better decisions. In the ESiWACE[6] project, the *Earth System Data Middleware* (ESDM) is developed. It is designed to address multiple I/O challenges simultaneously, such as: 1) automatic seperation of data and metadata when using data description frameworks 2) the distributing data accross different storage tiers and services in the data center using a description of the site configuration and 3) adaptive I/O strategies and data representations depending on anticipated workflows and service level agreements. A prerequisite for such automatic optimization is a capable software infrastructure on the one hand, but more importantly, this requires adequate and light-weight models to derive reasonable decisions and policies tunable to a data centers demands.

8 Summary

Modeling data center cost and performance is a complicated task but by approaching cost modeling in a systematic way it is possible to reuse and adapt existing models more easily. The paper presented a methodology and considerations relevant for modeling data centers intended for the climate and NWP community. In Section 1 domain and technological trends are discussed which are then accommodated in the hierarchical model presented in Section 3. It was demonstrated how to construct a coarse grained model not only for cost but also, with some limitations, for resilience and performance. In a case study of the DKRZ system, the paper briefly explored how alternative deployments may allow to prioritize cost reductions or performance improvements. In future work we want integrate cost models with I/O middleware such as with ESDM, to allow addressing the challenges of multiple stakeholders.

Acknowledgment

The ESiWACE project received funding from the EU Horizon 2020 research and innovation programme under grant agreement No 675191.

References

1. Performance Evaluation of the PVFS2 Architecture. Napoli, Italy
2. SST Simulator - The Structural Simulation Toolkit, `http://sst-simulator.org/`
3. Arjona, J.O.: Using UML State Diagrams for Modelling the Performance of Parallel Programs. Computación y Sistemas **11**(3), 199–210 (2008)
4. Carothers, C.: ROSS: Rensselaer's Optimistic Simulation System (Nov 2017), `https://github.com/carothersc/ROSS`
5. DEEP Projects, `http://www.deep-projects.eu/`

6. ESiWACE: Centre of Excellence in Simulation of Weather and Climate in Europe, https://www.esiwace.eu/
7. ExtremeEarth, http://www.extremeearth.eu/
8. Fontana, R.E., Decad, G.M., Hetzler, S.R.: The impact of areal density and millions of square inches (MSI) of produced memory on petabyte shipments of TAPE, NAND flash, and HDD storage class memories. In: 2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST). pp. 1–8. IEEE (2013), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6558421
9. HPSS Collaboration: List of Sites (2018), http://www.hpss-collaboration.org/customersT.shtml
10. Intel, The HDF Group, EMC, Cray: Fast Forward Storage and I/O (Jun 2014)
11. Jakob Luettgau, Julian Kunkel, Jens Jensen, Bryan Lawrence: ESIWACE D4.1 Business Model with Alternative Scenarios. Tech. rep., https://www.esiwace.eu/results/deliverables/d4-1-business-model-with-alternative-scenarios
12. Kunkel, J.M., Ludwig, T.: IOPm - Modeling the I/O Path with a Functional Representation of Parallel File System and Hardware Architecture. In: 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (2012). https://doi.org/10.1109/PDP.2012.13
13. Luettgau, J., Kunkel, J.: Simulation of Hierarchical Storage Systems for TCO and QoS. In: Kunkel, J.M., Yokota, R., Taufer, M., Shalf, J. (eds.) High Performance Computing. pp. 132–144. Springer International Publishing, Cham (2017)
14. Mubarak, M., Carothers, C.D., Ross, R., Carns, P.: Modeling a Million-Node Dragonfly Network Using Massively Parallel Discrete-Event Simulation. In: 2012 SC Companion: High Performance Computing, Networking Storage and Analysis. pp. 366–376 (Nov 2012). https://doi.org/10.1109/SC.Companion.2012.56
15. NEXTGenIO: Next Generation I/O for the Exascale, http://www.nextgenio.eu/
16. Overpeck, J.T., Meehl, G.A., Bony, S., Easterling, D.R.: Climate Data Challenges in the 21st Century. science **331**(6018), 700–702 (2011)
17. Pablo Llopis, Dolz, M.F., Blas, J.G., Isaila, F., Heidari, M.R., Kuhn, M.: Analyzing the energy consumption of the storage data path. The Journal of Supercomputing **72**(11) (Nov 2016). https://doi.org/10.1007/s11227-016-1729-4, http://link.springer.com/10.1007/s11227-016-1729-4
18. Pentzaropoulos, G.: Computer performance modelling: An overview. Applied Mathematical Modelling **6**(2), 74–80 (1982)
19. Pereverzeva, I., Laibinis, L., Troubitsyna, E., Holmberg, M., Pöri, M.: Formal Modelling of Resilient Data Storage in Cloud. In: International Conference on Formal Engineering Methods. pp. 363–379. Springer (2013)
20. Tribastone, M., Gilmore, S.: Automatic Extraction of PEPA Performance Models from UML Activity Diagrams Annotated with the MARTE Profile. In: Proceedings of the 7th International Workshop on Software and Performance. pp. 67–78. WOSP '08, ACM (2008). https://doi.org/10.1145/1383559.1383569, http://doi.acm.org/10.1145/1383559.1383569
21. Zhang, Y., Myers, D.S., Arpaci-Dusseau, A.C., Arpaci-Dusseau, R.H.: Zettabyte reliability with flexible end-to-end data integrity. pp. 1–14. IEEE (May 2013). https://doi.org/10.1109/MSST.2013.6558423, http://ieeexplore.ieee.org/document/6558423/
22. Zhao, T., March, V., Dong, S., See, S.: Evaluation of a performance model of Lustre file system. In: ChinaGrid Conference (ChinaGrid), 2010 Fifth Annual. pp. 191–196. IEEE (2010)