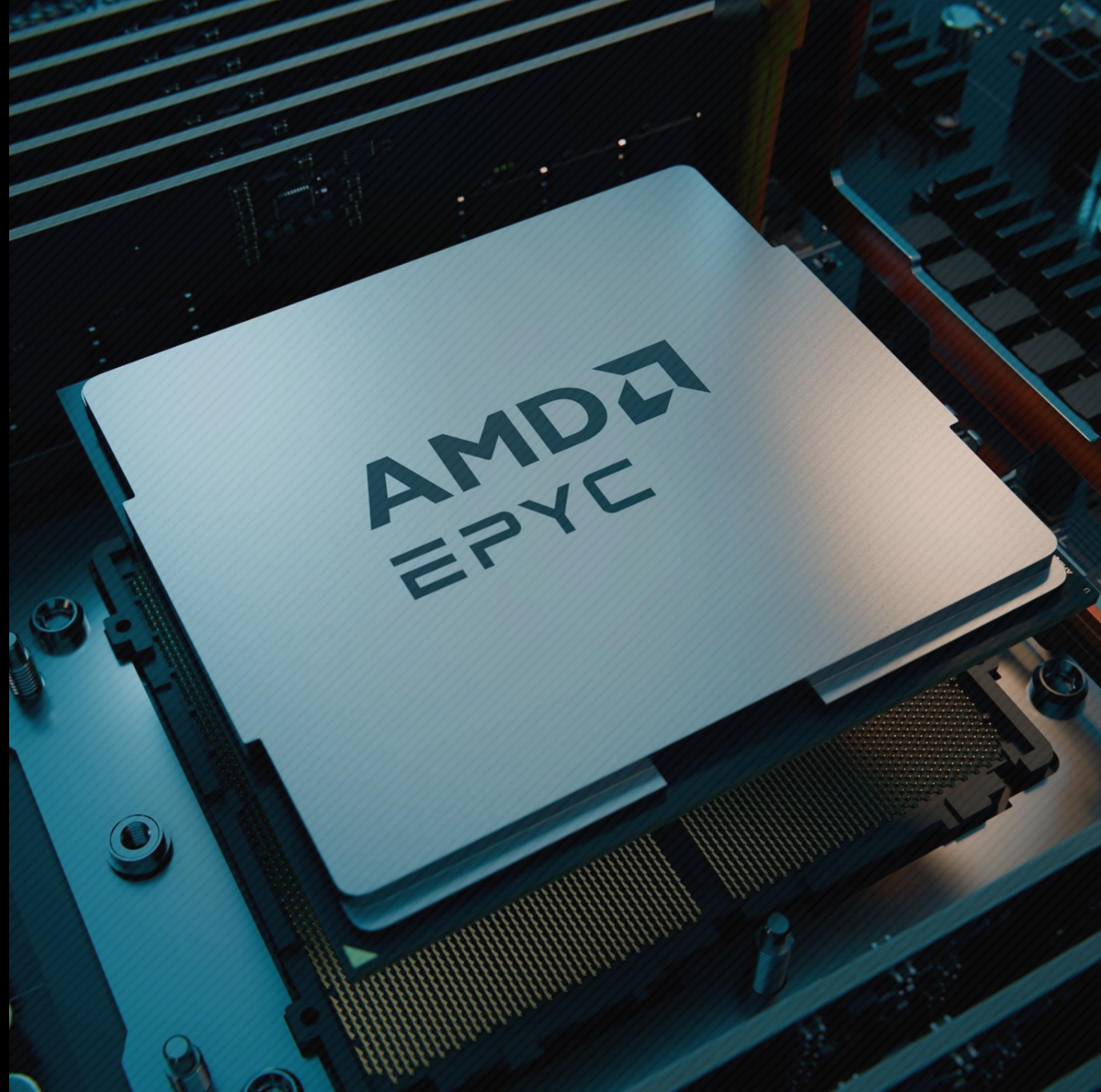
5th Gen AMD EPYC™ CPUs HPC Performance for Slurm Optimization

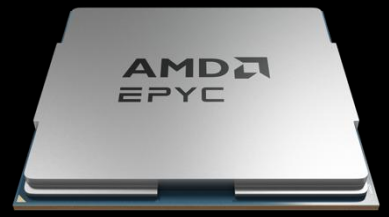
- Miguel Tiempos
- FAE
- México & CAC
- miguel.tiempos@amd.com



