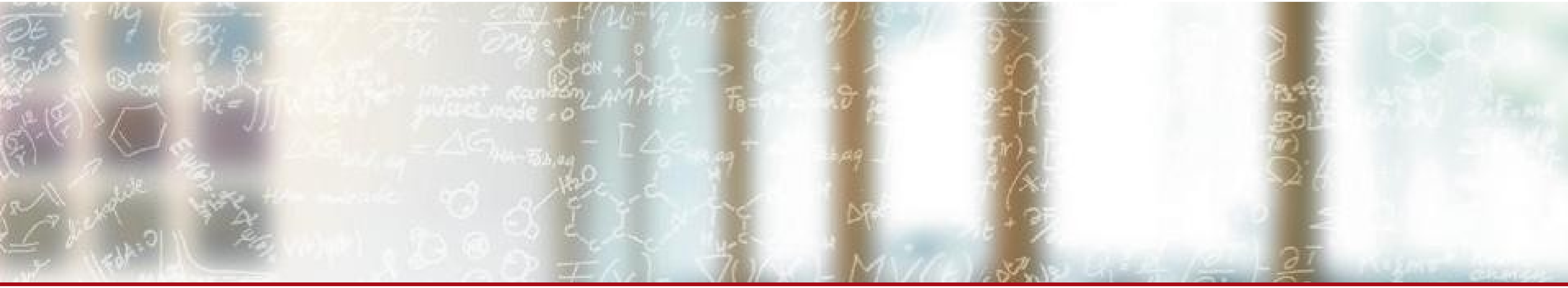
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Automated Hardware-Aware Node Selection for Cluster Computing

CUG24

Manuel Sopena Ballesteros, CSCS

May 07, 2024

Table of Contents

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Introduccion

ALPS

We develop and operate a high-performance computing and data research infrastructure that supports world-class science in Switzerland

HPC system

- Multi-tenancy (IaaS)
- Geographically distributed
- Heterogeneous hardware

Goals

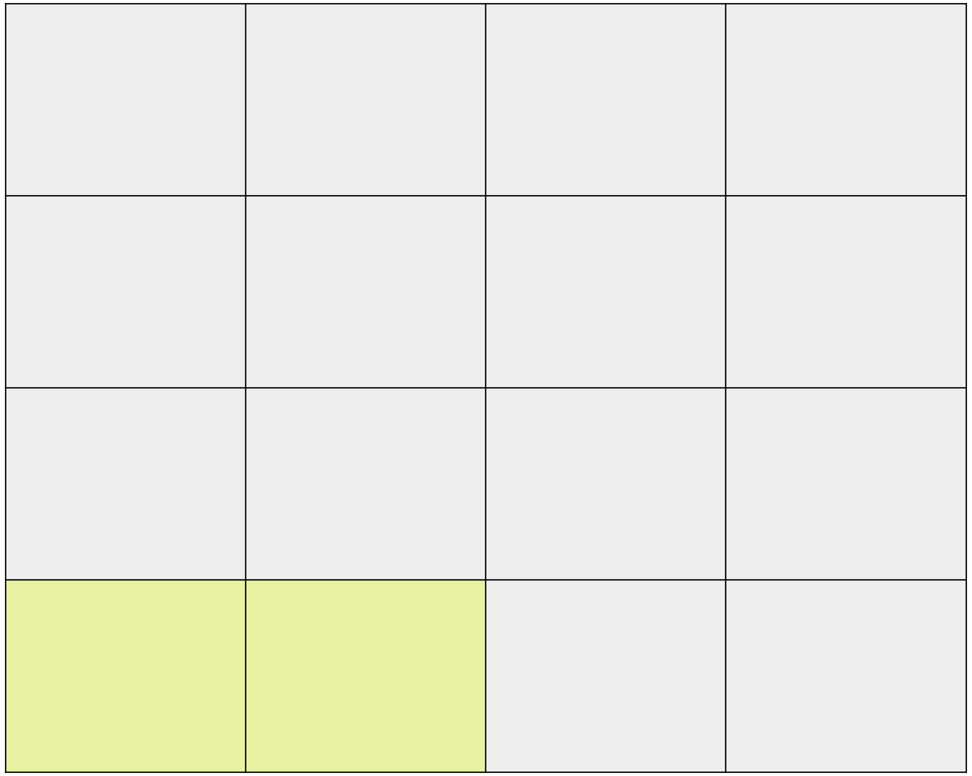
- Find nodes for a specific workload based on a hardware description
- Simplify cluster management
- Build an algorithm and an implementation
- Provide more flexibility to the user to define their clusters
  - https://aws.amazon.com/ec2/instance-types/
  - https://learn.microsoft.com/en-us/azure/virtual-machines/sizes-hpc
  - https://learn.microsoft.com/en-us/azure/virtual-machines/sizes-general
  - https://learn.microsoft.com/en-us/azure/virtual-machines/sizes-compute
  - https://learn.microsoft.com/en-us/azure/virtual-machines/sizes-memory
  - https://learn.microsoft.com/en-us/azure/virtual-machines/sizes-storage
  - https://learn.microsoft.com/en-us/azure/virtual-machines/sizes-gpu

