



AGENZIA NAZIONALE PER LE
NUOVE TECNOLOGIE, L'ENERGIA E LO
SVILUPPO ECONOMICO SOSTENIBILE

Workshop per l'utilizzo di CRESCO8

Portici 25/09/2025

Matteo Fois TERIN-ICT-HPC



CRESCO8 – Panoramica



Photo by Giuseppe Santomauro & Davide De Chiara (2025)

HPC Tier-0 di ENEA, C.R. Portici (NA)

- 13 racks
- 790 nodi di calcolo (774 CPU + 16 GPU)
- 414 TBytes memoria DDR5
- Rete Mellanox-NDR 200 GB/s
- 10 PBytes storage ad alte prestazioni (DDN/LUSTRE)
- ~7 PFlops potenza di calcolo combinata (5.5 CPU, 1.3 GPU)
- 800 kW potenza elettrica assorbita
- Raffreddamento a liquido CDU+Dry Cooler Adiabatico (95% calore rimosso)
- 10 km cablaggio

CRESCO8 – Hardware

- **Partizione CPU:** 758 nodi (+2 frontend), 2 socket da 64 core Intel(R) Xeon(R) Platinum 8592+, frequenza di clock 1.9-3.8GHz, 512 GB DDR5.
- **Partizione HBM:** 16 nodi, 2 socket da 56 core Intel(R) Xeon(R) Max 9480, frequenza di clock 1.9-3.5GHz, **1 TB DDR5 + 128 GB HBM**
- **Partizione GPU:** 16 nodi (+1 frontend), 2 socket da 64 core Intel(R) Xeon(R) Platinum 8592+, frequenza di clock 1.9-3.8GHz, 512 GB DDR5, **4 GPU Intel(R) Data Center GPU Max 1550** (Ponte Vecchio), con 128 Xe core e 128 GB HBM2e suddivisi su due unità di calcolo (tiles)
- **Partizione Grafica:** 2 nodi, 2 socket da 64 core Intel(R) Xeon(R) Platinum 8592+, frequenza di clock 1.9-3.8GHz, **2 TB DDR5, 2 x NVIDIA RTX A2000 12 GB.**
- **Partizione FPGA:** 4 nodi, Coming soon!

CRESCO8 – Software

- Sistema Operativo: RHEL 9.4 (Kernel Linux 5.14.0)
- Compilatori: GCC 11.4/15.1, Intel oneAPI 2023.2/2025.1, PGI 25.5
- MPI: OpenMPI GCC/Intel 4.1.7, IntelMPI 2023/2025
- Librerie Scientifiche: Intel MKL, OpenBLAS, fftw3, HDF5, (p)netcdf, petsc, blitz
- Applicazioni: cp2k, gromacs, lammps, quantum-espresso
- Job Scheduler: SLURM 23.11.5 (+ patch per integrazione con Kerberos/AFS)
- Filesystem: OpenAFS 1.8.13 (home utenti + software), LUSTRE 2.14 (aree scratch/store utenti + aree progetto)

CRESCO8 – Moduli

La personalizzazione dell'ambiente di lavoro (compilatori, librerie, flavour mpi ecc.) avviene tramite i modulefiles (come in CRESCO6, CRESCO7 e XCRESCO).

- Mostra i moduli attualmente in uso: *module list*
- Mostra i moduli disponibili: *module avail*
- Mostra informazioni sul contenuto di un modulo: *module show <module_name>*
es: *module show intel/oneapi2025*
- Carica un modulo nell'ambiente: *module load <module_name>*
es: *module load intel/oneapi2025*
- Toglie un modulo dall'ambiente: *module unload <module_name>*
es: *module unload mpi_flavour/openmpi_intel-4.1.7*

