JOURNAL OF INTERNATIONAL CRISIS AND RISK COMMUNICATION RESEARCH
ISSN: 2576-0017
2022, VOL 5, NO 2

AI Clusters and Elastic Capacity Management: Designing Systems for Diverse Computational Demands

Ravi Kumar Vankayalapati[1], Rama Chandra Rao Nampalli[2]

1. Cloud AI ML Engineer, ravikumar.vankayalapti.research@gmail.com, ORCID : 0009-0002-7090-9028
2. Solution Architect, CO-80134, nampalli.ramachandrarao.erp@gmail.com, ORCID : 0009-0009-5849-4676

Abstract
We present an architectural framework for AI clusters and highlight diverse computational demands from four target industry areas. We argue for elastic capacity management due to the dynamic nature of clusters to accommodate these diverse demands. We discuss key challenges and opportunities related to the design of elastic systems for managing diverse workloads based on AI clusters. The choice of system architecture and low-level orchestration mechanisms is guided by capacity planning requirements. We survey and contrast two distinct industry-driven architectural patterns for building AI clusters. We: (1) describe a logical architecture for managing dynamic AI clusters that is flexible and agnostic about the execution daemon; (2) propose two workload models based on the harvesting of real usage data from state-of-the-art AI clusters; (3) provide a proof-of-concept implementation of the proposed latency-QoS optimum placement methodology and analyze its performance.
AI dominates computation today and presents unique system design and management challenges. This paper's contribution is to espouse the idea that one-size-fits-all AI systems cannot work, precisely because AI computations are diverse. They can be computationally expensive, requiring a specialized GPU, or use modern software tricks to perform deep learning on a traditional CPU with reasonable latency. They may be high-performance, real-time, or high-batch throughput. They may demonstrate multi-modal steady-state behavior or varying degrees of start-up and steady-state variance. There are two design challenges: (1) what is the best AI cluster design with a blend of CPU and GPU platforms that satisfies the diverse AI computation needs of the above four scenarios? (2) How best to manage capacity in an AI cluster? We propose answers that critically also utilize data analytics and AI training to understand usage patterns and customer requirements. The system architectures we espouse are elastic; they can increase or decrease capacity in a serverless fashion. A diverse elastic system nicely pairs with elastic capacity management to manage clouds with AI workloads.
Keywords: AI Cluster Architecture, Elastic Capacity Management, Diverse Computational Demands, Dynamic Clusters, Workload Management, System Design Challenges, Industry-specific AI Patterns, Logical Architecture, Execution Daemon Agnosticism, Workload Models, Latency-QoS Optimization, Proof-of-concept Implementation, GPU and CPU Platforms, Real-time AI, High-batch Throughput, Multi-modal Behavior, Data Analytics, Serverless Elastic Systems, Capacity Planning, Cloud AI Workloads.

1Introduction

Enterprises and online organizations increasingly use modern data center clusters to perform large-scale machine learning and AI workloads, which have different demands for computation, I/O, and memory. This extreme heterogeneity is a significant challenge for organizations running these diverse workloads, as it is essential to multiplex the hardware efficiently. Event-driven functions are similarly composed of diverse traffic, with some requests requiring relatively longer to serve than others. In both these scenarios, operators can increasingly leverage elastic capacity management—varying the resources used based on the request being served—to ensure high utilization while keeping queue times below the desired target.
At scale, the efficiency of elastic capacity management has been found to depend heavily on effectively and accurately "right-sizing" requests for resources. This is essentially a hardware resource allocation problem, also known as Tetris. We highlight the importance of fundamental Tetris design principles when choosing an architecture for an AI cluster. The fundamental mathematical principles that dictate Tetris are just as important to get right as the choice of clustering technology. Modern deep learning and other AI technologies are transforming computation. AI technology has been developed further, integrating real-time video input, making it a requirement for training jobs to access and retrieve video data for training. The increasing ubiquity of AI technology in the industry makes AI systems and the problems AI models are trained on ripe for study. We argue that elastic capacity management is a key consideration when architecting such systems. The most state-of-the-art AI research has the potential to have a direct impact on several fields—including medicine, locational science, and computation. Critical research like this should receive the support needed to make a meaningful difference. We will examine AI systems—from both an

industry and research perspective—holistically. We argue that, within this context, elastic capacity management is crucial.

We argue that the fundamental design choices made around this problem in the context of AI clusters will be made around managing their extreme heterogeneity. In a demonstration of this, we will share the results of experiments to simulate different load profiles realistically to demonstrate nuance between Tetri's clusters designed with different systems. In designing and building a system, understanding and addressing the subtleties of traffic diversity and Tetris principles is of particular importance. In the rest of this paper, we will touch on these key approaches to understanding the system and its design. We will cover basic background on the different ways that AI workloads can be separated and the challenges with such workloads. We will also discuss the future of AI systems and the potential impact that they may have.