Simplify node management

- Cluster tenantA:
  - X1001c1s5b0n0
  - X1001c1s5b0n1
  - X1001c1s5b1n0
  - X1001c1s5b1n1
  - X1001c1s6b0n0
  - X1001c1s6b0n1
  - X1001c1s6b1n0
  - X1001c1s6b1n1
  - X1001c1s7b0n0
  - X1001c1s7b0n1
  - X1001c1s7b1n0
  - X1001c1s7b1n1
  - X1005c0s4b0n0
  - X1005c0s4b0n1
  - X1006c1s4b0n0
  - x1006c1s4b1n0

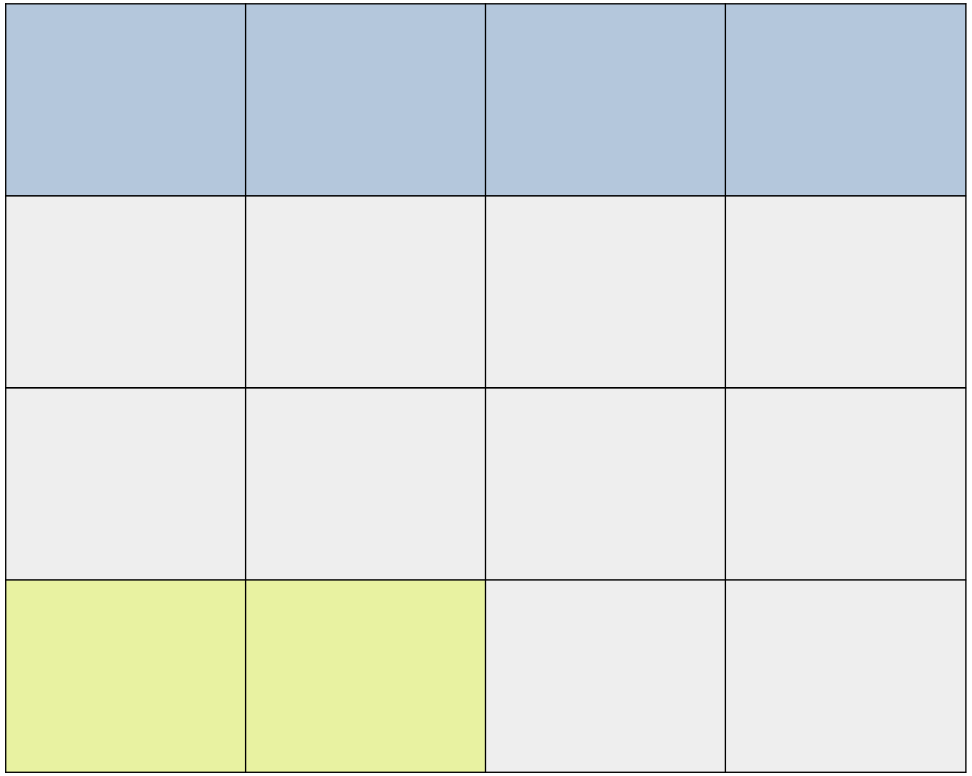
Simplify node management

- Cluster tenantA_AI:
  - x1005c0s4b0n0
  - x1005c0s4b0n1
- Cluster tenantA:
  - x1001c1s5b0n0
  - x1001c1s5b0n1
  - x1001c1s5b1n0
  - x1001c1s5b1n1
  - x1001c1s6b0n0
  - x1001c1s6b0n1
  - x1001c1s6b1n0
  - x1001c1s6b1n1
  - x1001c1s7b0n0
  - x1001c1s7b0n1
  - x1001c1s7b1n0
  - x1001c1s7b1n1
  - x1006c1s4b0n0
  - x1006c1s4b1n0