CRESCO8 – Aree Storage

- Aree utente su filesystem ad alte prestazioni LUSTRE:
 - ~/PFS/lpor → /lpor1/store_0/usr/<username> (Area Store, 100GB)
 - ~/PFS/ltmp → /lpor1/scratch_1/usr/<username> (Area Scratch, No Lim.)
- Controllo quota area store: `lustre_usr_quota_cresco8 <username>`
- Migrazione dati aree store e aree progettuali da GPFS a LUSTRE completata a inizio luglio 2025, sync periodico attivo fino al 30 settembre 2025 (dettagli su [documentazione cresco](#))
- Backup aree store e progetti: Coming soon!
- Policy di cancellazione file aree scratch: Coming soon!
- Trasferimento dati ad aree scratch/store LUSTRE, GPFS (temporaneo), AFS:
 - `cresco-io.portici.enea.it` (2 nodi in round-robin)
 - es: `scp file.dat username@cresco-io.portici.enea.it:/lpor1/store_0/usr/username/`*

CRESCO8 – Accesso al Cluster

- Accesso SSH diretto (dentro rete ENEA):
 - cresco8x001.portici.enea.it
 - cresco8x002.portici.enea.it
 - cresco8-gpu01.portici.enea.it (nodo con GPU)
es: ssh username@cresco8x001.portici.enea.it
- Accesso SSH da rete esterna:
 - cresco-in.portici.enea.it (numeri 5,6,7 nella lista)
 - cresco-in.frascati.enea.it (numeri 5,6,7 nella lista)
es: ssh username@cresco-in.portici.enea.it
- Accesso grafico Open-OnDemand (per ora dentro rete ENEA) sul sito:
 - <http://cresco-ondemand.portici.enea.it>

CRESCO8 – Partizioni SLURM

- Partizioni SLURM per sottomissione ed esecuzione di job:
 - cresco8_cpu (754 nodi, max 250 per utente, 24h)
 - cresco8_cpu_dbg (4 nodi, max 2 per utente, 1h)
 - cresco8_gpu (14 nodi, max 5 per utente, 24h)
 - cresco8_gpu_dbg (2 nodi, max 2 per utente, 1h)
 - cresco8_hbm (14 nodi, max 5 per utente, 24h)
 - cresco8_hbm_dbg (2 nodi, max 2 per utente, 1h)
 - cresco8_pp (2 nodi, per sessioni grafiche openondemand, 24h)
 - cresco8_fpga (4 nodi, max 2 per utente, 24h, coming soon)

CRESCO8 – Comandi SLURM: sinfo

- Stato dei nodi del cluster: *sinfo*

```
<matfois@cresco8x001 ~> sinfo
PARTITION      AVAIL  TIMELIMIT  NODES  STATE NODELIST
cresco7*       up 1-00:00:00      8   mix cresco7x[110-117]
cresco7*       up 1-00:00:00     16  alloc cresco7x[028-037,105-109,133]
cresco7*       up 1-00:00:00     89   idle cresco7x[003-027,038-100,118]
CLIMAHRM       up   infinite     24  alloc cresco7x[119-132,134-143]
CLIMAHRM       up   infinite      1   idle cresco7x144
forair         up   infinite      4   idle cresco7x[101-104]
cresco8_cpu    up 1-00:00:00    306 down* cresco8x[451-756]
cresco8_cpu    up 1-00:00:00     36   mix cresco8x[009-022,086-099,193-200]
cresco8_cpu    up 1-00:00:00    104  alloc cresco8x[003-008,023-028,082,100-147,201-243]
cresco8_cpu    up 1-00:00:00    308   idle cresco8x[029-081,083-085,148-192,244-450]
cresco8_cpu_dbg up    1:00:00      4   idle cresco8x[757-760]
cresco8_gpu    up 1-00:00:00     14   idle cresco8-gpu[02-15]
cresco8_gpu_dbg up    1:00:00      2 down~ cresco8-gpu[16-17]
cresco8_hbm    up 1-00:00:00     14   idle cresco8-hbm[01-14]
cresco8_hbm_dbg up    1:00:00      2   idle cresco8-hbm[15-16]
cresco8_pp     up 1-00:00:00      2   idle cresco8-pp[1-2]
```

CRESCO8 – Comandi SLURM: sinfo

- PARTITION: nome della partizione SLURM
- AVAIL: disponibilità della partizione (up/down)
- TIMELIMIT: tempo limite per l'esecuzione di un job
- NODES: numero di nodi nella partizione
- STATE: stato dei nodi (idle: a riposo, alloc: completamente allocato, mix: parzialmente allocato, drain: chiuso per manutenzione, down: spento)
- NODELIST: lista dei nodi nella partizione

