Automated Hardware-Aware Node Selection for Cluster Computing

Manuel Sopena Ballesteros
Systems engineer
CSCS
Email: msopena@cscs.ch

Matteo Chesi
Senior systems engineer
CSCS
Email: matteo.chesi@cscs.ch

Miguel Gila
Senior Systems Engineer
CSCS
Email: miguel.gila@cscs.ch

Mark Klein
Senior architect
CSCS
Email: mark.klein@cscs.ch

***Abstract*—This study introduces a set of novel algorithms designed to automate the grouping of compute nodes into clusters based on user-defined hardware requirements in high-performance computing (HPC) data centers. These algorithms effectively address the complexities of dynamic workloads by utilizing Cray System Management (CSM) [1] APIs to gather and process detailed hardware information. By automating the selection of nodes to form clusters according to user-specified criteria, algorithms enhance operational efficiency and optimize resource utilization.**

Automation streamlines node assignment, reduces human error, and expedites the selection process. By systematically grouping nodes with compatible hardware, the algorithms ensure the creation of clusters that meet precise user requirements, thus optimizing the performance and resource allocation in HPC environments.

One of the key contributions of this study is the introduction of a scoring mechanism that refines node selection. This mechanism allows for informed decisions based on a range of criteria, such as penalizing nodes with hardware configurations that do not align with the user preferences. The scoring system ensures that clusters are not only formed based on user-defined hardware requirements, but also in a way that maximizes the potential of available resources.

Beyond automated node selection, these algorithms offer additional benefits such as improved scalability and flexibility in resource management. They enable data centers to efficiently accommodate changing workloads and evolving user need, which is essential for maintaining competitiveness in HPC environments.

In summary, the proposed algorithms present a holistic solution for automated node selection, representing a significant advancement in optimizing resource allocation and strengthening the computing infrastructure across multiple tenants. By leveraging these algorithms, data center operators can achieve greater efficiency, flexibility, and reliability in HPC operations.

1. Introduction

Efficiency and resource optimization are of great importance in the field of high-performance computing (HPC). The Swiss National Super Computing Center (CSCS) has taken a significant step towards achieving this by introducing its latest HPC platform, Alps [2], built on the HPE Cray EX machine and managed by Cray System Management (CSM). This study aimed to investigate an algorithm specifically designed to streamline infrastructure provisioning by focusing on the new Alps platform. Alps serves as an Infrastructure as a Service (IaaS) platform, distinguishing itself from the traditional IaaS models commonly seen in public clouds. Unlike relying on virtualization for CPU and memory resources, Alps adopts a different approach by partitioning the HPE Cray Ex compute nodes into distinct groups dedicated to individual tenants. The main objective of Alps is to enhance the research workload by reducing administrative burdens through automation. The presented research concentrates on the development and implementation of an algorithm that can effectively translate hardware descriptions into a coherent set of compute node names. By doing so, it eliminates the need for users to navigate through extensive infrastructure documents to understand the type of nodes available. This algorithm was developed to address a variety of challenges. Its main purpose is to introduce a user-friendly approach to defining the hardware setup of a group of nodes. This setup acts as the basis for the algorithm assessment of node groups. The assessment of these node groups serves a dual function: First, it enables a comparison between the current state of the cluster and the desired state of the user. Furthermore, it identifies compute nodes that are suitable for integration into a new cluster, meeting the user's specific requirements. Users must grasp the hardware required for their workload. They provided a cluster definition that outlines this hardware and serves as the starting point for the algorithm. Through this definition, the algorithm understands and implements user needs by selecting the most appropriate nodes. The algorithm takes a practical approach, striving to minimize the addition of nodes to a new cluster while meeting user requirements. It operates by utilizing a resource pool to select nodes. In addition to meeting the user needs, the algorithm aims to offer insights into the hardware makeup of a cluster. These insights guide necessary adjustments, such as adding or removing nodes, to align effectively with the user requirements. Furthermore, the algorithm offers flexibility and is capable of creating, expanding, or reducing cluster sizes as needed. This adaptability ensures scalability and the ability to adjust to changing demand. One significant advantage of this algorithm is its ability to separate user requirements from specific compute node types or hostnames. This brings about a shift in resource allocation dynamics, allowing for a customized selection of compute nodes that aligns with the user's needs based on the availability of allocated resources. Moreover, this algorithm facilitates seamless cluster migrations between different HPE Cray EX machines, as it simplifies the complications associated with hardware, enabling users to focus solely on their research objectives.

