Unlocking malleability in HPC applications

Demo Outline

Alberto Cascajo, Javier Fernández, David E. Singh

October 14-16, 2026, Madrid, Spain

38th IEEE/SBC International Symposium on Computer Architecture and High-Performance Computing

ARCOS
COMPUTER ARCHITECTURE, COMMUNICATIONS AND SYSTEMS RESEARCH GROUP

Outline

- ❑ Docker cluster setup
- ❑ Malleability environment setup
- ❑ Samples versions:
  - Original, malleable and start-stop.
- ❑ C-based samples:
  - Hello world, jacobi, wacomm

Docker cluster setup

- Install Docker framework (Windows):
  - Install Windows Subsystem for Linux versión 2 (WSL 2)

    Run in PowerShell as Administrator

    $ wsl --install -d ubuntu
  - Install Docker Desktop for Windows

    Download from https://docs.docker.com/desktop/install/windows/

    Run the installer, After installation, launch Docker Desktop

    Ensure **"Use WSL 2 instead of Hyper-V"** is selected

    Docker Desktop → **Settings** → **Resources** → **WSL Integration**

    Allocate at least **4 CPUs** and **8 GB of memory**

    Docker Desktop → **Settings** → **Resources** → **Advanced**

    enable "Use WSL 2 based engine"

    Docker Desktop → **Settings** → **General**

Docker cluster setup

- Install Docker framework (Mac OS):
  - Install Docker Desktop for Mac OS

    Download from https://docs.docker.com/desktop/install/mac/

    Choose the **Apple Silicon** version if you have an M1/M2/M3/M4 Mac,

    Choose the **Intel** version for Intel Macs

    Drag the `.dmg` into the Applications folder and launch Docker Desktop

    Accept the terms and follow the setup wizard

- Install Docker framework (Linux):
  - Install Docker Desktop for Linux

    Download from https://docs.docker.com/desktop/install/linux/

Docker cluster setup

- Requirements:
  - Docker framework installed
  - Bash console and related tools.

- Download:
  - https://www.dropbox.com/s/ubujyrjra86g0uk/slurm_cluster_pdp_2023-03-01-4.tgz?dl=0

- Cluster Installation:
  - Decompress downloaded file

```
sh> tar zxvf tutorial_cluster_26-04.tgz
```

  - Activate the docker cluster

```
sh> cd tutorial_cluster_26-04
sh> ./launch_cluster.sh build -n 3 -c 3 -p cl
sh> ./launch_cluster.sh start -p cl
```

NOTE: launchs a docker cluster with 3 nodes, each with 3 CPU in slum configuration

Malleability environment setup

- Set monitoring tools
  - Open a console to each cluster node and run monitor (htop):

```
sh> docker exec -it -u mpiuser cl-node1 /bin/bash
sh> htop
```

  - Set htop to look for a specific application (F4 + ”app. name”).

  - Open a console and run slurm view script

```
sh> docker exec -it -u mpiuser cl-node1 /bin/bash
sh> shared/view_results.sh
```

Malleability environment setup

- Set Inteligent controller server
  - Open a console and run slurm view script

```
sh> docker exec -it -u mpiuser cl-node1 /bin/bash
sh> icc_server
```

- Compile samples
  - Open a console and execute make:

```
sh> docker exec -it -u mpiuser cl-node1 /bin/bash
sh> cd shared/tutorial_examples
sh> make
```

- launch samples
  - Open a console and launch sample (ex. helloworld_mpi) from home using absolute path:

```
sh> docker exec -it -u mpiuser cl-node1 /bin/bash
sh> sbatch
    shared/tutorial_examples/helloworld/helloworld_mpi.sbatch
```

Samples versions

- Original version:
  - Suffix: _mpi
  - This version is implemented using plain MPI (no malleability).
  - Executes 4 processes in 2 nodes (no changes).

Samples versions

- Malleable version:
  - Suffix: _dyn
  - This version is implemented using malleability features.
  - The file *_dyn.c is the original mpi version to be modified
  - The file *_dyn_solution.c is the correct version of the malleable solution
  - Starts executing 2 processes in 1 node.
  - Malleability sequence of the solution:

    spawn 2 processes in 1 node.
    spawn 2 processes in 1 node.
    remove 2 processes from 1 node.
    remove 2 processes from 1 node.

Samples versions

- Start-stop version:
  - Suffix: _ss
  - This version is implemented using start-stop features.
  - The file *_ss.c is the original mpi version to be modified
  - The file *_ss_solution.c is the correct version of the malleable solution
  - Starts executing 2 processes in 1 node.
  - Malleability sequence of the solution:

    spawn 2 processes in 1 node.
    spawn 2 processes in 1 node.
    remove 2 processes from 1 node.
    remove 2 processes from 1 node.

C-based samples

- Sample 1: Hello world (helloworld)
  - Initially all processes gather their hostnames.
  - Every second each process print the hostname of all processes.

- Sample 2: jacobi method (jacobi_IO)
  - Determines the solution of a strictly diagonally dominant system of linear equations.
  - Initially the root process broadcast the matrix and the vector solution of the equation.
  - Each iteration a section of the unknown variable vector is approximated by each process.
  - All processes gather the whole unknown variable vector.

- Sample 3: wacomm application (m_wacomm1)

EpiGraph application setup

- Launch application
  - Open a console and execute make:

```
sh> docker exec -it -u mpiuser cl-node1 /bin/bash
sh> cd shared/EpiGraph/EpiGraphFlexMPI
sh> ./TurboLanza.sh
```

- Set monitoring tools
  - Open a console and run slurm view script

```
sh> docker exec -it -u mpiuser cl-node1 /bin/bash
sh> shared/view_results.sh
```

  - Open a console and launch online graph:

```
sh> docker exec -it -u mpiuser cl-node1 /bin/bash
sh> cd shared/EpiGraph/EpiGraphFlexMPI
sh> python3 plot_infected.py --dir cities --global
NOTE: add "--static" to create an .png offline graph
```

Unlocking malleability in HPC applications

Demo Outline

Alberto Cascajo, Javier Fernández, David E. Singh

October 14-16, 2026, Madrid, Spain

38th IEEE/SBC International Symposium on Computer Architecture and High-Performance Computing

ARCOS
COMPUTER ARCHITECTURE, COMMUNICATIONS AND SYSTEMS RESEARCH GROUP