CRESCO8 – Comandi SLURM: *squeue*

- Stato dei job in coda e in esecuzione: *squeue*

```
<matfois@cresco8x001 ~> squeue
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(Reason)
940918	CLIMAHRM	IT3-2	manton	PD	0:00	25	(Dependency)
940916	CLIMAHRM	IT3	manton	R	2-14:48:32	24	cresco7x[119-132,134-143]
992849	cresco7	isfnt_ba	dongiova	R	23:12	8	cresco7x[110-117]
991718	cresco7	FLUENT	dinardo	R	34:15	5	cresco7x[105-109]
964553	cresco7	MTXM	funel	R	4:30:52	10	cresco7x[028-037]
992472	cresco7	despicco	fcichock	R	25:57	1	cresco7x133
994918	cresco8_c	napo0910	medsimf	R	5:20	1	cresco8x082
964714	cresco8_c	blastx_r	daddiego	R	22:27:36	1	cresco8x097
964720	cresco8_c	blastx_r	daddiego	R	21:28:39	8	cresco8x[193-200]
964719	cresco8_c	blastx_r	daddiego	R	21:35:35	3	cresco8x[082,098-099]
972757	cresco8_c	ERA5_196	anavf	R	10:49:43	14	cresco8x[009-022]
972758	cresco8_c	EC_1962	anavf	R	8:57:32	14	cresco8x[009-022]
972759	cresco8_c	MPI_1962	anavf	R	7:03:32	14	cresco8x[086-099]
972761	cresco8_c	HY_ATTIC	cimini	R	6:26:51	6	cresco8x[023-028]
972760	cresco8_c	HY_LIFT	cimini	R	6:26:54	6	cresco8x[003-008]
972763	cresco8_c	RMQL	dcecere	R	6:25:33	43	cresco8x[201-243]
972762	cresco8_c	NH	dcecere	R	6:25:43	48	cresco8x[100-147]

CRESCO8 – Comandi SLURM: squeue

- JOBID: numero identificativo del job
- PARTITION: partizione SLURM su cui è stato sottomesso il job
- NAME: nome del job
- USER: utente che ha sottomesso il job
- ST: stato del job (R: in esecuzione, PD: in attesa di risorse, CG: in completamento)
- TIME: tempo di esecuzione del job
- NODES: numero di nodi impegnati dal job
- NODELIST (REASON): lista dei nodi impegnati o motivo per cui il job è in attesa (es. Dependency: job con dipendenze, Resources: risorse di calcolo non disponibili)

CRESCO8 – sinfo/queue: argomenti utili

Alcuni argomenti utili da aggiungere a *sinfo* e *queue*:

- Mostra lo stato dei nodi di una particolare partizione:
 - *sinfo --partition cresco8_cpu*
- Mostra job in coda o in esecuzione in una particolare partizione
 - *queue --partition cresco8_gpu*
- Mostra job in coda o in esecuzione sottomessi da un particolare utente
 - *queue --user <username>*
- Mostra job in coda o in esecuzione sottomessi dalla propria utenza
 - *queue --me*

CRESCO8 – Job interattivi: srun

Il comando *srun* permette di eseguire operazioni in parallelo su uno o più nodi di calcolo in maniera interattiva. Alcuni esempi:

- Lancia su un nodo un singolo processo che esegue il comando *hostname*:
 - *srun --partition=cresco8_cpu hostname*
- Lancia su due nodi un singolo processo che esegue il comando *hostname*:
 - *srun --nodes=2 --partition=cresco8_cpu hostname*
- Lancia su quattro nodi due processi per nodo che eseguono il comando *hostname*:
 - *srun --nodes=4 --ntasks-per-node=2 --partition=cresco8_cpu hostname*
- Riserva due nodi per un'ora e lancia una shell bash sul primo:
 - *srun --time=01:00:00 --nodes=2 --partition=cresco8_cpu --pty /bin/bash*

CRESCO8 – srun: argomenti utili

Argomento	Effetto
--partition=<name>	Specifica la partizione su cui lanciare il job
--nodes=<number>	Richiede <number> nodi di calcolo
--ntasks=<number>	Richiede il lancio di <number> processi
--ntasks-per-node=<number>	Richiede il lancio di max <number> processi su ogni nodo
--ntasks-per-socket=<number>	Richiede il lancio di max <number> processi su ogni socket
--cpus-per-task=<number>	Richiede <number> core da assegnare ad ogni processo
--mem-per-cpu=<number>	Richiede <number> memoria da riservare per ogni core
--gres=gpu:<number>	Richiede l'uso di <number> GPU su ogni nodo
--time=<time>	Setta un limite di tempo al job (HH:MM:SS)

- [Lista completa degli argomenti.](#)

CRESCO8 – Job non interattivi: sbatch

Il comando *sbatch*, seguito dal percorso ad uno script, sottomette lo script alla coda di SLURM, che lo eseguirà non appena vi saranno risorse disponibili. Gli argomenti per la richiesta di risorse possono essere per convenienza inseriti nello script, preceduti dalla direttiva “*#SBATCH*”.