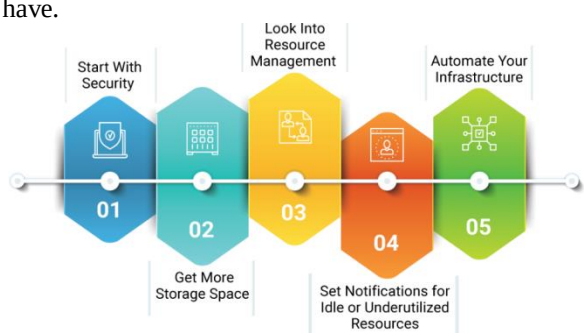

Fig 1 : Elastic Computing

1.1. Background and Significance

Recent years have seen the rapid popularization of artificial intelligence techniques, such as deep learning. These powerful tools can solve tasks that would have taken immense multidisciplinary effort from the computer sciences, such as image and video recognition, language generation, and many more. As a result of the growing number of machine learning and deep learning models being developed, model size has increased from a handful of neurons in the nineties to billions of neurons for models such as GPT-3. Therefore, to keep up with this computational demand, computational power usage is doubling every 3.5 months, and the trend is growing. The high computational demand of AI has also increased energy expenditure, with a single training run consuming just under 190 MWh, enough to power an American household for 6 years. With the advent of friendly, accessible artificial intelligence tools from cloud vendors, systems at small and large scales are witnessing its integration.

Multi-tenant edge clouds, which can dynamically change the infrastructure depending on the workloads and external conditions, require better and more dynamic scheduling algorithms to run these AI clusters efficiently. Traditionally, systems have been built to over-provision to accommodate the resource requirements, a technique that is not cost-effective and scales poorly for AI systems that require near real-time responses. Additionally, the amount of computational power made available to a model also determines the model's performance. Hence, having a system that can automatically vary resources to optimize inference time based on the model's performance in real time is quintessential. More applications are built on machine learning or deep learning models, and the increasing variety warrants the need for the system to be flexible with hardware and software configurations. Running multiple workloads on the same data center, including training, inference, and lightweight workloads that perform on-the-fly annotation, requires the system to distribute server resources depending on the complexity of the workload at that time, not just the nature of the workload itself.

$$ERA = C_{\text{base}} + \sum_{i=1}^{N_{\text{tasks}}} R_i$$

Equation 1 : Elastic Resource Allocation (ERA)

C_{base}: Base computational capacity of the system

R_i: Resource requirement for the i-th task

N_{tasks}: Number of active tasks

1.2. Research Objectives

The nature of the AI clusters is changing, and the machinery that builds the brains of AI is not being replaced frequently. We have seen clusters using hardware from 2012 to the most modern ones. Such heterogeneous use of the compute cluster necessitates an approach where jobs need to manage diverse computational resource requirements. Managing the varied computational needs in the AI clusters, from training large deep learning experiments to handling small inferences, is our primary objective. The idea is to design a scheduling policy along with a complementary feature called elastic capacity that can handle such explorative loads gracefully. For instance, should we allow oversubscription and utilize the slack capacity in the cluster for batch workloads, which experience temporal load variations, or manage the batch workloads via time-sharing policies? Another approach is to design systems that may have higher slack. For example, can we reduce the stragglers among small inferences and hence shrink the tail for the serving workloads to increase the slack in the cluster? We explore these design questions in the following case studies.

Design compute clusters for the new crowd of AI researchers. We will study the dynamic load management mechanisms behind the AI clusters and examine some of the characteristics of the explorative workloads that they handle. Elastic capacity management – Diagnose explorative workloads in AI clusters to understand temporal load variations. The vision is to implement an elastic batch queue mechanism, a new scheme apart from strictly static and elastic batch queues. This new scheme, wherein the explorative workloads are part of the set size-constrained best effort queue, and a separate dynamic queue is used for throughput-sensitive user jobs. It is now possible to oversubscribe the cluster to handle throughput-sensitive synchronous tasks. A scheduling policy around spot instances and slack handling, where spot instances are acquired in surplus and the jobs that have constraints are dynamically matched to the available capacity.

2. AI Clusters: Overview and Architecture

Large-scale deep learning requires a phenomenal amount of computation. An AI cluster is the industrial process center of AI training, composed of a large number of servers for efficient and resilient computation. In AI clusters, servers are interconnected by a high-bandwidth, low-latency network fabric to exchange a large quantity of training data. The interconnected servers further operate as a homogeneous pool of compute resources or as a set of specialized subsystems for tasks such as deep learning. Servers in AI clusters are further orchestrated by distributed and centralized software systems, which allocate and manage jobs based on incoming demand. This operational framework allows AI clusters to perform data-dependent, small-scale tasks such as model evaluation processes as gracefully as large-scale deep learning tasks requiring weeks of computation and days of human labor. In summary, AI clusters are composed of three fundamental elements: hardware, software, and networking. Large independent servers and clusters of smaller form factor servers are often used to create and maintain professional AI clusters. High-end servers or blades contain multiple GPUs and a significant amount of local main memory. Each of these servers shares a networked file system with other servers in the AI cluster over a high-bandwidth network. The software systems that orchestrate AI clusters are built on either a pool of resources or GPU islands. There are multiple designs based on these architectures. Large training clusters are increasingly complex and consist of numerous hardware islands of one of the designs detailed already. There are a myriad of hardware and software variants, but all islands use a similar base technology and thus the same constraints for capacity management. Solving capacity management for one is likely to work for all.

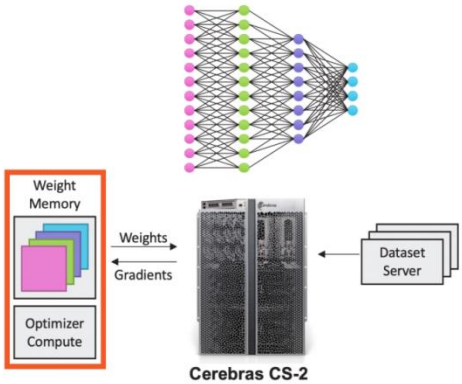

Fig 2 : Core AI Cluster Architecture

2.1. Definition and Components

Definition of an AI Cluster An AI cluster is an optimized computational cluster engineered to function as a cohesive system that is aligned toward a common application or suite of related applications. In the three-layer application-technology-hardware stack, we include applications in the top layer, comprising the services provisioned to the end users; technology makes up the middle layer, defining the orchestration and management functionality enabling the broader ecosystem; finally, hardware appears at the bottom, determining the system's constrained resources. The lower hardware layer consists of interconnected solution components. Each node unit contains an assortment of processors, memory, storage, and drives as it must be self-standing; however, the greater system is engineered to function when an assemblage of nodes is linked. System storage sustains an off-chassis network linkage beyond the set of nodes upon which the CPU nodes can temporarily park their swap and pause storage data state. Over the system link, only certain types of communication are allowed.

