



SLURM HPC ütemező Openstack felhőben: átmenet a HPC és HTC között



MEZŐ GYÖRGY
Csillagászati és Földtudományi Kutatóközpont
Wigner Adatközpont

Bemutakozás

Mező György



Structured Accretion Disk ERC csoport mérnöke
mezzo.gyorgy@csfk.org



WDC IT csapat tagja
mezzo.gyorgy@wigner.hu



Szuperszámítógép

- ▶ Mi a szuperszámítógép? Mit nevezünk szuperszámítógépnek?

<https://hpc.kifu.hu/hu/gyik>

“A szuperszámítógép egy olyan univerzális eszköz, amivel a nagy számítási igényű, hagyományos számítógéppel, vagy szerver rendszerekkel nem kezelhető feladatokat lehet megoldani. Elsősorban az extrém számítási mennyiséget, sebességet vagy óriási memóriakapacitást igénylő alkalmazások kiszolgálása a cél.”

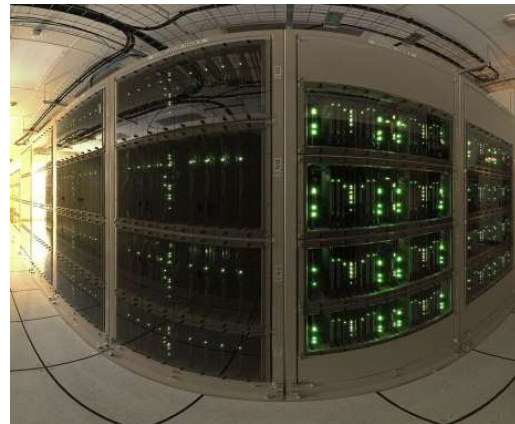
- ▶ Raspberry Pi Cluster ?

<https://glmdev.medium.com/building-a-raspberry-pi-cluster-784f0df9afbd>

SLURM, OpenMPI

Szuperszámítógép

- ▶ ALMA Correlator ?



17 petaOps

2013 TOP 500
Titan 17.59
petaFLOPS

- ▶ GINOP 3.1.5-20-2020-00001 KIFÜ-nek 6 milliárd Ft 2022.október.31-ig 5 petaFLOPS-ért

Komondor Debrecenben 2022 tavaszán indul

<https://hpc.kifu.hu/hu/komondor>

Szuperszámítógép

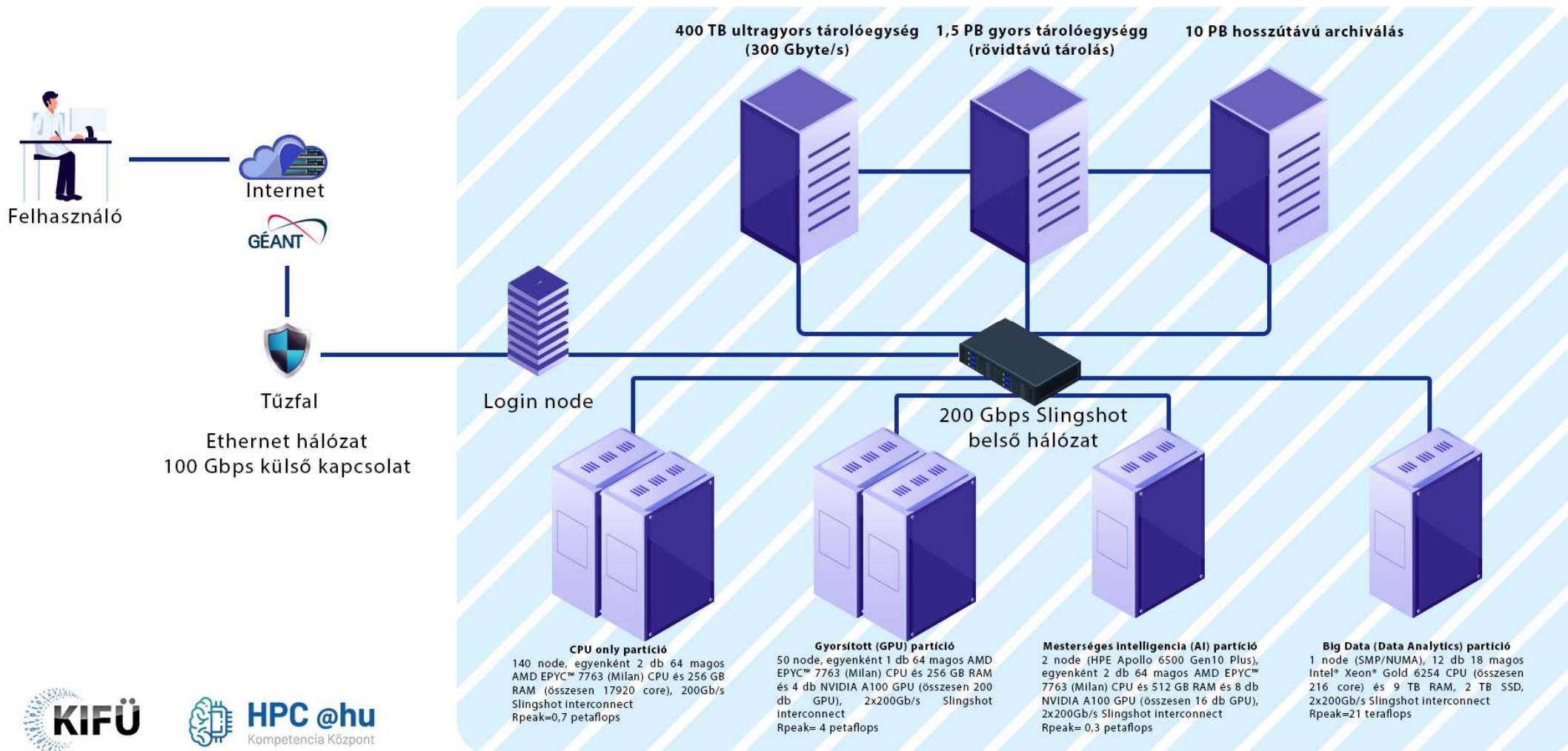
- Jelenleg Magyarországon: <https://wiki.niif.hu/HPC>
Ineterconnect értelmezéséhez: <https://en.wikipedia.org/wiki/InfiniBand>