```
#!/bin/bash

#SBATCH --job-name=myjobname      # nome del job
#SBATCH --partition=cresco8_cpu   # partizione
#SBATCH --output=job.out          # percorso file di output (stdout)
#SBATCH --error=job.err           # percorso file di errori (stderr)
#SBATCH --time=1:00:00            # massimo tempo di esecuzione
#SBATCH --nodes=2                 # numero di nodi
#SBATCH --ntasks-per-node=128     # numero di processi per nodo

mpirun hostname
```

- *sbatch script.sh* (sottomette lo script e stampa l’ID associato al job)

CRESCO8 – sbatch: argomenti utili

Il comando *sbatch* supporta gli stessi argomenti del comando *srun*, alcuni dei quali utili per gestire job non interattivi. Esempi:

Argomento	Effetto
--mail-type=<type>	Condizioni per inviare una email (BEGIN/END/FAIL/ALL)
--mail-user=<mail>	Indirizzo a cui inviare le email
--requeue	Rimette in coda il job in caso di problemi ai nodi
--exclusive	Richiede l'uso esclusivo dei nodi di calcolo assegnati
--parsable	Stampa esclusivamente il JobID alla sottimissione
--job-name=<name>	Nome del job, compare nell'output di <i>squeue</i>
--output=<filename>	File dove scrivere l'stdout dei processi
--error=<filename>	File dove scrivere l'stderr dei processi

- [Lista completa degli argomenti.](#)

CRESCO8 – sbatch: file di output

- Di default SLURM nomina i file output ed error del job nel seguente formato (“%j” viene sostituito dall’ID del job):
 - slurm-%j.out (file con stdout)
 - slurm-%j.err (file con stderr)
- E’ possibile richiedere un formato differente specificando i seguenti caretteri nel pattern del nome file degli argomenti “--output” e “--error”:
 - %j (ID del job)
 - %x (nome del job)
 - %n (nome del nodo di calcolo, crea un file di output per ogni nodo)
 - %t (task specifico del job, crea un file di output per ogni task)

es: *sbatch --output="%x_%j.out" --error="%x_%j.err"*
- E’ possibile specificare un percorso assoluto o relativo nei nomi file, se relativo SLURM crea i file sotto la cartella da cui viene eseguito il comando *sbatch*.

CRESCO8 – sbatch: monitoraggio dei job

- I job vengono eseguiti da SLURM tramite una serie di **step**:
 - **batch step**: viene eseguito sul nodo principale (il primo della lista di nodi assegnata), corrisponde al primo lancio dello script sottomesso con *sbatch*.
 - **normal steps**: corrispondono a chiamate *srun* o *mpirun* all'interno dello script sottomesso con *sbatch*, sono contrassegnati da indici (0,1 ecc.).
- Per controllare in tempo reale l'andamento del job si può usare il seguente comando:
 - *sattach <job_id>.<job_step>* (es: *sattach 12345.0*)
- In alternativa si può usare tail sui file di output ed error:
 - *tail -f <output_file>*

NB: Se i file vengono scritti sul filesystem AFS, l'output non viene mostrato in tempo reale per via del meccanismo di cache del filesystem.

CRESCO8 – sbatch: variabili d'ambiente

Al momento dell'esecuzione di un job non interattivo, SLURM esporta alcune variabili d'ambiente che possono essere usate nello script di sottomissione. Alcuni esempi:

Variabile	Descrizione
SLURM_JOB_ID	ID del job
SLURM_JOB_NAME	Nome del job
SLURM_JOB_NODELIST	Lista dei nodi su cui viene eseguito il job
SLURM_JOB_NUM_NODES	Numero di nodi su cui viene eseguito il job
SLURM_CPUS_ON_NODE	Numero di core su ogni nodo
SLURM_NTASKS	Numero totale di task del job
SLURM_NODEID	Indice relativo del nodo su cui viene eseguito lo script
SLURM_PROCID	Indice relativo del task (o rango MPI) che viene eseguito

- [Lista completa delle variabili.](#)

CRESCO8 – sbatch: job seriale

- Lancia un singolo processo su un singolo nodo