Moreover, the Cray System Management (CSM) platform offers a unique perspective on system management in high-performance computing (HPC) environments. Specifically designed for the HPE Cray EX system, CSM amalgamates traditional HPC system management functionalities with cloud-computing principles. By leveraging a microservice architecture and Kubernetes orchestration, CSM enhances automation, authentication mechanisms, and comprehensive monitoring capabilities. This amalgamation of cloud principles with HPC infrastructure management exemplifies a progressive approach towards efficient resource utilization and system administration.

2. Related work

2.1. Node allocation in CSCS

The current process of allocating nodes to clusters that we are trying to improve is as follows: Upon receiving a request from a user to update the hardware requirements, the system administrator undertakes a series of steps. Firstly, the administrator examines an Excel spreadsheet that encompasses a comprehensive list of connections among racks, nodes, and hardware components. This spreadsheet serves as a reference for the administrator to identify the new nodes that needs to be part of the cluster the user is focusing on.

To maintain an organized structure, the relationship between the compute nodes and the clusters they belong to is represented in a file structure within a git repository. Each folder within the repository corresponds to a cluster, while the files within the folders represent the compute nodes that are members of that particular cluster. It is crucial that the folder names match the cluster names, and the file names align with the compute node names as shown in Figure 1.

Next, the system administrator proceeds to pull or clone the git repository, retrieving the files that represent the cluster and compute nodes mentioned above. These files are then reshuffled so the tenant cluster represents the user request. Subsequently, the administrator commits and pushes these modified files back to the git repository, ensuring that the changes are saved and updated.

To automate the process, a scheduled script running on one of the management nodes continuously monitors the filesystem that represents the clusters. Upon detecting any changes, the script calculates the modifications and initiates subsequent calls to the CSM API, thereby updating the cluster definition accordingly.

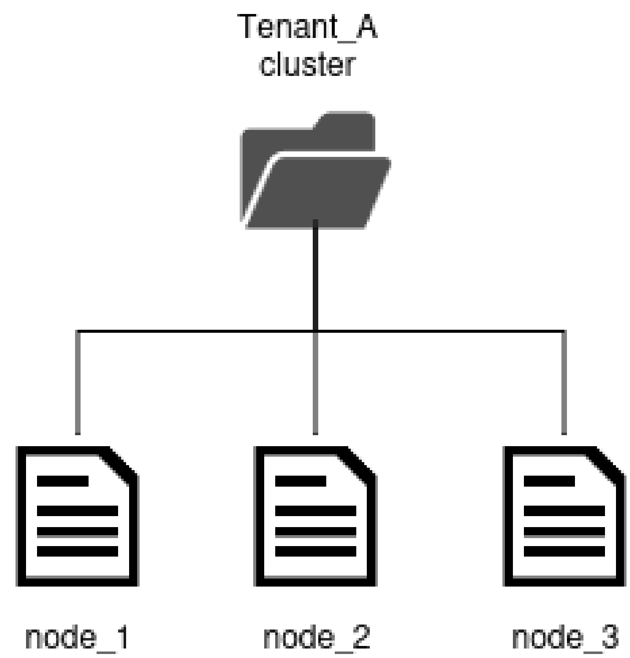

Figure 1. Node and cluster file structure relationship

Finally, the system administrator takes the responsibility of updating the documentation with the new cluster details, ensuring that all relevant information is accurately recorded for future reference.

In summary, our current process of allocating nodes to clusters involves a manual allocation process overseen by a system administrator using an Excel spreadsheet with subsequent configuration adjustments executed through a script that interacts with the CSM API. The algorithms developed in this study attempted to automate this process.

2.2. Node allocation in public clouds (non HPC)

The distribution of nodes within a cluster in a commercial cloud is performed through a wide range of virtual machine (VM) types, each designed to meet specific needs. Users often need to refer to detailed documentation to understand the unique features of each VM type, and then decide on the most suitable VMs and quantities required for their specific workloads. This step requires users to be familiar with the available options during the resource-provisioning phase.