AMD AI Platforms

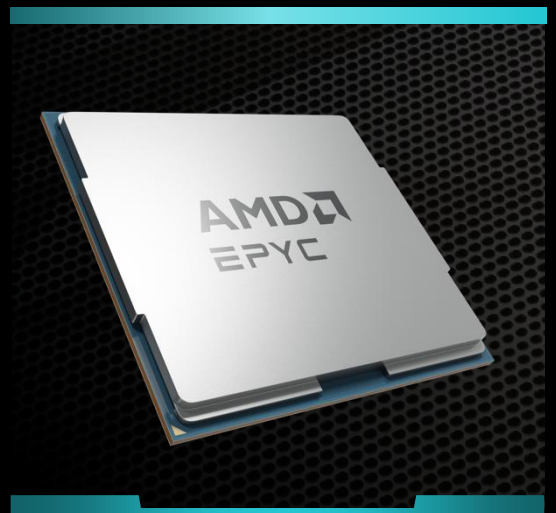
Portafolio de entrenamiento e inferencia.

Data Center | Edge | End Point



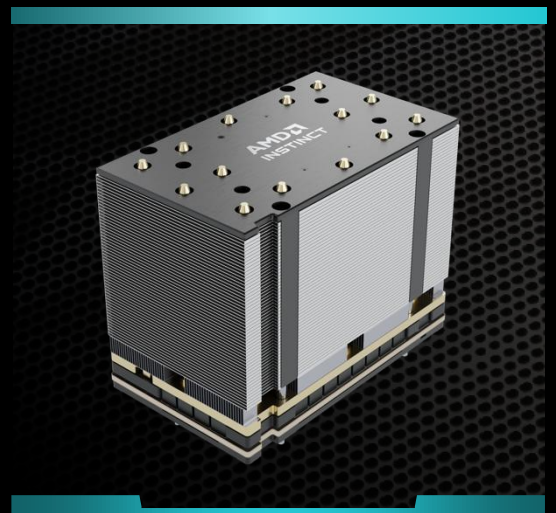
AMD Instinct™	AMD Alveo™	AMD EPYC™	AMD Versal™ AI Edge	AMD Ryzen™ 7040
Aceleradores	Aceleradores	Procesadores	Embedded	Procesadores móviles
Entrenamiento e inferencia sobre HPC y Data Center	Inferencia Edge y Data Center	CPU Lider IA	AI + Sensor Embedded Inferencia	Ryzen™ AI para Windows PCs