```
#!/bin/bash

#SBATCH --job-name=job_seriale      # nome del job
#SBATCH --partition=cresco8_cpu      # partizione
#SBATCH --output=%x-%j.out          # percorso file di output (stdout)
#SBATCH --error=%x-%j.err           # percorso file di errori (stderr)
#SBATCH --nodes=1                   # numero di nodi
#SBATCH --ntasks-per-node=1         # numero di processi per nodo

./program.exe
```

CRESCO8 – sbatch: job MPI

- Lancia 1280 processi distribuiti su 10 nodi, 128 per nodo

```
#!/bin/bash

#SBATCH --job-name=job_mpi           # nome del job
#SBATCH --partition=cresco8_cpu      # partizione
#SBATCH --output=%x-%j.out           # percorso file di output (stdout)
#SBATCH --error=%x-%j.err            # percorso file di errori (stderr)
#SBATCH --nodes=10                   # numero di nodi
#SBATCH --ntasks-per-node=128        # numero di processi per nodo

mpirun ./program.exe                 # slurm si occupa di passare ad mpi
                                     # le flag corrette per il lancio dei
                                     # processi
```

CRESCO8 – sbatch: job MPI con uso GPU

- Lancia 8 processi su un nodo, riservando l'uso di 8 GPU (NB: l'uso effettivo delle GPU dipende dal software utilizzato per le simulazioni, non tutti hanno supporto per GPU Intel)

```
#!/bin/bash

#SBATCH --job-name=job_gpu           # nome del job
#SBATCH --partition=cresco8_gpu      # partizione
#SBATCH --output=%x-%j.out           # percorso file di output (stdout)
#SBATCH --error=%x-%j.err            # percorso file di errori (stderr)
#SBATCH --nodes=1                    # numero di nodi
#SBATCH --ntasks-per-node=8          # numero di processi per nodo
#SBATCH --gres=gpu:8                 # numero di GPU richieste

mpirun ./program.exe
```

CRESCO8 – sbatch: job ibrido MPI-OpenMP

- Lancia 4 processi distribuiti su 2 nodi, 2 per nodo, 1 per socket, ognuno con 64 thread OMP

```
#!/bin/bash

#SBATCH --job-name=job_ibrido      # nome del job
#SBATCH --partition=cresco8_cpu    # partizione
#SBATCH --output=%x-%j.out         # percorso file di output (stdout)
#SBATCH --error=%x-%j.err          # percorso file di errori (stderr)
#SBATCH --nodes=2                  # numero di nodi
#SBATCH --ntasks-per-node=2        # numero di processi per nodo
#SBATCH --ntasks-per-socket=1      # numero di processi per socket
#SBATCH --cpus-per-task=64         # numero di core per processo

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

mpirun --bind-to none ./program.exe
```

CRESCO8 – Comandi SLURM: scancel

Per cancellare un job in coda o in esecuzione si usa il comando *scancel*:

- Cancella un job tramite ID: *scancel <job_id>*
- Cancella un job tramite JobName: *scancel --name=<job_name>*
- Cancella tutti i propri job su una partizione: *scancel --partition=<partition>*
- Cancella tutti i propri job su tutte le partizioni: *scancel --me*

CRESCO8 – Comandi SLURM: scontrol

Per informazioni su un job in coda o in esecuzione si usa il comando *scontrol*.

- Mostra informazioni su un job:
scontrol show job <job_id>
- Usando il modulo *slurm_tools*:
showjob <job_id>