Major System Components AI clusters are diverse machine learning as well as training and inferencing clusters. However, those consisting mostly, or exclusively, of DRAM-based AI accelerator servers are the most common instances. The first critical ingredient is, of course, the AI datatype processors employing specific support for gathering information over wildly divergent workloads. Equally imperative, the memory systems contain sizable capacities and memory data speeds such as transfer rates. Similarly, storage solutions, composed of drives and often in an all-flash or tiered topology, make up another essential resource on the server. Thus, each sub-component (processor, memory, storage, and drive) feeds into an amalgamated constellation that achieves and sustains maximum AI cluster functionality and performance. These hardware resources are hardware-component based and are host to subsequent software layer interfaces, technologies, orchestrations, and management. In the case of AI clusters, the interconnection linking hardware to software components is not unidirectional. A feedback pathway follows the principle assumption that efficient and effective engineered functionality, down to the component level, allows optimal and enhanced performance. Therefore, systemic feedback observes cluster-specific variables. These may include overall CPU utilization; next job completion times; node utilization; uplink saturation; inputs such as scaling commands at the chassis, rack, scale unit, or storage node and control plane level; feedback from the lower management or firmware control planes; and the performance feedback accumulations of the linkages between hardware and software components. Meanwhile, the control plane itself decides on potential hardware configuration efficiencies. For example, the configuration management plan records an observation to re architect the hierarchical AI interconnection or two-layer architecture. Such control actions are only possible by the use of systemic AI infrastructure feedback outputs, including such new fields as sensor data.

2.2. Key Technologies

Artificial intelligence has seen remarkable progress over the past few years with advancements in frameworks, algorithms, and networking technologies attaining popularity in parallel. Novel deep learning frameworks leverage billions of parameters for optimal performance, necessitating data processing algorithms capable of handling a diverse mixture of demands concerning computing, communication, and input/output operations. In conjunction with AI innovations, advances in the technology stack – including intelligent tools for data center management, such as network traffic prediction, adaptive routing, and automated failover systems – have pushed the performance of data center clusters even further. Critical for the AI sector is also the recent development of high-bandwidth, low-latency cluster networking technologies that leverage capabilities such as packet pacing, credit-based flow control, and priority flow queueing to ensure that critical workloads, such as the ones needed for machine learning models, experience reduced latency with low jitter.

Furthermore, cloud computing has enabled managed virtualization, which provides a flexible and scalable way to allocate resources to virtual groups without the need for modifications at the physical infrastructure layer. Emerging hardware platforms such as Graphics Processing Units and Tensor Processing Units have also evolved rapidly in recent years to process the rapidly growing AI workload in an energy-efficient manner. Among these, the microsecond memory access time and high memory bandwidth provided by the GPU’s highly parallel memory hierarchy are particularly appealing for accelerating machine learning inference and training operations. Currently, many cloud providers offer infrastructure with GPUs and TPUs, which can be crucial for machine learning researchers and data scientists who wish to leverage significant computing capabilities in their data centers or test novel solutions at scale. This exclusive combination of software and hardware technologies drives the development of these AI clusters, allowing them to effectively address a diverse set of demands. It is highly relevant for cluster stakeholders to understand the rapidly evolving technological landscape and how it currently contours and will shape the functioning of these AI cluster systems. In the following subsection, we discuss the important role these technologies play in AI clusters. We also provide an overview of potential future trends for these technologies.

3. Elastic Capacity Management

Elastic capacity management is a core concept that optimizes resource utilization for AI clusters. It refers to the properties of the system that allow for effectively matching the given computational demands and workload requirements, which may fluctuate over time, by dynamically allocating different system resources. It can also adapt to changing system requirements and priorities. All existing technologies to support elastic capacity management consist of some combination of the following approaches: spatial and temporal. While the detailed implementation of both strategies is a function of numerous factors such as target systems, resource allocation techniques, and system scaling strategies, the focus is on the temporal aspects of the elastic strategies where the system decides how long to retain allocated resources.

Several application workloads that run in large-scale AI clusters, such as video transcoding and large matrix computation, exhibit significant fluctuation in their demand for system resources such as memory and computation. Existing static and over-committed data center environments are challenged by these diverse and fluctuating resource requirements. In particular, a lack of elasticity in an over-committed environment may still lead to allocation inconsistencies via resource scaling. Static and over-committed data centers cannot cater to the diverse needs of the AI workload. Elastic capacity management is a requirement in such a workload due to the dynamic change in demand according to the amount of computational need. It could also involve various methodologies or techniques that enable or automate the elastic capacity strategy implementation in the context of large-scale training. Elastic capacity management is also a broad capability and requirement in any shared data center environment where operational efficiency depends on using as few resources as possible to meet a service's objective. Elastic capacity strategies refer to the design of a system with the ability to cater to the diverse computational demands of a user's workload. In this context, the architectural resource usage diversities are related to the demand diversities in the individual iterative AI applications launched in the cluster. Crystallized in terms of clusters, the diversity of the individual iterative workloads naturally leads to vast differences in the premium paid across users who choose to use the resources unconstrained. Thus, any framework that gives the ability to elastically manage capacity has the additional potential to also enable policy-based, dynamic cost-recovery strategies that a system operator may wish to utilize given the pricing for the resources at their command.