Szuperszámítógépeink összehasonlítása

Helyszín	Budapest	Budapest2	Szeged	Debrecen	Debrecen2-GPU (Leo)	Debrecen3-Phi (Apollo)	Pécs	Miskolc
Típus	HP CP4000SL	HP SL250s	HP CP4000BL	SGI ICE8400EX	HP SL250s	HP Apollo 8000	SGI UV 1000	SGI UV 2000
CPU-k / node	2	2	4	2	2	2	192	44
Core-ok / CPU	12	10	12	6	8	12	6	8
Memória / node	66 GB	63 GB	132 GB	47 GB	125 GB	125 GB	6 TB	1.4 TB
Memória / core	2.6 GB	3 GB	2.6 GB	2.6 GB	7 GB	5 GB	5 GB	3.75 GB
CPU	AMD Opteron 6174 @ 2.2GHz	Intel Xeon E5-2680 v2 @ 2.80GHz	AMD Opteron 6174 @ 2.2GHz	Intel Xeon X5680 @ 3.33 GHz	Intel Xeon E5-2650 v2 @ 2.60GHz	Intel Xeon E5-2670 v3 @ 2.30GHz	Intel Xeon X7542 @ 2.66 GHz	Intel Xeon E5-4627 v2 @ 3.33 GHz
GPU	-	-	2 * 6 Nvidia M2070	-	68 * 3 Nvidia K20x + 16 * 3 Nvidia K40x	-	-	-
Intel Xeon Phi (KNC)	-	14 * 2 * Intel(R) Xeon Phi(TM) MIC SE10/7120	-	-	-	45 * 2 * Intel(R) Xeon Phi(TM) MIC SE10/7120	-	-
Linpack teljesítmény (Rmax)	5 Tflops	27 Tflops	20 Tflops	18 Tflops	254 Tflops	~106 Tflops	10 Tflops	8 Tflops
Compute node-ok száma	32	14	50	128	84	45	1	1
Dedikált storage	50 TB	500 TB	250 TB	500 TB	585 TB (Phi-vel közös)	585 TB (GPU-val közös)	500 TB	240 TB
Interconnect	IB QDR	IB NB FDR	IB QDR	IB QDR	IB NB FDR	IB NB FDR	Numalink 5	Numalink 6
Scheduler	SLURM	SLURM	SLURM	SLURM	SLURM	SLURM	SLURM	SLURM
MPI	OpenMPI (ompi)	IntelMPI (impi)	OpenMPI (ompi)	SGI MPT (mpt)	OpenMPI (ompi)	OpenMPI (ompi)	SGI MPT (mpt)	SGI MPT (mpt)