```
<matfois@cresco8x001 ~/mpi_tests> scontrol show job 1073442
JobId=1073442 JobName=jobMPI
  UserId=matfois(28140) GroupId=enea(333) MCS_label=N/A
  Priority=21 Nice=0 Account=enea QOS=normal
  JobState=RUNNING Reason=None Dependency=(null)
  Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
  RunTime=00:00:08 TimeLimit=1-00:00:00 TimeMin=N/A
  SubmitTime=2025-09-24T09:59:31 EligibleTime=2025-09-24T09:59:31
  AccrueTime=2025-09-24T09:59:31
  StartTime=2025-09-24T09:59:32 EndTime=2025-09-25T09:59:32 Deadline=N/A
  SuspendTime=None SecsPreSuspend=0 LastSchedEval=2025-09-24T09:59:32 Scheduler=Main
  Partition=cresco8_cpu AllocNode:Sid=cresco8x001:2752837
  ReqNodeList=(null) ExcNodeList=(null)
  NodeList=cresco8x[366-367]
  BatchHost=cresco8x366
  NumNodes=2 NumCPUs=8 NumTasks=8 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
  ReqTRES=cpu=8,mem=1010332M,node=2,billing=8
  AllocTRES=cpu=8,node=2,billing=8
  Socks/Node=* NtasksPerN:B:S:C=4:0:*:* CoreSpec=*
  MinCPUsNode=4 MinMemoryNode=0 MinTmpDiskNode=0
  Features=(null) DelayBoot=00:00:00
  OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
  Command=/afs/enea.it/por/user/matfois/mpi_tests/submit.sh
  WorkDir=/afs/enea.it/por/user/matfois/mpi_tests
  StdErr=/afs/enea.it/por/user/matfois/mpi_tests/1073442.err
  StdIn=/dev/null
  StdOut=/afs/enea.it/por/user/matfois/mpi_tests/1073442.out
  Power=
```

CRESCO8 – Comandi SLURM: scontrol

scontrol può mostrare informazioni anche sullo stato di un nodo di calcolo.

- Mostra informazioni sullo stato di un nodo di calcolo:
scontrol show node <node_name>
- Usando il modulo *slurm_tools*:
shownode <node_name>

```
<matfois@cresco8x001 ~> scontrol show node cresco8x010
NodeName=cresco8x010 Arch=x86_64 CoresPerSocket=64
CPUAlloc=128 CPUEfctv=128 CPUTot=128 CPULoad=128.04
AvailableFeatures=(null)
ActiveFeatures=(null)
Gres=(null)
NodeAddr=cresco8x010 NodeHostName=cresco8x010 Version=23.11.5
OS=Linux 5.14.0-427.13.1.el9_4.x86_64 #1 SMP PREEMPT_DYNAMIC Wed Apr 10 10:29:16 EDT 2024
RealMemory=515166 AllocMem=0 FreeMem=421068 Sockets=2 Boards=1
MemSpecLimit=10000
State=ALLOCATED ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
Partitions=cresco8_cpu
BootTime=2025-09-11T16:45:13 SlurmdStartTime=2025-09-19T10:04:24
LastBusyTime=2025-09-23T19:29:24 ResumeAfterTime=None
CfgTRES=cpu=128,mem=515166M,billing=128
AllocTRES=cpu=128
CapWatts=n/a
CurrentWatts=1023 AveWatts=376
ExtSensorsJoules=n/a ExtSensorsWatts=0 ExtSensorsTemp=n/a
```

CRESCO8 – Comandi SLURM: sacct

Per mostrare informazioni su job eseguiti in passato si usa il comando *sacct*.

- Mostra i job eseguiti di recente: *sacct*