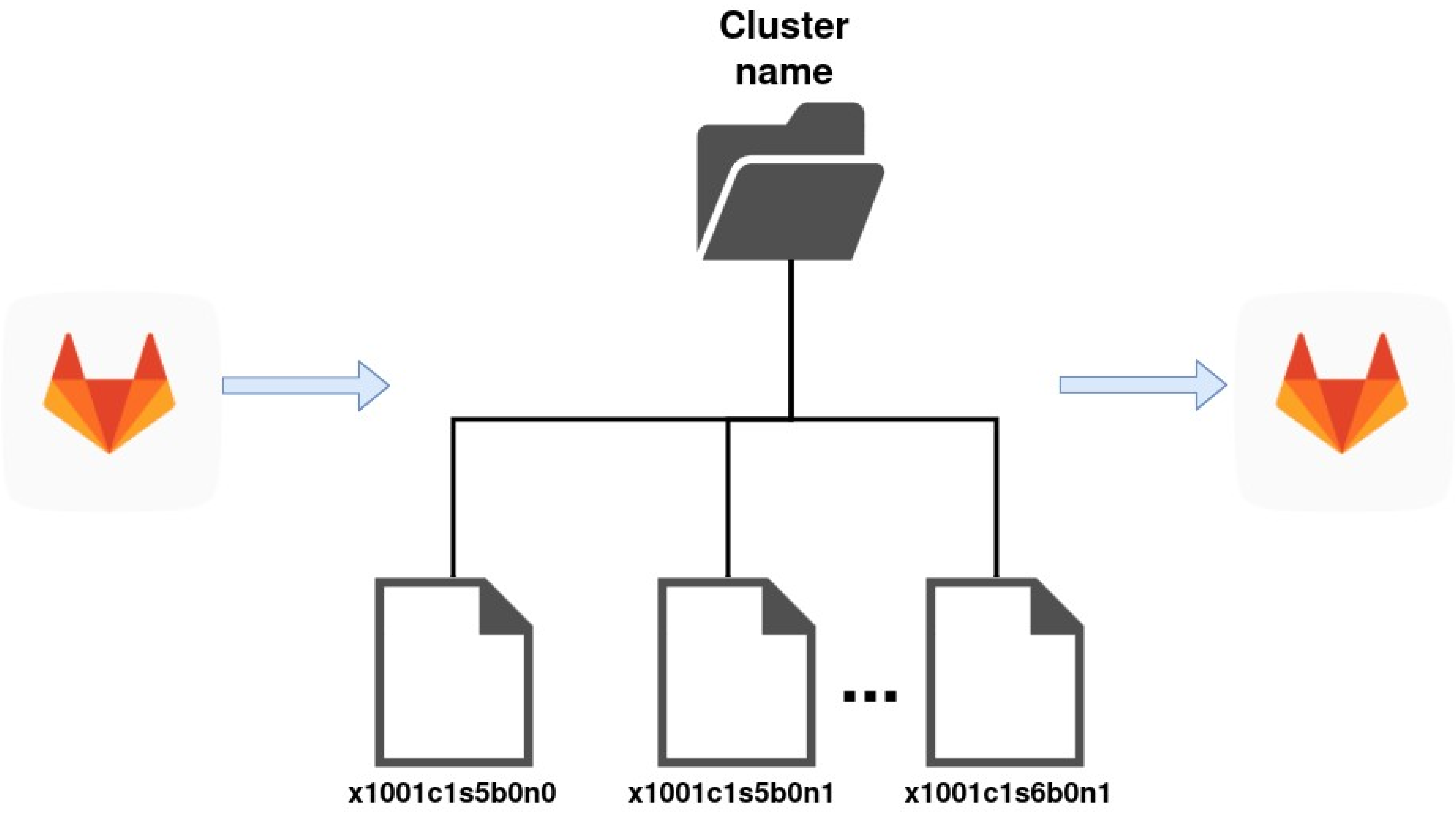


Simplify node management

- Cluster tenantA_CPU
  - X1001c1s5b0n0
  - X1001c1s5b0n1
  - X1001c1s5b1n0
  - X1001c1s5b1n1
- Cluster tenantA_AI
  - x1005c0s4b0n0
  - x1005c0s4b0n1
- Cluster tenantA
  - X1001c1s6b0n0
  - X1001c1s6b0n1
  - X1001c1s6b1n0
  - X1001c1s6b1n1
  - X1001c1s7b0n0
  - X1001c1s7b0n1
  - X1001c1s7b1n0
  - X1001c1s7b1n1
  - X1006c1s4b0n0
  - x1006c1s4b1n0



Simplify node management



Challenges

- User friendly (user can ask resources with minimal effort)
- Maximize hardware utilization:
  - minimize the number of nodes allocated to clusters
  - must fulfill user request
- Resolve a list of hardware requirements to a list of nodes
- User requests x2 Nvidia gpus a100
  - x1005c0s4b0n0: x4 NVIDIA_A100-SXM4-80GB, x1 AMD EPYC 7713 64-Core Processor, 512GiB memory
- User requests x6 Nvidia gpus a100 & AMD epyc cpu
  - x1005c0s4b0n0: x4 NVIDIA_A100-SXM4-80GB, x1AMD EPYC 7713 64-Core Processor, 512GiB memory
  - x1005c0s4b0n1: x4 NVIDIA_A100-SXM4-80GB, x1 AMD EPYC 7713 64-Core Processor, 512GiB memory

Find nodes based on hardware description

- Algorithm based on hardware components (not nodes)
- Hardware quantification
- Atomic (If user request can’t be fulfilled, then operation is canceled and no changes committed)

Concepts

Hardware summary

- Need a way to to describe a group of nodes
- Easy to write
- Hardware summary