A KOMONDOR SZUPERSZÁMÍTÓGÉP FELÉPÍTÉSE



TREND: Felhő szolgáltatók kínálnak HPC szolgáltatásokat

- ▶ Korábban HPC specializált CLOUD szolgáltatók: Rescale, UberCloud ...
- ▶ AMAZON AWS: <https://aws.amazon.com/hpc/> Parallel Cluster 3 2021.09.10.
- ▶ AZURE HPC: [https://azure.microsoft.com/en-us/solutions/high-performance computing/](https://azure.microsoft.com/en-us/solutions/high-performance-computing/)
- ▶ Google Cloud: <https://cloud.google.com/solutions/hpc>
- ▶ Nagy teljesítményű instance-okat, parallel file rendszereket, speciális hálózati megoldásokat kínálnak

The Crossroads of Cloud and HPC: OpenStack for Scientific Research

- ▶ 2017-es tanulmány címe a cím
https://indico.cern.ch/event/570594/contributions/2339816/attachments/1364956/2067316/OpenStack_CloudandHPC6x9BookletandCover_v4_print.pdf
- ▶ ElastiCluster: <https://elasticcluster.github.io/elasticcluster/>
<http://dev.videotorium.hu/hu/recordings/16415/elasticcluster-automated-deployment-and-scaling-of-computing-and-storage-clusters-on-iaas-cloud-infrastructures>
- ▶ StackHpc: <https://www.stackhpc.com/>
- ▶ HPC-ben hardware dedikált kihasználása ↔ OpenStack hardware abstraction
- ▶ OpenStack Cloud nem helyettesíti a dedikált HPC hardware-t !!!

ELKH CLOUD



location	ELKH Cloud - Wigner Compute	ELKH Cloud - Wigner GPU	ELKH Cloud - SZTAKI
type	A+ Server 2124BT-HTR	A+ Server 4124GS-TNR	1029GQ-TVRT
CPU/node	2	2	2
core/CPU	32	48	26
memory/node (GB)	512	1024	768
memory/core (GB)	8	10.666	14.769
CPU	AMD EPYC 7452 @ 2.35 GHz	AMD EPYC 7552 @ 2.2 GHz	Intel Xeon Gold 6230R @ 2.1 GHz
GPU	-	28 × NVIDIA A100 (40 GB)	40 × NVIDIA V100 (32 GB)
Intel Xeon Phi	-	-	-
linpack performance	-	-	-
compute nodes	32	4	10
dedicated storage (TB)	1018		568
interconnect	100 Gbit (Ethernet)	100 Gbit (Ethernet)	100 Gbit (Ethernet)

Milyen lehetőségeket ad az új ütemező az új infrastruktúrával

- ▶ Korábbi kisebb HPC számítások megvalósíthatóak vele
- ▶ Kutatócsoportokkal, kutatókkal együttműködve létrehozhatóak olyan klaszterek, amik korábbi HPC számolásaikat segítik, esetleg megvalósítják
- ▶ Lehetővé teszik a gyors fejlesztést szuperszámítógépekre
- ▶ Gyors egyéni szoftver telepítés
- ▶ Kutatói, kutatócsoporti projektek jó izolációja

SLURM WDC ÁG

- ▶ Jelenleg a régi felhőben csak az adminisztrátorok tudják létrehozni a SLURM referencia alkalmazás segítségével.
- ▶ A SLURM referencia alkalmazás terraform implementációját használjuk, de az Occopus is működik
- ▶ Az új felhőben lehetővé tesszük a felhasználóknak is a felhő orkesztrációs eszközök használatát
- ▶ SLURM 19.05.5, OS Ubuntu 20.04.3, NFS