Leadership HPC Data Center Solutions



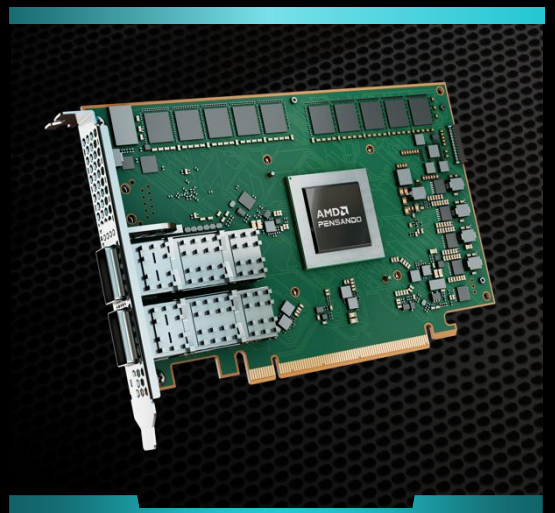
Server CPU Family

AMD EPYC



AI and HPC Accelerator

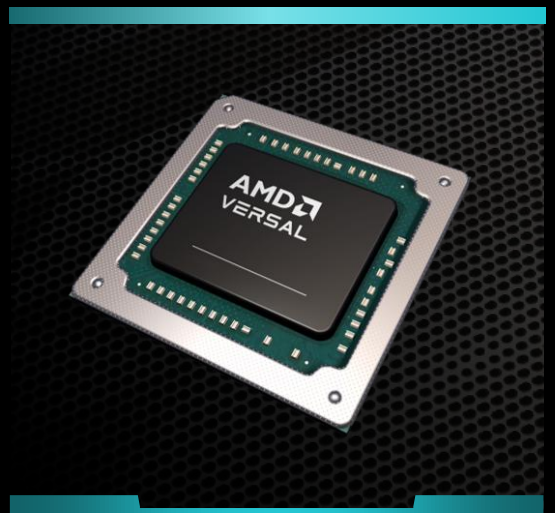
AMD INSTINCT



Networking and DPU

AMD ALVEO

AMD PENSANDO



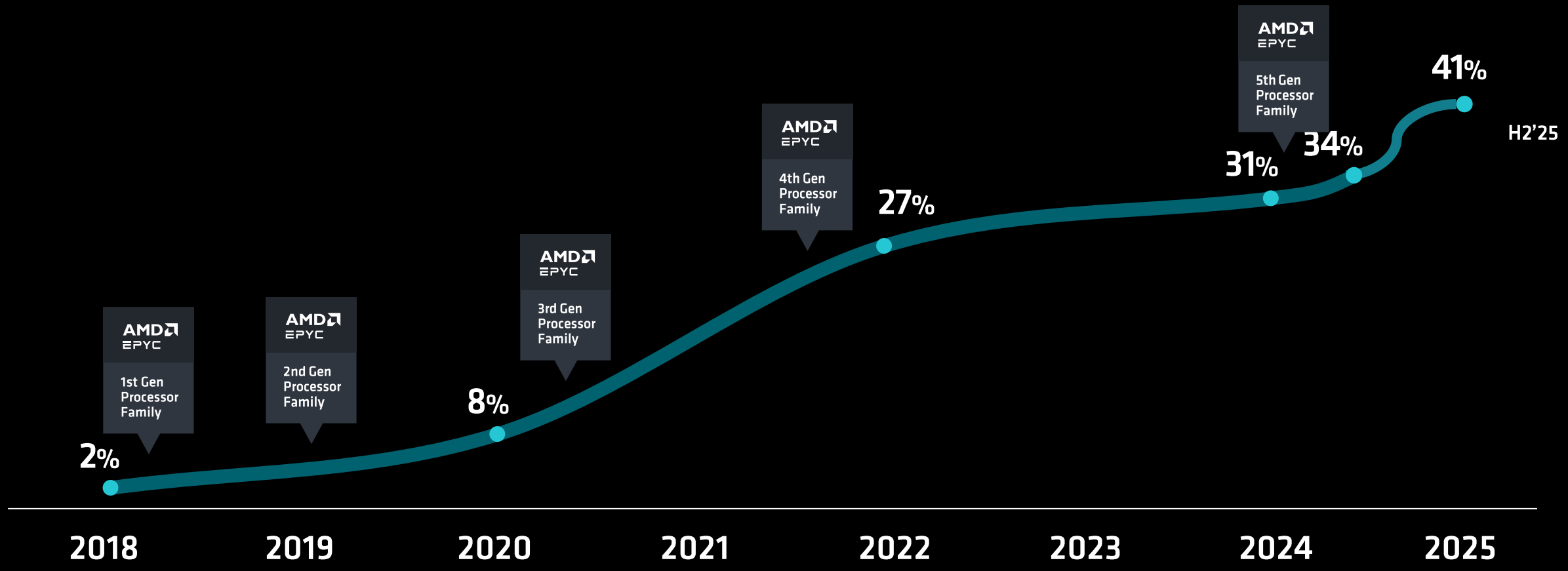
FPGA and Adaptive SoC

AMD VERSAL

AMD ALVEO

AMD delivers the broadest technology portfolio to the HPC data center

AMD EPYC™ Record Market Share…and Growing



>350 OEM Platforms | **>950 Cloud Instances**

Source: Mercury Research Sell-in Revenue Shipment Estimates

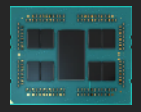
AMD EPYC™ Processors

Five generations of on time technology innovation

1st Gen AMD EPYC™

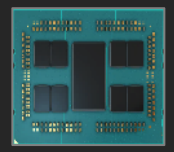
7001
SP3

2nd Gen AMD EPYC™



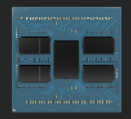
7002
SP3

3rd Gen AMD EPYC™

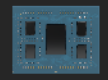


7003
SP3

4th Gen AMD EPYC™



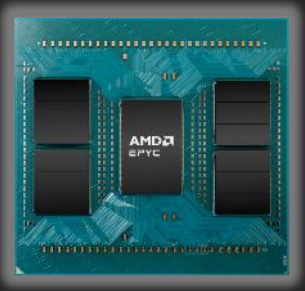
9004
SP5



8004
SP6

4004
AM5

5th Gen AMD EPYC™



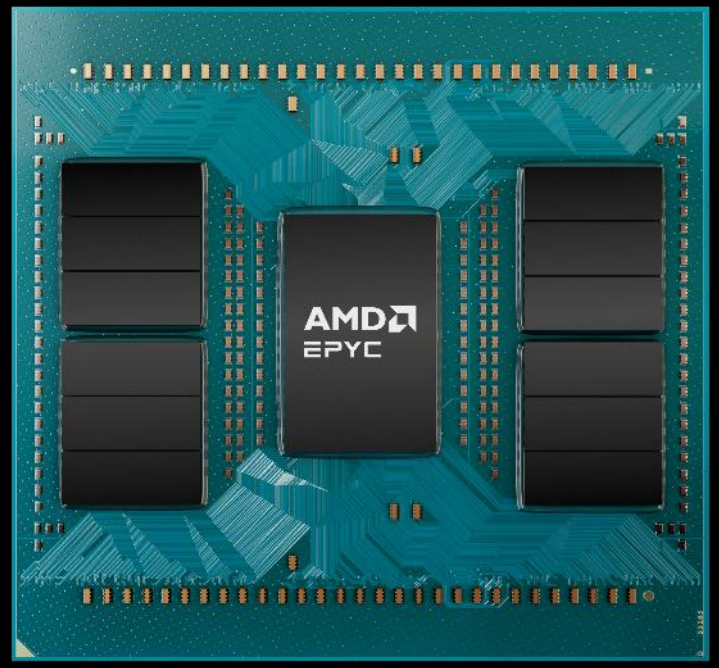
9005
SP5

Future products

All roadmaps are subject to change.

Introducing
5th Gen AMD EPYC™ Processors

Formerly codenamed “Turin”

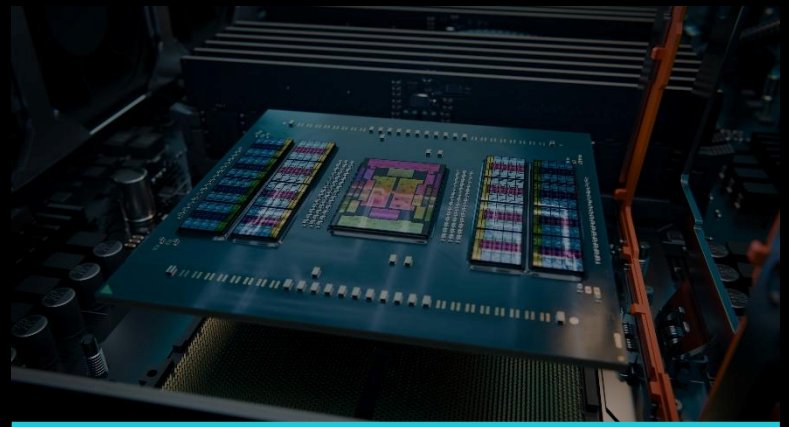


World’s best CPU for cloud, enterprise HPC & AI

ZEN 5 **TSMC 3/4nm**	Up to **192 cores** Up to **384 threads**	Up to **5GHz** **AVX512** full 512b data path	**17% IPC Uplift**	**SP5 Platform** Compatible with “Genoa”

See endnote: 9xx5-001, 048

“Turin” Addressing the Needs of the Modern Data Center



Extending Server CPU Leadership

+

Efficient Modernization

+

End-to-End AI Leadership

Slurm Background and Operation

Slurm was developed at Lawrence Livermore National Laboratory, specifically as a Job Scheduler
Which is the basis for syncing the Jobs launched In our HPC Cluster.
We can prioritize Jobs, configure in what nodes To run these, and control the Queque of the work To be performed.

It is in our best interest to have the best CPU & GPU Performance in order to have the Most efficient environment, this is where AMD Epyc provides the very best in per core Performance, and yields the most powerful HPC ecosystem in order to get more work done Quicker.

Lawrence Livermore National Laboratory

1 - CXL Type 1&2 devices and PCIe link encryption support dependent upon ecosystem readiness
*Standard roadmap offerings on AMD.com support 6000 MT/s
See endnotes 9xx5-048, 072, 083, GD-183A