In the traditional sense, provisioning in commercial cloud environments under the Infrastructure as a Service (IaaS) model demands that users have a deep understanding of the different instance types at their disposal. Users need to evaluate performance characteristics and make informed decisions that align with their application needs. Based on this knowledge, clusters are configured with specifications tailored to user preferences and application requirements.

In summary, while current IaaS provisioning methods rely on user knowledge of available instance types, the new algorithm proposed in this study presents an novel approach that enables users to specify their hardware requirements directly. This method simplifies the process by abstracting the selection of specific instance types, thereby offering a more streamlined and user-friendly experience. Furthermore, the incorporation of cloud computing principles into system management, as exemplified by the Cray System Management platform, showcases the evolution of High-Performance Computing (HPC) management towards enhanced automation and scalability. This transition signifies a shift towards more efficient, intelligent, and adaptable resource-management strategies in contemporary computing environments.

2.3. vClusters vs pool of resources vs HSM groups

The main objective of this study was to establish clusters that consist of nodes within a cluster equipped with the necessary hardware components to fit a specific workload. In CSM terminology, a cluster is represented by a HSM (Hardware System Management) group, which represents a cluster. Each cluster is identified with a specific label and a set of nodes, which are referred to as either ”vCluster” or ”pool of resources”. In this context, the distinction between these terms lies in the fact that ”vCluster” is utilized by a group of users to run workloads, while ”pool of resources” is where shared resources are maintained. When adding hardware components to a vCluster, they are transferred from the ”pool of resources”, while removing them returns them to the ”pool of resources”. The algorithms presented in this study should support various operations, such as adding or removing hardware components from or to a vCluster or creating a new vCluster based on final configuration. It is crucial that the algorithm allows a vCluster to function as a ”pool of resources” for a transaction if the user owners agree to move nodes from one vCluster to another. All operations must be conducted transactionally to ensure that any issues that arise during node transfer between different vClusters remain unchanged.

3. Methods

The algorithm outlined in this research aligns with the CSCS’s multitenancy strategy, emphasizing the segregation of resources for distinct user groups to uphold system integrity, security, and efficient resource allocation based on diverse user requirements.

Before delving into the specifics of the algorithm, it is imperative to establish fundamental concepts that form the basis of its operation.

3.1. Target and parent vClusters

The compute nodes were organized into vClusters or HSM groups. HSM, which represents Hardware System Management, serves as the CSM service responsible for managing clusters and hardware. It maintains a comprehensive list of all the clusters in the system, including the member nodes, and provides information about the hardware components within each node. Within this framework, we identified two types of HSM groups: the target HSM group and parent HSM group, both of which can be referred to as HSM groups or clusters.

The proposed algorithm operates with two different types of virtual clusters (vClusters) or hardware security module (HSM) groups. These groups play specific roles in the system. The target HSM group consists of nodes that execute the user workload following the hardware specifications provided by the user. These specifications help the algorithm determine the adjustments required for the target vCluster. Throughout this document, the terms cluster, vCluster, and HSM group are used interchangeably.

Parent vCluster represents the pool of spare resources in the system. It acts as a reservoir for resource management, based on the operational state of the target vCluster. If the target vCluster is scaled down, any additional resources are allocated to the parent vCluster. Conversely, if the target vCluster must be scaled up, the necessary additional resources are obtained from the parent vCluster.

The interaction between these two vClusters enables dynamic and efficient resource management in the Alps system, enhancing the efficiency and adaptability in response to changing workload demands.

3.2. Hardware component, hardware component summary and user input

The HSM service within CSM keeps an inventory of the hardware in the HPE Cray Ex machine, and HSM contains a functionality to scan the network looking for BMCs, query them, and fetch the hardware they contain; this information is then stored in a local database and it is accessible through a http API. Our algorithm queries the HSM endpoints to fetch the hardware components of each node in both the target and the parent vClusters.