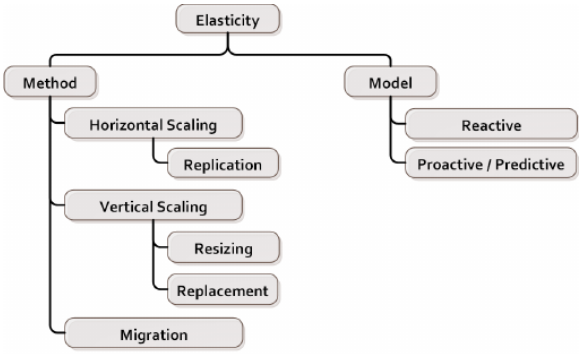

Fig 3 : Classification of Elasticity Solutions

3.1. Concepts and Principles

A good way to handle bursty traffic patterns due to a variable number of EVs is by making AI cluster services elastic—that is, seamlessly and automatically tuning the processing capacity up and down. There are various ways to achieve elasticity, and elasticity is particularly achieved by optimizing the load generators. However, in our case, splitting the target of a user across two different services or statically assigning some target parameters to users is not appropriate. In this respect, we can use two approaches: load balancing and resource provisioning. The former tries to distribute load equally among replica servers by dynamically mapping the arriving requests, while the latter proactively adjusts the processing capacity to match the load. Such an approach is called autoscaling when it is supported by management features such as automated cloud provisioning, workload orchestration, and real-time monitoring and decision-making.

Conceptually, request balancing and auto scaling operate based on a similar view of system resources and elastic QoS (Quality of Service) policy: it is necessary to keep the system load slightly below the system capacity to recover from fluctuations. The difference between them is that auto scaling works for computational systems in general—that is, it increases or decreases the number of computational resources, and it is well suited for serverless computing, microservices, and cloud deployment in general since cloud resources are usually charged by the allocation time. In contrast, request balancing works for server clusters, but it focuses on distributing incoming

requests. For DMaaS, either autoscaling or request balancing is similarly applicable. Elastic scale-up and scale-down can be done by autoscaling, terminating, and creating new VM instances supporting the DMaaS in the cloud. This may result in downtime issues for API users consuming the DMaaS. It looks like there is no notion of capacity, so startups are likely to use request balancing or static performance tuning as alternatives. Automatic performance tuning could avoid user-side downtime issues, but the startup itself has less economic incentive for doing that. In general, resource cleanup duration can vary in a server-side VM termination process due to substitutable instances, content synchronization, etc.

3.2. Importance in AI Clusters

We shall focus on the specific demands of elastic capacity management in AI clusters and cloud-based AI platforms. Running multiple applications and tasks, AI clusters exhibit diverse demands on capacity at any given time. Elastic capacity management is thus of critical importance in an AI cluster as it dynamically allocates resources to applications to minimize the makespan. Stricter SLAs also come into play to ensure business objectives, such as system-wide performance guarantees under competition, penalties, or reputation. The effective management of computing resources in such environments can have a profound impact on performance and operating costs. In particular, periodic over-provisioning may improve plans due to a lower probability of missing SLAs and equally maintain a higher utilization level, but their cost may be untenable on competing systems – no company can sacrifice 50 percent of its profits to meet SLAs.

Where elastic systems come in different instances, the optimal amount of each such resource is important to manage. While this is a more typical concern for capacity management considering different services, it is relevant to AI. Elastic architecture design is closely tied to capacity management and is a focus of both industrial research and practice due to its benefit to the business. We note that AI applications may also be designed to dynamically trade quality and speed to adapt to varying operational environments, a trend that has grown since. Ultimately, the business is interested only in the outcome – namely delivering the best available application as fast as possible for the lowest cost. Where there is a free or very low-cost trial, the enterprise does not want to charge the customer prematurely, nor does it want to lose custom because the AI was too slow or if the service failed. There are, however, very few public statements on the performance of any training process, and even fewer in heterogeneous environments.

$$UE = \frac{\sum_{i=1}^{N_{\text{nodes}}} U_i}{N_{\text{nodes}}}$$

Equation 2 : Utilization Efficiency (UE)

U_i: Utilization of the i-th node

N_{nodes}: Total number of nodes in the cluster

4. Design Considerations for Systems with Diverse Computational Demands

AI systems today carry out a broad diversity of tasks with different computational demands. These may be short or long tasks, metadata indexing or image recognition, running great varieties of DNNs or graph algorithms, working on large or small inputs at once, and so on. Therefore, designers of such systems need to prepare them for changing workload characteristics, balancing efficiency and cost. Achieving this poses distinct design considerations. This outlines the important ones, with the ultimate goal of becoming a starting point for designers and explaining state-of-the-art work and reasonable trade-offs involved in all of them.