  - key value structure with hardware component types and its quantity across a number of nodes
  - <hw component>:<quantity>[:<hw component>:<quantity>]
  - NVIDIA_A100-SXM4-80GB:4:AMD EPYC 7713 64-Core Processor:2:memory:8
- Data taken from HSM hardware inventory
  - Processors
  - Accelerators
  - Memory

Memory

- Can’t use dimm quantity as unit of measurement
- Use Greatest Common Factor across all memory dimms as unit of measurement
- 16GiB

Pool of resources

- Target
  - Runs user workload
  - Can be created from scratch
  - Can already exists
- Parent
  - Free pool of resources
  - Target downscaling returns to parent
  - Target upscaling takes from parent
  - User needs to have access to this pool of resources

Nodes operations

- Upscale cluster
  - Move nodes from parent to target cluster
- Downscale cluster
  - Move nodes from target to parent cluster
- Deltas
  - Parent cluster: AMD epyc 7742:24:NVIDIA_A100:4:AMD INSTINCT:16:AMD epyc 7713:2:AMD epyc 7A13:2
  - Target cluster: NVIDIA_A100:4:AMD epyc 7713:1:AMD epyc 7742:6
  - User request: NVIDIA_A100:8:AMD epyc 7713:2:AMD epyc 7742:2
  - Deltas: NVIDIA_A100:**+4**:AMD epyc 7713:**+1**:AMD epyc 7742:**-4**

Simplify node management

- Cluster tenantA (hw component summary)

HW component	Quantity
AMD EPYC 7742 64-Core Processor	24
NVIDIA_A100-SXM4-80GB	8
Memory (16GiB)	320
AMD EPYC 7713 64-Core Processor	2
AMD EPYC 7A53 64-Core Processor	2
AMD INSTINCT MI200 (MCM) OAM LC	18