The term ‘hardware component’ refers to a specific part of the hardware that is directly associated with the HSM (Hardware Security Module) inventory. The HSM hardware inventory service provides examples of hardware components such as:

- $NVIDIA_A100 - SXM4 - 80GB$
- $AMDINSTINCTMI200(MCM)OAMLC$
- $AMDEPYC774264 - CoreProcessor$
- $SS11200Gb2PNICMezzREV02(HSN)$

Each hardware component belongs to one of the four different types.

- Processors: This group includes Central Processing Units (CPUs) that are responsible for executing instructions and performing calculations on a computer system. Processors are essential components in the functioning of computers or servers.
- Node accelerators: Devices like GPUs (Graphics Processing Units) belong to this group. GPUs are specialized processors designed to handle complex

graphics rendering and parallel-processing tasks. They are commonly used in applications that require high-performance computing such as gaming, scientific simulations, and artificial intelligence.
- HSM NIC: This group includes network interfaces. Network Interface Cards (NICs) are hardware components that enable computers or servers to connect to a network. They provide the necessary communication capabilities for data transmission between the devices.
- Memory: This group comprises memory Dual In-Line Memory Modules (DIMMs). Memory modules are electronic components that temporarily store data and instructions while a computer or server is running. They provide fast access to data, allowing for the efficient processing and execution of tasks.

It is important to note that certain types of memory, such as soldered memory in chips (e.g., GPUs) or super chips (e.g., grace hoppers), are not considered part of the memory group in the HSM hardware inventory.

While not all hardware components in the HSM hardware inventory are required, this study will focus on utilizing the following types: processors, accelerators, and memory. Network cards were excluded from this study because they were the same model used in Alps.

A structure is required to depict the number of hardware components in groups of nodes. This structure serves as user input and aids in monitoring the status of the target vCluster and the parent vCluster, as nodes are transferred between them by the algorithm. The chosen format is known as ”Hardware component pattern” and adheres to the format ”¡hw component¿:¡quantity¿[:¡hw component¿:¡quantity¿]”, which showcases the various hardware components and the quantity in which they are present in a set of compute nodes. For instance, if we have two types of compute nodes: Node Type A:

- $x4 - NVIDIA_A100 - SXM4 - 80GB$
- $x1 - AMDEPYC771364 - CoreProcessor$
- $x16 - 16GiBmemory$

Node type B:

- $x2 - AMDEPYC774264 - CoreProcessor$
- $x32 - 16GiBmemory$

Assuming we have a small cluster with three nodes–two of type A and one of type B–the hardware component summary for this cluster would be $NVIDIA_A100 - SXM4 - 80GB : 8 : AMDEPYC771364 - CoreProcessor : 1 : AMDEPYC774264 - CoreProcessor : 2 : memory : 80$, indicating that the cluster comprises eight Nvidia A100 GPUs, one AMD EPYC 7713 CPU, two AMD epyc 7742 cpus, and 80 × 16 GiB of RAM. It is important to note that a unit of memory is not represented by the DIMM sizes because of potential variations across the compute nodes. To ensure that hardware components can be quantified for comparison with other vClusters, we defined a unit of memory as 16 GiB, calculated as the maximum common multiplier of all DIMMs in the system. This calculation, although costly, only needs to be performed during HSM hardware discovery. The advantage of this format in defining a cluster is its ability to summarize a group of nodes from a hardware perspective.

3.3. Fuzzy finder

In this study, the algorithm incorporated a fuzzy finder to streamline the search process for hardware components within the HSM hardware inventory. The purpose of this feature is to improve the user experience by simplifying search operations.

For example, if a user is looking for a cluster with specific hardware components, such as $NVIDIA_A100 - SXM4 - 80GB : 12 : AMDEPYC771364 - CoreProcessor : 3 : memory : 96$, they can achieve the same outcome by using the shorthand notation $a100 : 12$.

The algorithm filters the hardware components from the HSM hardware inventory and conducts a fuzzy search to identify those that match the input $a100$. This fuzzy search allows flexibility in the search process.

Once the algorithm identifies the hardware components that match the input, it determines that the remaining required components ($AMDEPYC771364 - CoreProcessor$ and $memory$) are already present in the same compute node. This eliminates the need for the user to include these components in the user-defined hardware component summary.