When large clusters or entire data centers are designed for AI work, there is a presumption that computational capabilities are either scarce or that tailored designs can achieve the required performance at a lower cost. Different scale clusters requiring the same pool of resources at once

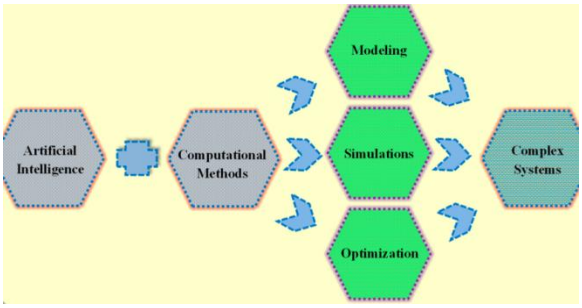

Fig 4 : Advanced Computational Methods for Modeling

4.1. Scalability and Flexibility

Scalability and flexibility are both important properties of the system. When designing a system, scalability is the ability to handle increasing loads (i.e., it is the system's capability that doesn't require major design changes to cope with more significant loads). Elasticity is the ability to provide additional resources to the system and reduce resources from the system as the workloads decrease. Typically, the goal of system design is to be scalable so that the system can absorb increasing loads without major changes. To achieve scalability, a system is frequently designed to consist of multiple components that can distribute the loads and costs. The techniques used can be either through purposive division, such as modular component design, or non-purposive division, such as using cloud-based components or functions. Both divisions provide better resiliency against failures as they typically focus on single components and proficient failover mechanisms.

While under optimal operation, a scalable system surpasses the performance of systems that are not scalable. A scalable system is even more beneficial in cloud-based and distributed systems as it has operational efficiency. As the loads and costs are distributed, we can expect predictable performance behavior. Elasticity will then provide flexibility in terms of per operation or real-time. Operations can still function properly even though there is a delay in retrieving data from the databases; this is due to the versatile design of the scaling middleware, which implements task queues between services. Capable automatic scaling will allow for even better efficiencies as resources are available when needed yet have the elasticity to release resources when they are not needed. Successful elastic capacity utilization gives you operational performance that is beneficial to cost and operational management.

Products often need to be measured for scalability and deployed so that the scaling mechanism is given full consideration, such as avoiding write fragmentation or replication lag from read dependency. The fastest way to deploy and split your identical schematic data is in parallel. Systems that are not designed for massive parallelism cannot be scaled in a cost-effective manner. In terms of a simplistic, single-capacity AI apparatus, there are challenges and also an optimal application design of the big databases and document databases that remediate big data retrievals.

4.2. Performance Optimization Strategies

Utilizing one or a combination of the following strategies, the system's performance can be optimized as per the specific demand. The details of each strategy are as follows:

Workload Prioritization Caching Parallel Processing Workload prediction and placement Machine learning techniques to predict resource demand Recommend quasi-optimizations Predictive modeling Ways in which configurations could be modified for classes of jobs Challenges Scheduling frameworks are generally rigid. Performance is highly dependent on fine-tuned system parameters, ranging from radix to buffer threshold on each chip. Significant monitoring is required. Sensitivity of demand predictability. Forecasting does not equal optimum performance. These are techniques to improve the use of such time series analysis techniques.

Timely development has significantly increased the service's performance, as demonstrated by benchmarks. However, there are many future research opportunities including developing service-based techniques of such models, scheduling system parameters, and validating the models on more complex workloads, such as diverse user time series data. Optimizing the performance and resource utilization of a large-scale analytics system that handles diverse, data-driven computational demands is challenging. In this chapter, we leverage a case study of a state-of-the-art operational AI cluster to elucidate principles and techniques. Given the small value of the benefits, it would have been tempting to not continue with the full-depth optimization. One of our optimizations led to the upgrade of the task name cache to double at a cost of in the deduced monthly hardware cost division. We argue, with evidence, that our model further benefited our customers.

5. Case Studies and Applications

Case Studies and Applications

Dell's 'liquid-cooled' AI cluster vividly illustrates the value of elastic capacity management in practice. There are many such efforts in diverse sectors of the economy. The Tevatron and LHC Large Hadron Collider clusters are examples of very large computer system infrastructures that use grid computing for elastic capacity reconfiguration. Such initiatives are not limited to physics. An inertial odometric guidance system based on an array of AI-based steering mechanisms enables economies of more than 10% of costs. An effort to deploy similar systems in a networked sensor-electronic environment aims to save fuel and enable long-duration military missions. An initiative is funding an effort to design next-generation smart buildings, and a center is developing 'the intelligent office', a sensor- and AI-enriched environment that has the capacity for elastic computational adjustments.

For example, a system is being used to combine a variety of information, e.g., weather, transportation, and light, with the cursor/eye movements of a user to adapt a human-machine interface. A variety of case studies in AI-based home automation systems can be found in the literature. We are experimenting with these two possible applications. Their end-user value has not been determined, but their application will provide us with practical issues of how to use the AI cluster. There are many other initiatives in clusters, grids, and clouds, including hundreds of projects supported by national funding initiatives and a few global activities: • A project involving 30 neuroscientists is building a research infrastructure to enable MEG scanning. They have produced a library of videos.

5.1. Real-world Implementations

Below we summarize five case studies: the operational context, the technical choices, some outcomes, and a few lessons learned.

- AI cluster in a shared radio environment:

Context: Scarce computational resources in CU. Challenges: Dynamically scaling among diverse services and communication technologies. Return: Performance up to par with specialized software across the scenarios.

- AI agents for resource management:

Context: High volume of workloads in a multi-user video transcoding system using the FaaS paradigm. Challenges: Diverse models and unpredictable service operability. Return: Energy savings compared to a conservative scheduler. Lessons learned: Difficulty assessing the actual utility of elastic resources and throttling policies from AIs, despite using off-the-shelf agent libraries. Relatively high costs of calling back the model functions could limit the integration of such algorithms at a large scale today.