SLURM WDC ÁG

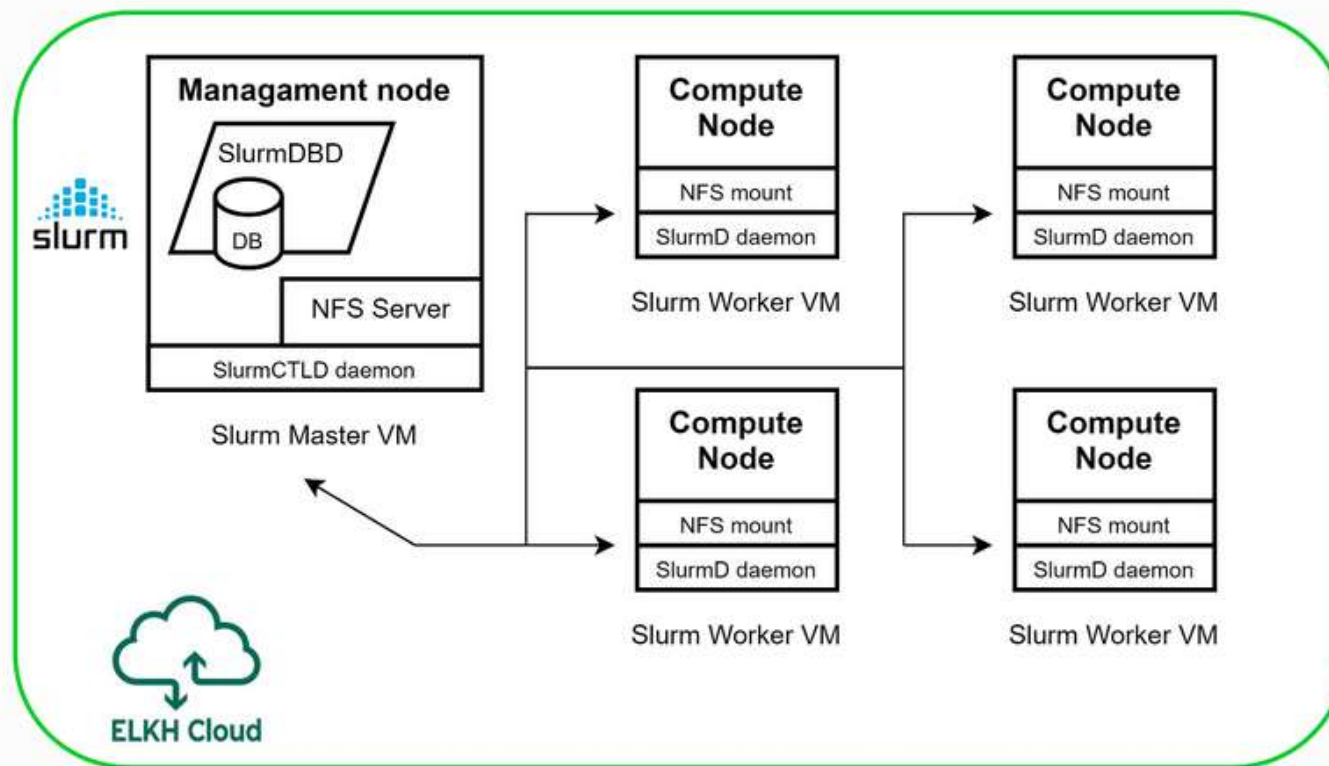


Figure 2. Slurm cluster architecture

SLURM WDC ÁG

- ▶ Terraform telepítés a VPN-en csatlakozó kliens gépen
- ▶ A terraform.tfvars szerkesztése a projektnek megfelelően, terraform init, terraform apply

```
# cloud
tenant_name = "slurm-test"
tenant_id   = "1fc713d937c5487da7072d58bb60d727"
auth_url    = "https://openstack.cloud.local:13000/v3"
user_name   = "mezogy"
user_domain_name = "wdc"
region      = "regionOne"
insecure    = "true"

# master
master_image_id      = "418d14f0-c92f-4a33-a421-01d61c89a060"
master_flavor_name    = "m1.large"
master_key_pair       = "mgy_va"
master_security_groups = ["ping", "default"]
master_network_name   = "slurm"

# worker
worker_image_id      = "418d14f0-c92f-4a33-a421-01d61c89a060"
worker_flavor_name    = "m1.large"
worker_key_pair       = "mgy_va"
worker_security_groups = ["ping", "default"]
worker_network_name   = "slurm"
volume_os_size        = 40
worker_count          = 3
```

SLURM WDC ÁG

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

```
slurm-master = "slurm-master"
slurm-master-ip = "192.168.100.7"
slurm-workers = [
  "slurm-worker-cpu-compute-able-kangaroo",
  "slurm-worker-cpu-compute-finer-ray",
  "slurm-worker-cpu-compute-refined-wren",
]
slurm-workers-ips = [
  "192.168.100.12",
  "192.168.100.14",
  "192.168.100.8",
]
gmezo@screw:~/ceph-test/v3/slurm/terraform-vukics1$ |
```