Overall, the incorporation of the fuzzy finder algorithm in this study aims to simplify and expedite the search for hardware components within the HSM hardware inventory, ultimately enhancing user experience.

3.4. Deltas calculation

The algorithm relies heavily on the calculation of deltas to identify the differences between the current hardware summary of a vCluster and the specified requirements of the user, which define the vCluster end state. These deltas serve as a guide for necessary modifications to be made in vCluster.

To illustrate this, we consider the following example. Suppose the user input is $a100 : 4 : epyc : 1$, which means that the user wants to have four Nvidia GPUs of type a100 and one AMD CPU of type epyc in vCluster. However, the current hardware summary for the target vCluster indicates that two epyc CPU present.

In this case, the algorithm calculates delta by comparing the user requirements with the current hardware summary. The delta was computed as $a100 : +4$ and $epyc : -1$. This calculation indicates that the algorithm should add four a100 Nvidia GPUs and remove one epyc AMD CPU from the target vCluster to fulfill the user's request.

By relying on these calculated deltas, the algorithm can efficiently determine the necessary modifications in vCluster. This approach ensures that vCluster is configured according to the user's specified requirements, considering the current hardware summary.

3.5. The algorithm

The system operates through a defined set of steps to efficiently manage hardware allocation within vClusters following Figure 2:

1) The user submits a hardware component summary which defines the hardware components the target vCluster should have, outlining the requisite resources.
2) Get the list of hardware components related to both, the target and parent vClusters through the HSM hardware inventory API in CSM, apply fuzzy searching to make sure the keywords provided by the user matches with the hardware components information in the system.
3) Combining all nodes in both the target vCluster and the parent vCluster into a single vCluster, this new vCluster will be called a ”combined vCluster” and its members called ”eligible nodes” because they are all potential candidates for fulfilling user requests. Once the combined vCluster has been created, its members are sorted alphabetically. Please note that cray systems use a special format for hostnames called xnames, which defines the geographical location of a hardware component; in our case, when sorting hostnames alphabetically, we also sort them geographically.
4) Calculate each node’s score. This scoring mechanism enables the identification of the optimal or best candidates for node relocation, promoting the right choice when moving nodes to its final destination being the target vCluster.
5) Identify the node with the highest score, indicative of the best candidate for fulfilling the user request.
6) Once the best candidate node is determined, the system orchestrates its migration to the target vCluster, ensuring the allocation of resources where they are most needed.
7) Adjusts the hardware component summaries for the user request, the target and parent vClusters. Effectively the hardware component of the best candidate need to be subtracted from the user request and should reflect the new state of the target and parent vCluster.
8) If the user requests still has resources to move, go to step 4 and continue from there.

In essence, this systematic approach underscores the system’s capacity to dynamically manage hardware allocation within vClusters, leveraging data-driven decision-making to optimize resource utilization and meet user demands effectively. Through its iterative process and strategic node migration, the system facilitates seamless resource allocation.

4. Results

The research presents a method for quantifying hardware components to enable comparison between nodes. It