- Deep-learning-based diagnosing model for an electric submersible pump:

Context: Training of predictive models out of the cloud. Challenges: Modeling of the engineering domain. Return: Approximately $3M/year if 5% of total pumps were subjected to the discrepancy check. As the model is validated, it is yet to be deployed for real-time monitoring.

- AI cluster in an Italian data center for video transcoding:

Context: High level of heterogeneity regarding encoding requirements. Challenges: Qualitatively assessing the overhead introduced by job dispatching recommendations. Return: The transcoded length of the video in question is up to 4.9% higher.

- Elastic AEM search index in the cloud:

Lessons learned: The results suggest that AI-driven recommendations lead to longer job completion times. The operational practicality and benefits connected to the ease of maintenance and reduced clutter of the Elastic service recommendations should also be weighed before a fully concrete operational plan can be established. The course of action is to reassess in a growth phase whenever architectural changes can drastically change the operational balance.

It is interesting to notice that the set of domains on which AI clusters were tried seems to be quite diverse and not necessarily directed toward the typical AI high-performance computing application scenarios. This is an illustrative list of the domains investigated so far:

- Cloud request management
- Computer vision
- Deep learning
- Radio signal processing
- IoT
- Video encoding/transcoding

5.2. Benefits and Challenges

These AI clusters offer vast benefits. By having a dedicated group of machines that can handle these specialized operations, organizations may be able to get better results from their AI infrastructure. They offer improved

performance, can be more easily and efficiently scaled, and may in some cases be less expensive to operate than traditional infrastructure. They may additionally simplify other parts of your infrastructure, such as backup and disaster recovery. AI clusters are elastic. This means jobs requiring a vast amount of computing and memory resources can be scheduled to run on such clusters. This is handled by the scheduler and the runtime.

However, despite the many benefits, there are several challenges. For example, the numerous types of AI clusters that may be available could lead to complexity in integrating your AI infrastructure. Some clusters could fail. If there is insufficient capacity on other clusters for the workloads running on failed clusters, workloads that were on the failed cluster(s) will need to be stopped or can be rescheduled to other clusters. Lastly, AI technologies are still evolving rapidly. This means best practices and hardware for operating AI workloads will continue to evolve as well, which poses some challenges with capacity planning and assessing the long-term potential cost savings from investing in AI clusters. So organizations need a well-thought-out roadmap of tools and features to add and when to add them. A well-thought-out implementation of AI clusters will take these concerns into account and will also put some careful thought into how to manage workloads that have elastic AI cluster requirements.

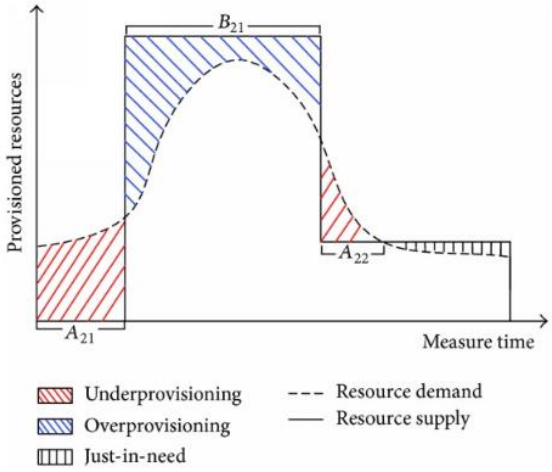

Fig 5 : Elasticity Measurement in Cloud Computing

6. Conclusion

This paper presents features of AI workloads, such as real-time, highly variable needs for computation that are new to centers that run large clusters of AI computation. New designs that can meet these needs will be required to make future large-scale machine-learning systems feasible. The paper also discusses a specific strategy called "elastic capacity management," which a center can use to organize its cluster but has never been feasible to implement for AI because of its high overhead. However, new algorithmic advances are making such an approach possible; specifically, adding extra capacity adapts to the highly variable computational needs of AI. This can save a center 25% or more of the cost and increase the jobs that a cluster can complete by this amount. We hope that managers of large AI systems will find the recommendations and the examples brought from previous explorations of technology interesting. Large clustering of computers generally employs a strategy called "batch scheduling," an approach based on making computation and resources available before computing jobs arrive, using prediction to make those decisions. AI clusters should instead be deployed in an "elastic capacity management" style. That is one such approach, with the added feature that, unlike batch, more servers can be incremented or removed to adjust to the needs of the workloads. We have seen many proposals over the last few years that can improve batch scheduling through smarter ways of placing and running jobs that are known in advance, to improve performance. Yet each of these can do no better than reserving this added amount because jobs are permanently launched and cannot elastically be increased or reduced according to the recent prediction or activities in the AI cluster. For these reasons, in contrast to manual adjustments, many changes are required to the cluster operating systems that are in vogue.

$$ESC = C_{\text{base}} + \sum_{i=1}^{N_{\text{nodes}}} \alpha \cdot C_i$$

Equation 3 : Elastic Scaling Cost (ESC)

C_{base}: Base cost of operation

α: Cost multiplier for scaling

C_i: Operational cost of the i-th node

N_{nodes}: Total number of nodes

6.1. Future Trends

AI computing capacity will continue to grow with the development of technology. This will impact system design choices. The AI research community has been looking into non-traditional models of computation for AI workloads, including quantum computing and reversible computing. New computation models may lead to a redesign of AI clusters. New machine learning frameworks and new AI research directions might affect the design of AI clusters. For instance, hardware-friendly machine-learning frameworks and machine-learning hardware accelerators may replace today's software stack. User demands and interactions with AI systems may change over time; AI clusters may need to be able to adapt quickly to these changing needs. AI clusters also have a non-negligible negative environmental impact. Countries and companies are investing in green data centers that operate in an environmentally friendly way. Elastic resource management will become increasingly important. State-of-the-art managing AI cluster techniques will have to be continuously evolved to handle new computational paradigms, research directions, and user demands in an environmentally sustainable way.