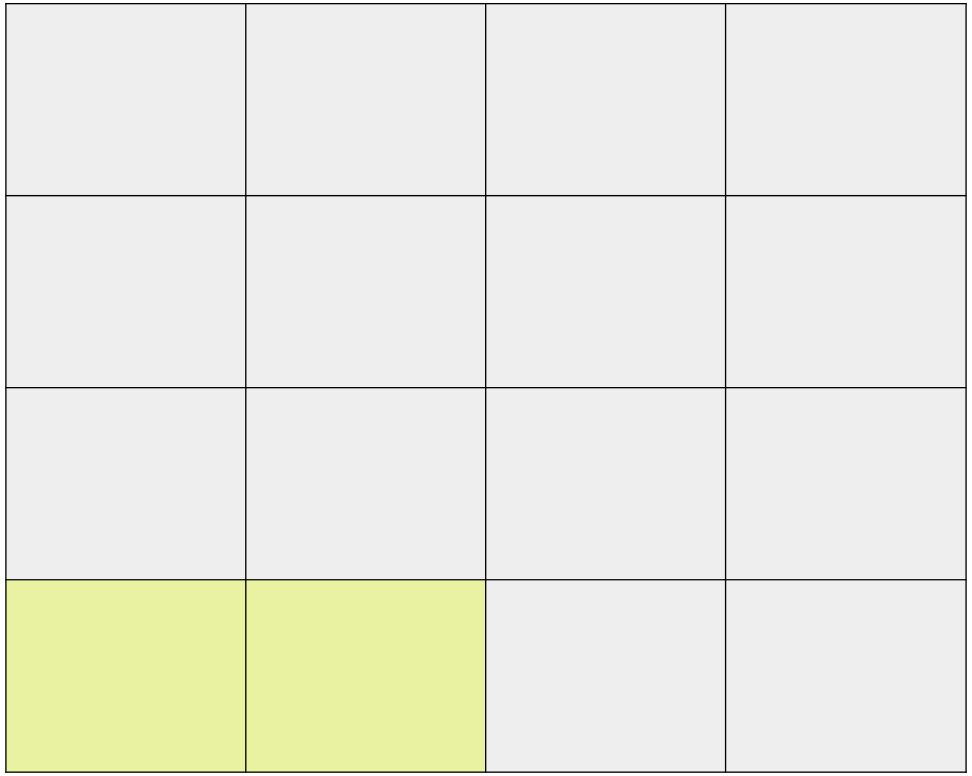
Simplify node management

- Cluster tenantA_AI (hw component summary)

HW component	Quantity
NVIDIA_A100-SXM4-80GB	8
Memory (16GiB)	64
AMD EPYC 7713 64-Core Processor	2

- Cluster tenantA (hw component summary)

HW component	Quantity
AMD EPYC 7742 64-Core Processor	24
Memory (x16GiB)	256
AMD EPYC 7713 64-Core Processor	0
AMD EPYC 7A53 64-Core Processor	2
AMD INSTINCT MI200 (MCM) OAM LC	18
NVIDIA_A100-SXM4-80GB	0



Simplify node management

- Cluster team_CPU (hw component summary)

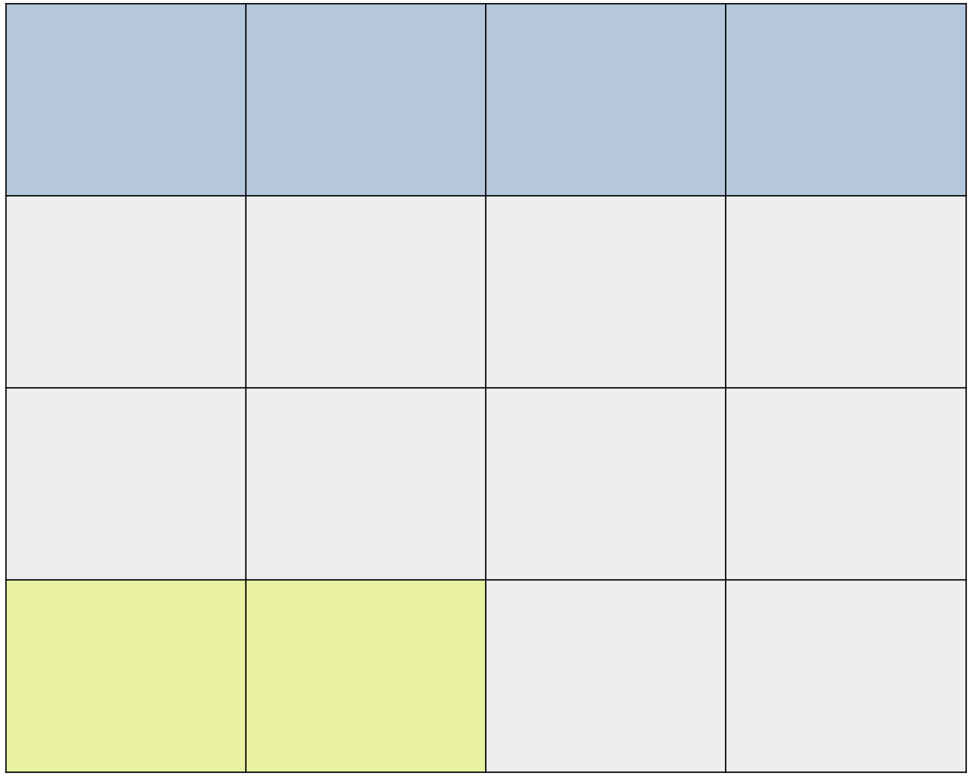
HW component	Quantity
AMD EPYC 7742 64-Core Processor	8
Memory (16GiB)	64