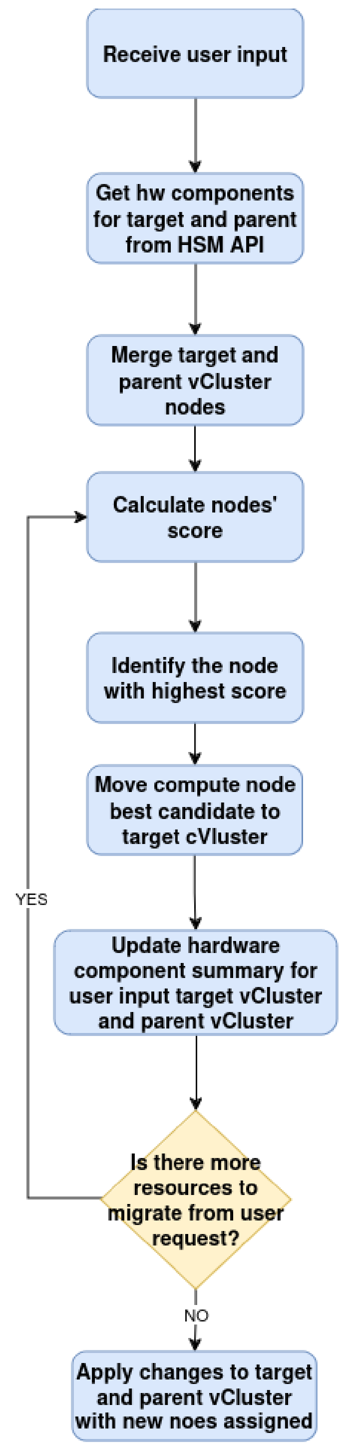

Figure 2. Algorithm diagram

supports various types of hardware (processors, accelerators, and memory) and empowers users to configure a cluster with hardware tailored to their needs.

The system ensures that user requests are met; otherwise, it will fail. For instance, if the user requests more a100 that is available in the parent vCluster, then the operation will fail. Fuzzy searching streamlines the process of filtering/selecting the desired hardware, while simultaneously simplifying the work the user has to do to find the hardware components it wants.

When migrating clusters from one CSM to another, this algorithm offers the benefit of avoiding the need to manually input the hostname list in the cluster description file. Typically, the list of nodes in a cluster is closely linked to the hardware required by the user. In this scenario, the hostname cannot be reused between different CSM instances due to potential variations in the number of nodes or the hardware configuration of the same hostname in different CSMs.

We also identified use cases in which this algorithm may not be valid or require adjustments. The algorithm involves merging the target and parent vClusters at a certain stage, followed by moving nodes from the combined vCluster of eligible nodes back to the target vCluster. This process could result in the target vCluster receiving completely different nodes than it had previously, which could be highly disruptive for clusters utilizing a workload manager, such as Slurm. Ideally, we aim for the algorithm to minimize the number of nodes in the target vCluster that are altered, while retaining those that align with the user's requirements. Therefore, we do not recommend following this algorithm in a production environment.

Furthermore, all nodes assigned to the vCluster are sorted alphabetically, with the best candidate (the node with the highest score) being selected for movement in a sequential manner. While this may reduce the risk of fragmentation among all the vCluster managed by a tenant, it compromises high availability features. It would be beneficial to provide the option of distributing hardware components across different racks/chassis within a cluster or target racks with specific energy requirements.

Currently, they way users can use this functionality is through Manta. Manta is a cli developed by CSCS to simplify user operations and also add new features like an implementation of this study.

5. Discussion

Based on the concerns highlighted in the preceding section, additional functionalities can be introduced to broaden the scope of the new use cases. One potential enhancement could involve implementing a feature that filters the nodes eligible for transfer to the target vCluster. This enhancement has two key advantages. First, in cases where multiple candidates possess identical scores, preference could be given to nodes that were previously part of the original target vCluster, which would maximize the number of nodes that could be reused from the original target vCluster and optimize downscale and/or upscale operations on clusters running workload managers. Second, the ability to filter nodes based on specific criteria, such as nodes within the same chassis or rack, could prevent an excessive concentration of nodes within a single rack or chassis and increase the availability of capabilities. Both of these benefits would make the final solution more production-ready.

Consider creating a hardware component summary format to map workloads and cluster sizes. This format streamlines the cluster definition by aligning the workload requirements with cluster size. For example, workloads can be categorized based on the percentage of hardware components required relative to the overall cluster resources with the total figures derived from the cluster size.

To address this issue, we plan to explore a more efficient and tailored data structure. Relying solely on a standard library may not offer an optimal solution in this scenario.

References

[1] Hewlett Packard Enterprise, "Cray System Management Documentation," https://cray-hpe.github.io/docs-csm/en-10/, 2021, accessed: 2024-03-15.

[2] Swiss National Supercomputing Centre (CSCS) and Hewlett Packard Enterprise (HPE) and NVIDIA, "CSCS, Hewlett Packard Enterprise, and NVIDIA Announce World's Most Powerful AI-Capable Supercomputer," https://www.cscs.ch/science/computer-science-hpc/2021/cscs-hewlett-packard-enterprise-and-nvidia-announce-worlds-most-powerful-ai-capabl 2021, accessed: 2024-03-15.