The era of big data has led to exponential growth in data centers. As data center needs and AI computing resources increase with the growth of large-scale data, we will be able to process and utilize big data in a short time. The algorithms, architectures, and computation of massive data computing clusters will change. Multi-tenant data centers are not currently the basis for efficiently managing multiple parallel tasks. In addition, the concept of computational storage will be reused in this AI cluster. The newly deployed AI cluster may be close to the user center. Currently, computing resources are concentrated in a few data centers to serve most users, while AI clusters serving users are deployed in user centers to reduce network delays between users. Future trends in data centers, such as thermal design, are trying to reduce the temperature of data center facilities to room temperature to eliminate the need for refrigeration, but AI clusters are highly server-oriented, so heat generation will be a problem. AI clusters with a liquid cooling mode may appear. At the same time, the waste heat generated by the AI cluster is passed to the surrounding room of the user center and used to heat the user center.

7. References

[1] Syed, S. (2022). Breaking Barriers: Leveraging Natural Language Processing In Self-Service Bi For Non-Technical Users. Available at SSRN 5032632.

[2] Nampally, R. C. R. (2022). Neural Networks for Enhancing Rail Safety and Security: Real-Time Monitoring and Incident Prediction. In Journal of Artificial Intelligence and Big Data (Vol. 2, Issue 1, pp. 49–63). Science Publications (SCIPUB). https://doi.org/10.31586/jaibd.2022.1155

[3] Dilip Kumar Vaka. (2019). Cloud-Driven Excellence: A Comprehensive Evaluation of SAP S/4HANA ERP. Journal of Scientific and Engineering Research. https://doi.org/10.5281/ZENODO.11219959

[4] Rajesh Kumar Malviya , Shakir Syed , RamaChandra Rao Nampally , Valiki Dileep. (2022). Genetic Algorithm-Driven Optimization Of Neural Network Architectures For Task-Specific AI Applications. Migration Letters, 19(6), 1091–1102. Retrieved from https://migrationletters.com/index.php/ml/article/view/11417

[5] Patra, G. K., Rajaram, S. K., Boddapati, V. N., Kuraku, C., & Gollangi, H. K. (2022). Advancing Digital Payment Systems: Combining AI, Big Data, and Biometric Authentication for Enhanced Security. International Journal of Engineering and Computer Science, 11(08), 25618–25631. https://doi.org/10.18535/ijecs/v11i08.4698

[6] Syed, S. (2022). Integrating Predictive Analytics Into Manufacturing Finance: A Case Study On Cost Control And Zero-Carbon Goals In Automotive Production. Migration Letters, 19(6), 1078-1090.

[7] Nampally, R. C. R. (2022). Machine Learning Applications in Fleet Electrification: Optimizing Vehicle Maintenance and Energy Consumption. In Educational Administration: Theory and Practice. Green Publication. https://doi.org/10.53555/kuey.v28i4.8258

[8] Vaka, D. K. (2020). Navigating Uncertainty: The Power of 'Just in Time SAP for Supply Chain Dynamics. Journal of Technological Innovations, 1(2).

[9] Chintale, P., Korada, L., Ranjan, P., & Malviya, R. K. (2019). Adopting Infrastructure as Code (IaC) for Efficient Financial Cloud Management. ISSN: 2096-3246, 51(04).

[10] Kumar Rajaram, S.. AI-Driven Threat Detection: Leveraging Big Data For Advanced Cybersecurity Compliance. In Educational Administration: Theory and Practice (pp. 285–296). Green Publication. https://doi.org/10.53555/kuey.v28i4.7529

[11] Syed, S. (2022). Leveraging Predictive Analytics for Zero-Carbon Emission Vehicles: Manufacturing Practices and Challenges. Journal of Scientific and Engineering Research, 9(10), 97-110.

[12] RamaChandra Rao Nampally. (2022). Deep Learning-Based Predictive Models For Rail Signaling And Control Systems: Improving Operational Efficiency And Safety. Migration Letters, 19(6), 1065–1077. Retrieved from https://migrationletters.com/index.php/ml/article/view/11335

[13] Vaka, D. K. " Integrated Excellence: PM-EWM Integration Solution for S/4HANA 2020/2021.

[14] Sarisa, M., Boddapati, V. N., Kumar Patra, G., Kuraku, C., & Konkimalla, S. (2022). Deep Learning Approaches To Image Classification: Exploring The Future Of Visual Data Analysis. In Educational Administration: Theory and Practice. Green Publication. https://doi.org/10.53555/kuey.v28i4.7863

[15] Syed, S. (2022). Towards Autonomous Analytics: The Evolution of Self-Service BI Platforms with Machine Learning Integration. Journal of Artificial Intelligence and Big Data, 2(1), 84-96.

[16] Nampally, R. C. R. (2021). Leveraging AI in Urban Traffic Management: Addressing Congestion and Traffic Flow with Intelligent Systems. In Journal of Artificial Intelligence and Big Data (Vol. 1, Issue 1, pp. 86–99). Science Publications (SCIPUB). https://doi.org/10.31586/jaibd.2021.1151

[17] Vaka, D. K. "Artificial intelligence enabled Demand Sensing: Enhancing Supply Chain Responsiveness.