- Cluster tenantA_AI (hw component summary)

HW component	Quantity
NVIDIA_A100-SXM4-80GB	8
Memory (16GiB)	64
AMD EPYC 7713 64-Core Processor	2

- Cluster tenantA (hw component summary)

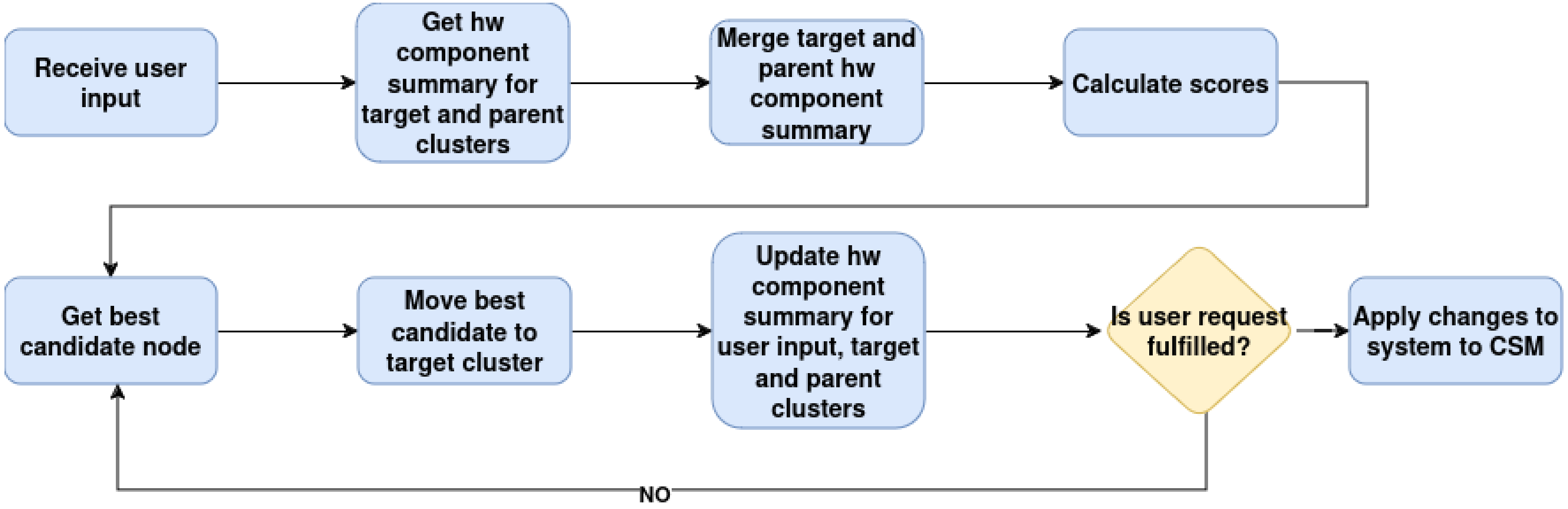
HW component	Quantity
AMD EPYC 7742 64-Core Processor	16
Memory (16GiB)	192
AMD EPYC 7713 64-Core Processor	0
AMD EPYC 7A53 64-Core Processor	2
AMD INSTINCT MI200 (MCM) OAM LC	18
NVIDIA_A100-SXM4-80GB	0



Strategy

- Scarcity
  - Minimize the number of nodes allocated to user request
  - Nodes should be geographically close to each other (minimize fragmentation)
  - Target cluster must contain all hardware requested by user
  - Test/dev clusters