Slurm Operation squeue -u <username>

root@cns-master-01:~

root@cns-master-01:~ 225x39

```
[root@cns-master-01 ~]# hpc
             JOBID PARTITION                NAME                      USER    STATE       TIME   NODES NODELIST(REASON)
             16709 nodos_fat              test_R       hpc-gibran-anderson  PENDING       0:00       1 (launch failed requeued held)
             16865 nodos_fat              Na-277          hpc-sinhue-lopez  RUNNING 1-19:33:34       1 cns-nodo-03
             16863 nodos_fat              Na-303          hpc-sinhue-lopez  RUNNING 1-19:34:08       1 cns-nodo-02
             16736 nodos_fat               qe-neb     hpc-jonathan-guerrero  RUNNING 3-23:12:06       1 cns-nodo-04
             16478 nodos_fat             octamero       hpc-carlos-frontana  RUNNING 7-06:45:05       1 cns-nodo-06
             16928 nodos_gpu         loxa_7rhy_r4         hpc-carmen-pastor  RUNNING      27:43       1 cns-gpu-01
             16927 nodos_sta          strain_1.05         hpc-juan-martinez  PENDING       0:00       1 (Resources)
             16926 nodos_sta          strain_1.04         hpc-juan-martinez  RUNNING      17:30       1 cns-nodo-15
             16925 nodos_sta          strain_1.03         hpc-juan-martinez  RUNNING      30:21       1 cns-nodo-14
             16920 nodos_sta                 vasp            hpc-ana-torres  RUNNING    2:38:23       1 cns-nodo-16
             16919 nodos_sta                 vasp            hpc-ana-torres  RUNNING    2:38:42       1 cns-nodo-38
             16918 nodos_sta                E-312       hpc-pricila-romero  RUNNING    3:10:22       1 cns-nodo-33
             16917 nodos_sta                E-269       hpc-pricila-romero  RUNNING    3:11:25       1 cns-nodo-25
             16916 nodos_sta                E-271       hpc-pricila-romero  RUNNING    3:12:04       1 cns-nodo-18
             16915 nodos_sta                E-286       hpc-pricila-romero  RUNNING    3:13:20       1 cns-nodo-24
             16903 nodos_sta                 vasp            hpc-ana-torres  RUNNING    5:46:09       1 cns-nodo-37
             16902 nodos_sta                 vasp            hpc-ana-torres  RUNNING    5:46:37       1 cns-nodo-36
             16891 nodos_sta              extend3         hpc-jesus-acosta  RUNNING   19:33:19       1 cns-nodo-42
             16887 nodos_sta             QEGO-1Vac         hpc-juan-aguilera  RUNNING   22:53:56       1 cns-nodo-41
             16867 nodos_sta               g09_aci         hpc-juan-aguilera  RUNNING 1-16:15:21       1 cns-nodo-29
             16866 nodos_sta                Na-291          hpc-sinhue-lopez  RUNNING 1-19:30:06       1 cns-nodo-40
             16864 nodos_sta                Na-283          hpc-sinhue-lopez  RUNNING 1-19:33:38       1 cns-nodo-45
             16731 nodos_sta     20KT255EdfaRi3       hpc-daniel-salgado  RUNNING 3-23:53:30       1 cns-nodo-12
             16730 nodos_sta 20KT2575srFaedRi       hpc-daniel-salgado  RUNNING 4-02:12:11       1 cns-nodo-27
             16729 nodos_sta 20KT2575FaedRi3s       hpc-daniel-salgado  RUNNING 4-02:13:26       1 cns-nodo-26
             16728 nodos_sta 20KT2575srEdfaRi       hpc-daniel-salgado  RUNNING 4-02:14:40       1 cns-nodo-20
             16496 nodos_sta             heptamero       hpc-carlos-frontana  RUNNING 7-05:01:54       1 cns-nodo-34
             16460 nodos_sta               mb2_test       hpc-ricardo-guirado  RUNNING 7-08:53:50       1 cns-nodo-11
             16357 nodos_sta   test_Esm_CAHS13     hpc-esmeralda-escobar  RUNNING 8-01:05:30       1 cns-nodo-32
             16356 nodos_sta   test_Esm_CAHS13     hpc-esmeralda-escobar  RUNNING 8-01:07:59       1 cns-nodo-28
             16355 nodos_sta   test_Esm_CAHS13     hpc-esmeralda-escobar  RUNNING 8-01:09:48       1 cns-nodo-22
             16347 nodos_sta   test_Esm_CAHS13     hpc-esmeralda-escobar  RUNNING 8-03:00:49       1 cns-nodo-23
             16317 nodos_sta             oxy_test       hpc-ricardo-guirado  RUNNING 8-06:02:51       1 cns-nodo-17
[root@cns-master-01 ~]#
```

*Standard roadmap offerings on AMD.com support 6000 MT/s
See endnotes 9xx5-048, 072, 083, GD-183A