SLURM WDC ÁG

Instances - OpenStack Dashboard

https://openstack.cloud.local/dashboard/project/instances/

WIGNER DATA CENTER wdc • slurm-test

Project / Compute / Instances

Instances

Displaying 4 items

☐	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Action
☐	slurm-worker-cpu-compute-able-kangaroo	focal-server-cloudimg-amd64-20211022	192.168.100.12	m1.large	mg_ya	Active	nova	None	Running	10 minutes	Launch Instance
☐	slurm-worker-cpu-compute-refined-wren	focal-server-cloudimg-amd64-20211022	192.168.100.8	m1.large	mg_ya	Active	nova	None	Running	10 minutes	Launch Instance
☐	slurm-worker-cpu-compute-finer-ray	focal-server-cloudimg-amd64-20211022	192.168.100.14	m1.large	mg_ya	Active	nova	None	Running	10 minutes	Launch Instance
☐	slurm-master	-	192.168.100.7 Floating IPs: 193.224.192.116	m1.large	mg_ya	Active	nova	None	Running	11 minutes	Launch Instance

Displaying 4 items

SLURM WDC ÁG

```
ubuntu@slurm-master:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE MODELIST
debug*      up    infinite     3   idle slurm-worker-cpu-compute-able-kangaroo,slurm-worker-cpu-c
compute-finer-ray,slurm-worker-cpu-compute-refined-wren
ubuntu@slurm-master:~$ |
```

```
ubuntu@slurm-master:~$ srun --nodes=3 hostname
slurm-worker-cpu-compute-able-kangaroo
slurm-worker-cpu-compute-refined-wren
slurm-worker-cpu-compute-finer-ray
ubuntu@slurm-master:~$ |
```

```
ubuntu@slurm-master:~$ sudo srun --nodes=3 apt install ntpstat
```

```
ubuntu@slurm-master:~$ srun --nodes=3 ntpstat
synchronised to NTP server (148.6.0.1) at stratum 3
  time correct to within 12 ms
  polling server every 64 s
synchronised to NTP server (148.6.0.1) at stratum 3
  time correct to within 11 ms
  polling server every 64 s
synchronised to NTP server (148.6.0.1) at stratum 3
  time correct to within 12 ms
  polling server every 64 s
ubuntu@slurm-master:~$ |
```

SLURM WDC ÁG

```
GNU nano 4.8                                hello_mpi.c
#include <stdio.h>
#include <mpi.h>

int main(int argc, char** argv){
    int node;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &node);
    printf("Hello World from Node %d!\n", node);
    MPI_Finalize();
}
```

<https://glmdev.medium.com/building-a-raspberry-pi-cluster-f5f2446702e8>

```
ubuntu@slurm-worker-cpu-compute-able-kangaroo:/storage/hellompi$ mpicc hello_mpi.c
ubuntu@slurm-worker-cpu-compute-able-kangaroo:/storage/hellompi$ ls
a.out  hello_mpi.c
```

```
#!/bin/bash
cd /storage/openmpi
# Print the node that starts the process
echo "Master node: $(hostname)"
# Run our program using OpenMPI.
# OpenMPI will automatically discover resources from SLURM.
mpirun a.out
```

SLURM WDC ÁG

```
ubuntu@slurm-master:/storage/hellompi$ sbatch --nodes=3 --ntasks-per-node=4 sub_mpi.sh
Submitted batch job 10
ubuntu@slurm-master:/storage/hellompi$ ls
a.out  hello_mpi.c  slurm-10.out  sub_mpi.sh
ubuntu@slurm-master:/storage/hellompi$ more slurm-10.out
Master node: slurm-worker-cpu-compute-able-kangaroo
Hello World from Node 2!
Hello World from Node 3!
Hello World from Node 1!
Hello World from Node 0!
Hello World from Node 9!
Hello World from Node 4!
Hello World from Node 10!
Hello World from Node 5!
Hello World from Node 11!
Hello World from Node 6!
Hello World from Node 8!
Hello World from Node 7!
ubuntu@slurm-master:/storage/hellompi$ |
```

Raspberry Pi klasztertől OpenStack felhőn át Szuperszámítógépekig vezethet út!

Köszönöm a figyelmet!