Algorithm


Receive user input
Get hw component summary for target and parent clusters
Merge target and parent hw component summary
Calculate scores
Get best candidate node
Move best candidate to target cluster
Update hw component summary for user input, target and parent clusters
Is user request fulfilled?
Apply changes to system to CSM
NO

User input

- Hardware component summary
  - Intuitive and flexible (no need to specify the whole text like “AMD EPYC 7713 64-CoreProcessor”, keywords like “epyc” should work
  - a100:8:instinct:8:epyc:4
- Target HSM is the cluster running the user workload
  - tenantA_group_AI
- Parent HSM is the cluster where resources are taken or returns from/to target HSM
  - tenantA

Get hardware components

- Get the list of nodes in both target and parent HSM groups through HSM group API in CSM
- Use HSM hardware inventory API in CSM to fetch all hardware components in node list provided above
- Use the information provided by the APIs and create the hardware component summary data structure
- Hardware components are then filtered and grouped based on user input
- List of nodes and hw components
  - x1001c1s5b0n0: epyc:2:memory:16
  - x1001c1s5b0n1: epyc:2:memory:16
  - x1005c0s4b0n0: a100:4:epyc:1:memory:32
  - x1005c0s4b0n1: a100:4:epyc:1:memory:32
  - x1006c1s4b0n0: instinct:8:epyc:1:memory:32
  - x1006c1s4b1n0: instinct:8:epyc:1:memory:32
- Hardware component summary for target cluster
  - "instinct": 16, "a100": 8, "memory": 240, "epyc": 18
- Hardware component summary for parent cluster
  - "a100": 8, "instinct": 16, "epyc": 28, "memory": 320

Merge target and parent hardware component summary

- Merge the hardware component summary structures for target and parent HSM groups into a single structure

Calculate node’s scores

- The goal is to assign a score to each node
- The nodes will be selected one by one based on their score and moved from one cluster to the other
- The node with the highest score is selected to go to the target cluster.
- The node’s score is calculated based on the aggregated scores for each of its hardware components
- The scores are calculated based on the hw components available and requested by the user
- Each hardware components will also have a score, its score is calculated based on scarcity
- If a hardware component is not in the user request, then its score penalizing the node’s score

Calculate node’s scores – hardware component (scarcity) scores

Hw component scarcity score

- The goal is to assign a score to each hardware component the user has access to.
- The lower the quantity the higher the score (scarce)
- Hardware component scarcity score = Total number of hardware components / number of hw components

Eg:

- hw component summary for a set of nodes: "a100": 8, "instinct": 16, "epyc": 28, "memory": 320
- Total number of hardware components: 372
- Hw component scarcity scores:
  - epyc: 372/28 = 13.285714
  - memory: 372/320 = 1.1625
  - instinct: 372/16 = 23.25
  - a100: 372/8 = 46.5

Calculate node's scores

User request → epyc:X:a100:Y:instinct:Z

- X1001c1s5b0n0: epyc:2:memory:16 → 2*13.285714 - 16*1.1625 = 7.971428
- X1005c0s4b0n0: a100:4:epyc:1:memory:32 → 4*46.5 + 13.285714 – 32*1.1625 = 162.08571
- X1006c1s4b0n0: instinct:8:epyc:1:memory:32 → 8*23.25 + 13.285714 – 32*1.1625 = 162.08571

Identify node with highest score and move it to target cluster

- X1005c0s4b0n0: a100:4:epyc:1:memory:32 → 4*46.5 + 13.285714 – 32*1.1625 = 162.08571
- Moves to target cluster