Slurm Operation sinfo -p <partition-name>

```
[root@cns-master-01 ~]# sinfo
PARTITION      AVAIL  TIMELIMIT  NODES  STATE NODELIST
all*              up   infinite      1  inval cns-nodo-48
all*              up   infinite      8  down* cns-gpu-04,cns-nodo-[08,13,30-31,35,43,46]
all*              up   infinite      3  drain cns-nodo-[01,07,09]
all*              up   infinite     29    mix cns-gpu-01,cns-nodo-[02-04,06,11-12,14-18,20,22-29,32-34,36-38,40,45]
all*              up   infinite      2  alloc cns-nodo-[41-42]
all*              up   infinite      2   idle cns-gpu-[02-03]
all*              up   infinite      6   down cns-nodo-[05,19,21,39,44,47]
nodos_fat         up   infinite      1  down* cns-nodo-08
nodos_fat         up   infinite      2  drain cns-nodo-[01,07]
nodos_fat         up   infinite      4    mix cns-nodo-[02-04,06]
nodos_fat         up   infinite      1   down cns-nodo-05
nodos_standard    up 15-00:00:0      1  inval cns-nodo-48
nodos_standard    up 15-00:00:0      6  down* cns-nodo-[13,30-31,35,43,46]
nodos_standard    up 15-00:00:0      1  drain cns-nodo-09
nodos_standard    up 15-00:00:0     24    mix cns-nodo-[11-12,14-18,20,22-29,32-34,36-38,40,45]
nodos_standard    up 15-00:00:0      2  alloc cns-nodo-[41-42]
nodos_standard    up 15-00:00:0      5   down cns-nodo-[19,21,39,44,47]
escuela_hpc       up 15-00:00:0      3    mix cns-nodo-[36-37,45]
escuela_hpc       up 15-00:00:0      1   idle cns-gpu-03
escuela_hpc       up 15-00:00:0      1   down cns-nodo-44
nodos_gpu         up 15-00:00:0      1    mix cns-gpu-01
nodos_gpu         up 15-00:00:0      2   idle cns-gpu-[02-03]
[root@cns-master-01 ~]#
```


1 - CXL Type 1&2 devices and PCIe link encryption support dependent upon ecosystem readiness
*Standard roadmap offerings on AMD.com support 6000 MT/s
See endnotes 9xx5-048, 072, 083, GD-183A

Slurm Operation sacct / User Acctivity Jobs Launched

```
root@cns-master-01:~
root@cns-master-01:~ 225x39
[root@cns-master-01 ~]# sacct -o Partition --state=COMPLETED --user=hpc-daniel-salgado -S 2025-06-01 -E 2025-06-30  | sort | uniq -c
      1 ----------
     38
      8  nodos_gpu
      1  Partition
     30 nodos_sta+
[root@cns-master-01 ~]#
```