[18] Polineni, T. N. S., Pandugula, C., & Ganti, V. K. A. T. (2022). AI-Driven Automation in Monitoring Post-Operative Complications Across Health Systems. Global Journal of Medical Case Reports, 2, 1225.

[19] Syed, S. (2021). Financial Implications of Predictive Analytics in Vehicle Manufacturing: Insights for Budget Optimization and Resource Allocation. Journal Of Artificial Intelligence And Big Data, 1(1), 111-125.

[20] Polineni, T. N. S., Maguluri, K. K., Yasmeen, Z., & Edward, A. (2022). AI-Driven Insights Into End-Of-Life Decision-Making: Ethical, Legal, And Clinical Perspectives On Leveraging Machine Learning To Improve Patient Autonomy And Palliative Care Outcomes. Migration Letters, 19(6), 1159-1172.

[21] Danda, R. R. (2021). Sustainability in Construction: Exploring the Development of Eco-Friendly Equipment. In Journal of Artificial Intelligence and Big Data (Vol. 1, Issue 1, pp. 100–110). Science Publications (SCIPUB). https://doi.org/10.31586/jaibd.2021.1153

[22] Chandrakanth Rao Madhavaram, Eswar Prasad Galla, Hemanth Kumar Gollangi, Gagan Kumar Patra, Chandrababu Kuraku, Siddharth Konkimalla, Kiran Polimetla. An analysis of chest x-ray image classification and identification during COVID-19 based on deep learning models. Int J Comput Artif Intell 2022;3(2):86-95. DOI: 10.33545/27076571.2022.v3.i2a.109

[23] Mandala, V., & Mandala, M. S. (2022). ANATOMY OF BIG DATA LAKE HOUSES. NeuroQuantology, 20(9), 6413.

[24] Nimavat, N., Hasan, M. M., Charmode, S., Mandala, G., Parmar, G. R., Bhangu, R., ... & Sachdeva, V. (2022). COVID-19 pandemic effects on the distribution of healthcare services in India: A systematic review. World Journal of Virology, 11(4), 186.Nimavat, N., Hasan, M. M., Charmode, S., Mandala, G., Parmar, G. R., Bhangu, R., ... & Sachdeva, V. (2022). COVID-19 pandemic effects on the distribution of healthcare services in India: A systematic review. World Journal of Virology, 11(4), 186.

[25] Korada, L. (2022). Using Digital Twins of a Smart City for Disaster Management. Journal of Computational Analysis and Applications, 30(1).

[26] Vankayalapati, R. K., & Rao Nampalli, R. C. (2019). Explainable Analytics in Multi-Cloud Environments: A Framework for Transparent Decision-Making. Journal of Artificial Intelligence and Big Data, 1(1), 1228. Retrieved from https://www.scipublications.com/journal/index.php/jaibd/article/view/1228

[27] Maguluri, K. K., Yasmeen, Z., & Nampalli, R. C. R. (2022). Big Data Solutions For Mapping Genetic Markers Associated With Lifestyle Diseases. Migration Letters, 19(6), 1188-1204.

[28] Sondinti, L. R. K., & Yasmeen, Z. (2022). Analyzing Behavioral Trends in Credit Card Fraud Patterns: Leveraging Federated Learning and Privacy-Preserving Artificial Intelligence Frameworks.

[29] Vankayalapati, R. K., Edward, A., & Yasmeen, Z. (2021). Composable Infrastructure: Towards Dynamic Resource Allocation in Multi-Cloud Environments. Universal Journal of Computer Sciences and Communications, 1(1), 1222. Retrieved from https://www.scipublications.com/journal/index.php/ujcsc/article/view/1222

[30] Kothapalli Sondinti, L. R., & Syed, S. (2021). The Impact of Instant Credit Card Issuance and Personalized Financial Solutions on Enhancing Customer Experience in the Digital Banking Era. Universal Journal of Finance and Economics, 1(1), 1223. Retrieved from https://www.scipublications.com/journal/index.php/ujfe/article/view/1223

[31] Subhash Polineni, T. N., Pandugula, C., & Azith Teja Ganti, V. K. (2022). AI-Driven Automation in Monitoring Post-Operative Complications Across Health Systems. Global Journal of Medical Case Reports, 2(1), 1225. Retrieved from https://www.scipublications.com/journal/index.php/gjmcr/article/view/1225

[32] Maguluri, K. K., Pandugula, C., Kalisetty, S., & Mallesham, G. (2022). Advancing Pain Medicine with AI and Neural Networks: Predictive Analytics and Personalized Treatment Plans for Chronic and Acute Pain Managements. Journal of Artificial Intelligence and Big Data, 2(1), 112–126. Retrieved from https://www.scipublications.com/journal/index.php/jaibd/article/view/1201

[33] Tulasi Naga Subhash Polineni , Kiran Kumar Maguluri , Zakera Yasmeen , Andrew Edward. (2022). AI-Driven Insights Into End-Of-Life Decision-Making: Ethical, Legal, And Clinical Perspectives On Leveraging Machine Learning To Improve Patient Autonomy And Palliative Care Outcomes. Migration Letters, 19(6), 1159–1172. Retrieved from https://migrationletters.com/index.php/ml/article/view/11497

[34] Ravi Kumar Vankayalapati , Chandrashekar Pandugula , Venkata Krishna Azith Teja Ganti , Ghatoth Mishra. (2022). AI-Powered Self-Healing Cloud Infrastructures: A Paradigm For Autonomous Fault Recovery. Migration Letters, 19(6), 1173–1187. Retrieved from https://migrationletters.com/index.php/ml/article/view/11498