Update user request based on new target cluster

- Iteration 0
  - User request: a100:8:instinct:8:epyc:4
  - Hw component summary: a100: 8, instinct: 16, epyc: 28, memory: 320
- Iteration 1
  - User request: a100:8:instinct:8:epyc:4
  - Best candidate **X1005c0s4b0n0**: a100:4:epyc:1:memory:32
  - New updated user request: a100:4:instinct:8:epyc:3
  - Hw component summary: a100: 4, instinct: 16, epyc: 27, memory: 288
- Iteration 2
  - User request: a100:4:instinct:8:epyc:3
  - Best candidate **X1005c0s4b0n1**: a100:4:epyc:1:memory:32
  - New updated user request: a100:0:instinct:8:epyc:2
  - Hw component summary: a100: 0, instinct: 16, epyc: 26, memory: 256
- Iteration 3
  - User request: a100:0:instinct:8:epyc:2
  - Best candidate **x1006c1s4b0n0**: instinct:8:epyc:1:memory:32
  - New updated user request: a100:0:instinct:0:epyc:1
  - Hw component summary: a100: 0, instinct: 8, epyc: 25, memory: 224
- Iteration 4
  - User request: a100:0:instinct:0:epyc:1
  - Best candidate **x1001c1s5b0n0**: epyc:2:memory:16
  - New updated user request: a100:0:instinct:0:epyc:-1
  - Hw component summary: a100: 0, instinct: 8, epyc: 23, memory: 208

x1001c1s5b0n0: epyc:2:memory:16

x1001c1s5b0n1: epyc:2:memory:16

x1001c1s5b1n0: epyc:2:memory:16

x1001c1s5b1n1: epyc:2:memory:16

x1001c1s6b0n0: epyc:2:memory:16

x1001c1s6b0n1: epyc:2:memory:16

x1001c1s6b1n0: epyc:2:memory:16

x1001c1s6b1n1: epyc:2:memory:16

x1001c1s7b0n0: epyc:2:memory:16

x1001c1s7b0n1: epyc:2:memory:16

x1001c1s7b1n0: epyc:2:memory:16

x1001c1s7b1n1: epyc:2:memory:16

x1005c0s4b0n0: a100:4:epyc:1:memory:32

x1005c0s4b0n1: a100:4:epyc:1:memory:32

x1006c1s4b0n0: instinct:8:epyc:1:memory:32

x1006c1s4b1n0: instinct:8:epyc:1:memory:32

Demo

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Future work

Future work

- Extend SAT file with cluster hardware information
- Add new strategies for production systems:
  - Rack/power information (HA)
  - Node distribution across different chassis or racks (HA)
  - Reuse as much nodes as possible in target pool group (minimize impact on WLM clusters)
- High level abstraction to match workloads with hardware requirements to define clusters
  - Eg: cluster size XXL for AI workload (75% GPU and 25% CPU) → a100:200:epyc:50
- Target specific nodes and reuse as much nodes as possible in target cluster

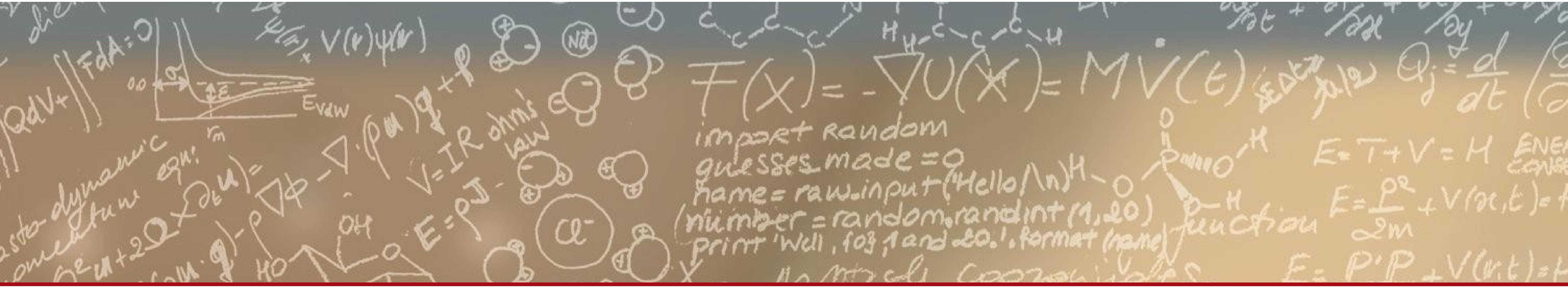
Acknowledge

- Mark Klein
- Miguel Gila
- Maxime Martinasso
- Felipe Cruz
- Matteo Chesi
- Peter Tiernan
- Hussein Harake
- Riccardo Di Maria
- Chris Gamboni
- Victor Holanda
- Marco Induni
- Gennaro Oliva
- Guilherme Peretti-Pezzi
- Derek Feichtinger
- Elsa Germann
- Marc Caubet
- Hans-Nikolai Viessmann

Q&A

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Thank you for your attention.