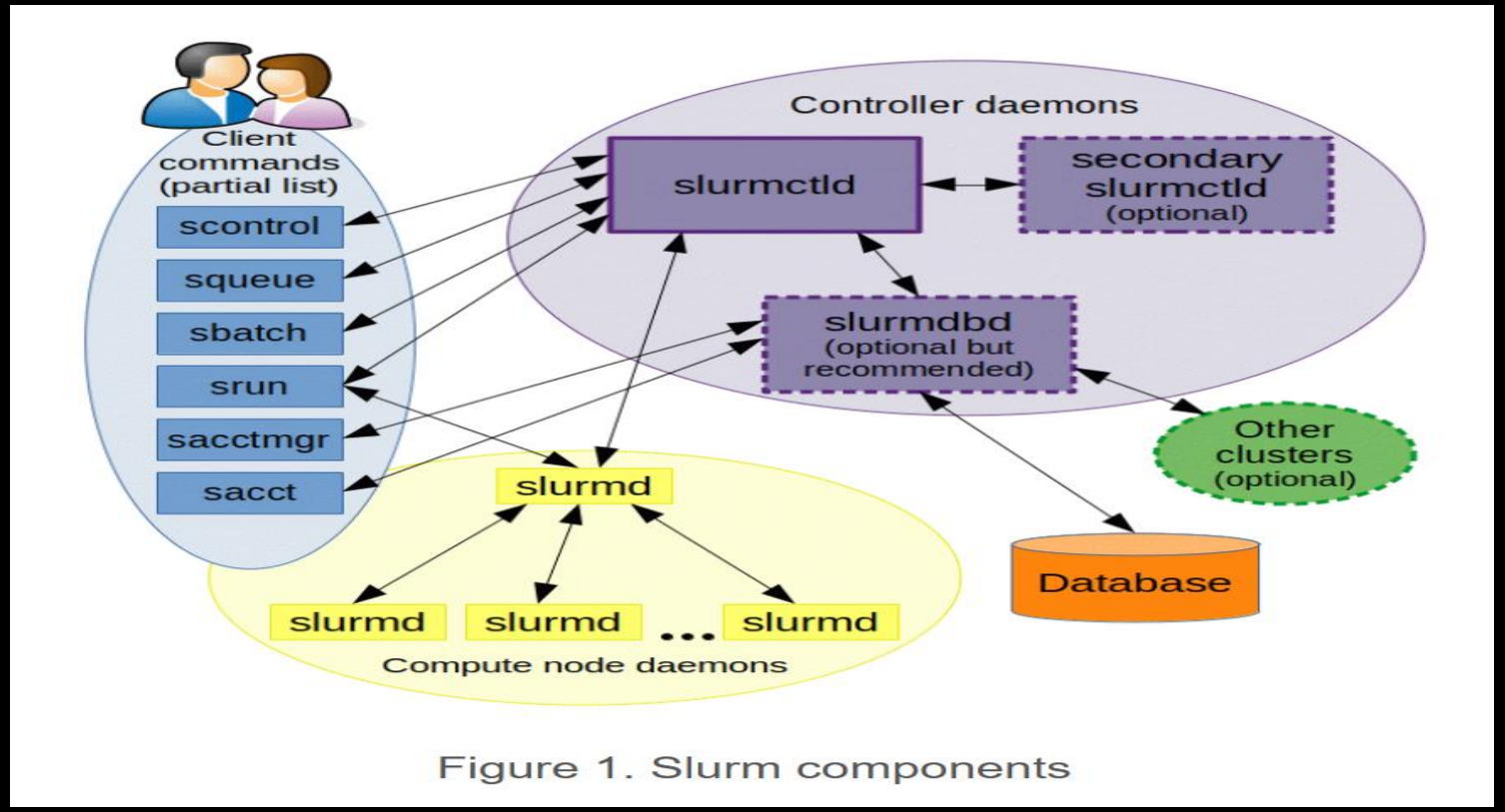
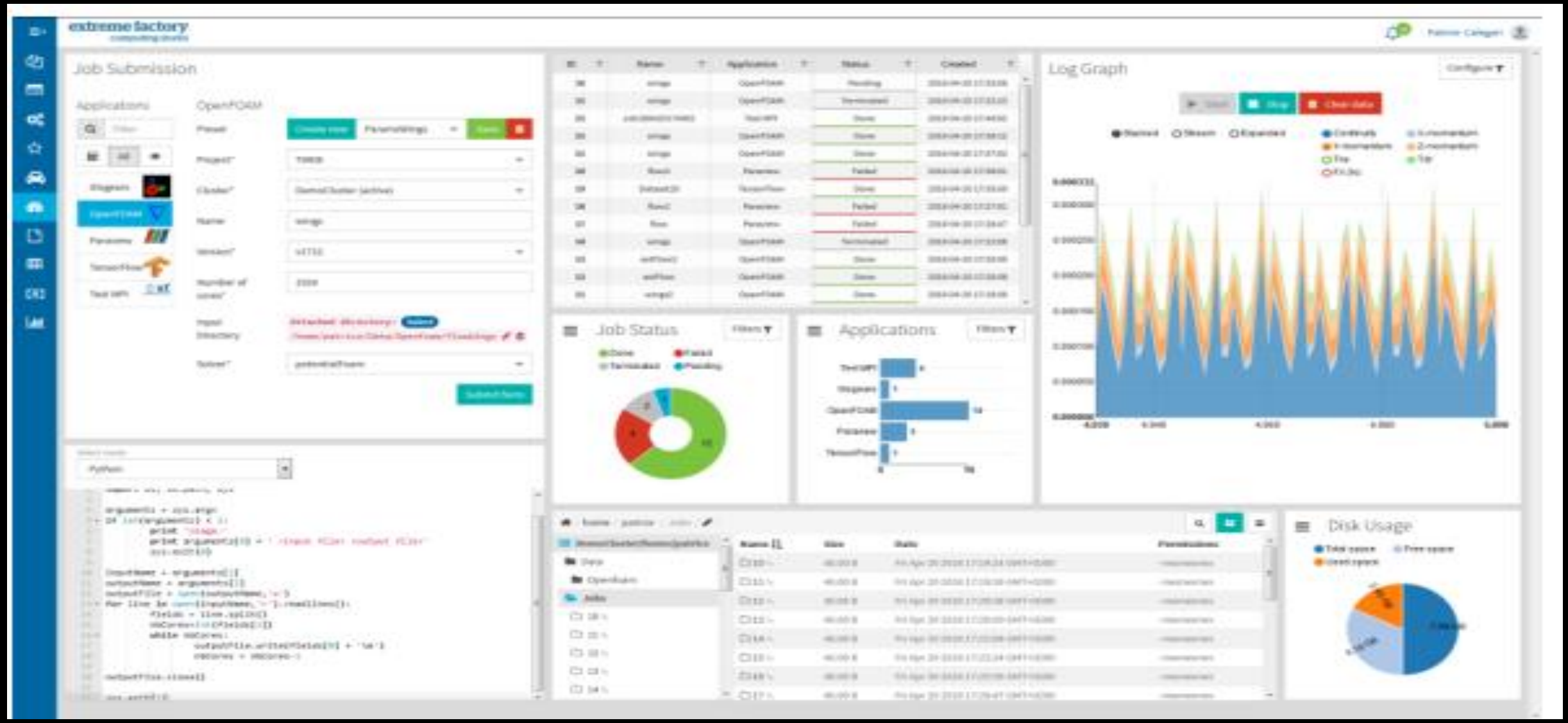