```
<matfois@cresco8x001 /lpor1/scratch_1/usr/matfois> sacct
```

JobID	JobName	Partition	Account	AllocCPUS	State	ExitCode
16369	fluent	cresco7	enea	0	PENDING	0:0
1073442	jobMPI	cresco8_c+	enea	8	COMPLETED	0:0
1073442.bat+	batch		enea	4	COMPLETED	0:0
1073442.0	orted		enea	8	COMPLETED	0:0

CRESCO8 – sacct: argomenti utili

- Mostra i job eseguiti nel mese corrente: `sacct --starttime=2025-09-01`
- Mostra i job eseguiti nel mese corrente in una particolare partizione: `sacct --starttime=2025-09-01 --partition=cresco8_cpu`
- Mostra i job del mese corrente con un certo nome: `sacct --name=<job_name> --starttime=2025-09-01`
- Mostra più informazioni sui job: `sacct --long`
- Customizza il formato dell'output: `sacct --format=JobID,ConsumedEnergy,ExitCode`
- Stampa la lista di campi passabili all'opzione "--format": `sacct --helpformat`
- [Lista completa degli argomenti.](#)

CRESCO8 – Job Array

A volte è necessario sottomettere un gran numero di job seriali, che eseguono la stessa operazione su dati diversi. Con l'argomento “**--array**” di *sbatch* un singolo script determina il lancio di più job, il cui indice viene aggiunto come suffisso al JobID ed è accessibile nello script tramite la variabile **SLURM_ARRAY_TASK_ID**

```
#!/bin/bash
#SBATCH --job-name=job_array      # nome del job
#SBATCH --partition=cresco8_cpu   # partizione
#SBATCH --output=%x-%j.out        # percorso file di output (stdout)
#SBATCH --error=%x-%j.err         # percorso file di errori (stderr)
#SBATCH --nodes=1                 # numero di nodi
#SBATCH --ntasks-per-node=1       # numero di processi per nodo
#SBATCH --array=0-99              # indici del jobarray (da 0 a 99)

./program.exe data.$SLURM_ARRAY_TASK_ID
```

- E' possibile specificare gli indici in modo flessibile, ad esempio l'array “**--array=0-10,55**” genera 12 job, 11 con indici da 0 a 10 e uno con indice 55

CRESCO8 – Job con dipendenze

E' possibile subordinare l'esecuzione di un job alla terminazione (con successo o meno) di altri job, tramite l'argomento "--dependency" di *sbatch*. Alcuni esempi:

- Il job parte dopo che i job specificati sono partiti:
sbatch --dependency=after:<jobid>
- Il job parte dopo che i job specificati sono terminati:
sbatch --dependency=afterany:<jobid>
- Il job parte dopo che i job specificati sono terminati con successo:
sbatch --dependency=afterok:<jobid>
- Il job parte dopo che i job specificati sono terminati senza successo:
sbatch --dependency=afternotok:<jobid>

NB: In caso di fallimento di una dipendenza il job resta in PENDING per sempre!

CRESCO8 – Software con licenza

Alcuni software (ad esempio Ansys) sono disponibili attraverso licenza. Per usarli con SLURM occorre passare ad *sbatch* il seguente argomento:

- `-L <nome_licenza>@<server_licenza>:<numero_licenze>`
- Esempio per Ansys CFD su 2 nodi di CRESCO8 (128 core per nodo):

```
sbatch -L ansys\_cfd@cresco-licn3.portici.enea.it,ansys\_hpc@cresco-licn3.portici.enea.it:256  
--nodes=2 --ntasks-per-node=128 --partition=cresco8_cpu fluent2023r2_slurm.sh 3ddp -i  
myjournalfile
```

- [Maggiori dettagli.](#)

CRESCO8 – Risorse utili

- [Documentazione Ufficiale di SLURM](#)
- [Documentazione CRESCO di SLURM](#)
- [Documentazione del Cluster CRESCO8](#)
- [Guida all'uso di SLURM dell'Università di Princeton](#)
- [Guida all'uso di SLURM dell'Università di Stanford](#)
- [Portale di Assistenza CRESCO](#)

Fine



1101 0110 1100
0101 0010 1101
0001 0110 1110
1101 0010 1101
1111 1010 0000