Figure 1. Slurm components

1 - CXL Type 1&2 devices and PCIe link encryption support dependent upon ecosystem readiness
*Standard roadmap offerings on AMD.com support 6000 MT/s
See endnotes 9xx5-048, 072, 083, GD-183A

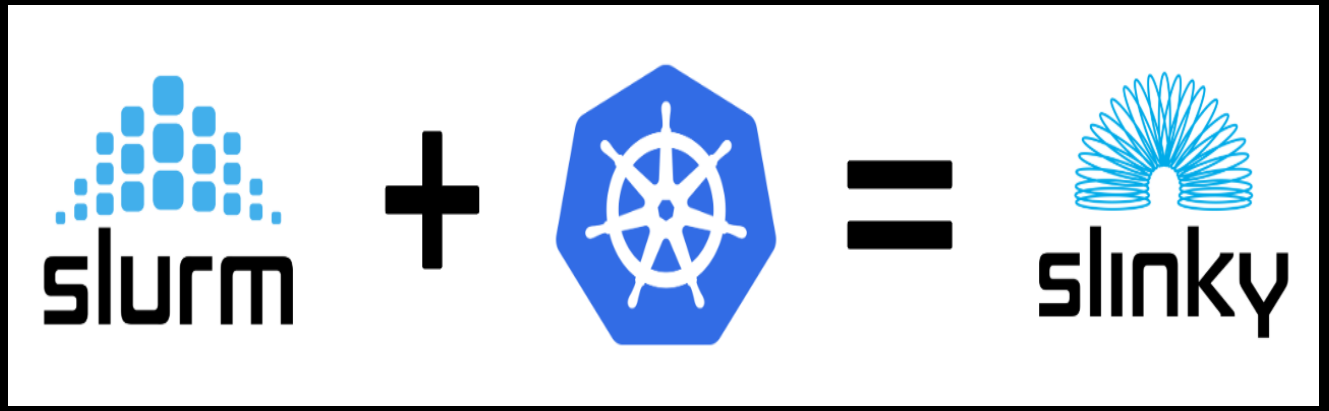
Slurm Operation /Slurm Web



1 - CXL Type 1&2 devices and PCIe link encryption support dependent upon ecosystem readiness
*Standard roadmap offerings on AMD.com support 6000 MT/s
See endnotes 9xx5-048, 072, 083, GD-183A

Slurm Operation /Slurm slinky

With the complement of Kubernetes to Our HPC Slurm Job scheduling, We add a new Set of possibilities in Docker based applications
And AI Workbench options, so that We can Control these through Slurm, with the use of Slinky.



Seamless Hybrid Workload Management

Slinky enables the simultaneous management of HPC workloads using Slurm and containerized applications via Kubernetes on the same infrastructure. This is ideal for organizations running AI/ML training, AI/ML inference, scientific simulations, or data-intensive tasks alongside modern, cloud-native applications. It removes the need to maintain separate clusters,

Dynamic Autoscaling

Slinky's Slurm Operator allows automatic scaling of compute nodes. Slinky dynamically adds or removes Slurm nodes based on workload demands, ensuring that compute resources are efficiently allocated. This leads to reduced overhead costs and ensures that the right amount of infrastructure is available when needed.

Resource Optimization

Slurm Bridge brings the full capacity of Slurm's node packing systems into Kubernetes. This allows for the intelligent scheduling of the right resources at the right time, leading to optimized resource allocation and better coordination between systems.

1 - CXL Type 1&2 devices and PCIe link encryption support dependent upon ecosystem readiness
*Standard roadmap offerings on AMD.com support 6000 MT/s
See endnotes 9xx5-048, 072, 083, GD-183A


AMD Instinct™ MI350 Series GPUs
And deployed by hyperscalers.
AMD


AMD