# Integration of the ECP Proxy Apps Suite into the Pavilion Test Harness

Anaira Quezada Christine Kendrick Yolanda Reyes

New Mexico State University

Trinity University

American River College

Mentors: Jennifer Green, Paul Ferrell, Francine Lapid

LA-UR: 20-26116

# Exascale Computing Project (ECP)

- Collaboration between DOE and NNSA
- System that can perform $10^{18}$ (1 exaflop) operations per second

  “50 times faster than the most powerful supercomputers being used today”

- Goal:

  “Develop exascale-ready applications and solutions that address currently intractable problems of strategic importance and national interest.”

  **“Create and deploy an expanded and vertically integrated software stack on DOE HPC pre-exascale and exascale systems.”**

  “Deliver US HPC vendor technology advances and deploy ECP products to DOE HPC pre-exascale and exascale systems.”

# ECP Proxy Application Suite

❑ Currently holds 15 applications

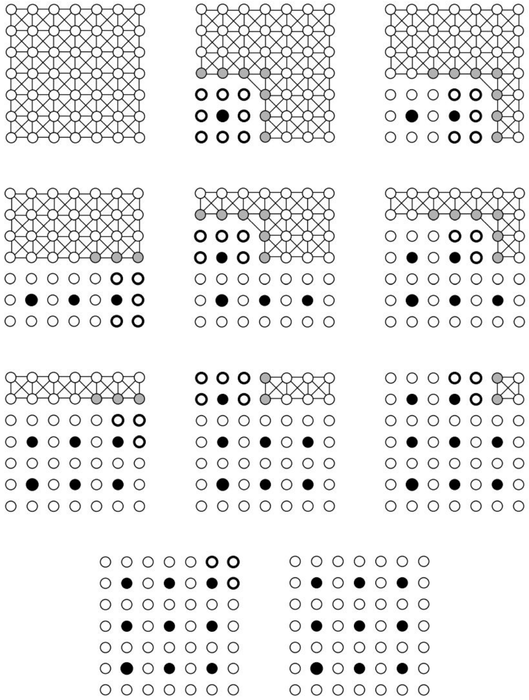
❑ Representative of critical features of future exascale applications

❑ Employ modern parallel programming methods

❑ Model performance-critical computations

❑ Benchmark various HPC subsystem’s performance.

# AMG



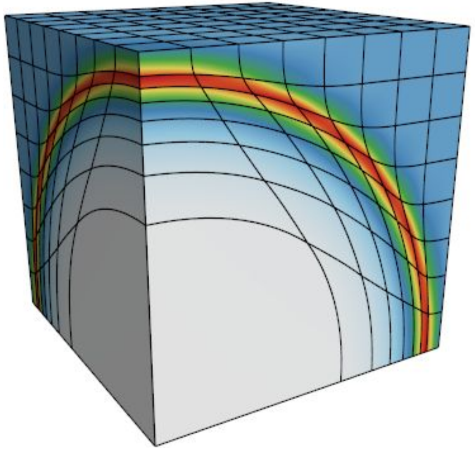
Depicted: Sequence of coloring steps for the nine-point Laplacian on a uniform grid

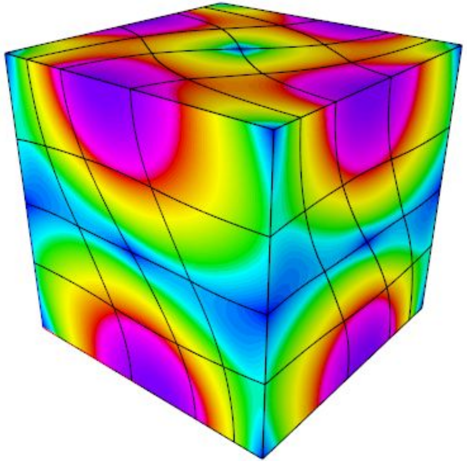
**Requires parallel efficiency**

**Tests memory access bound problems**

https://computing.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods/boom.pdf

# Laghos





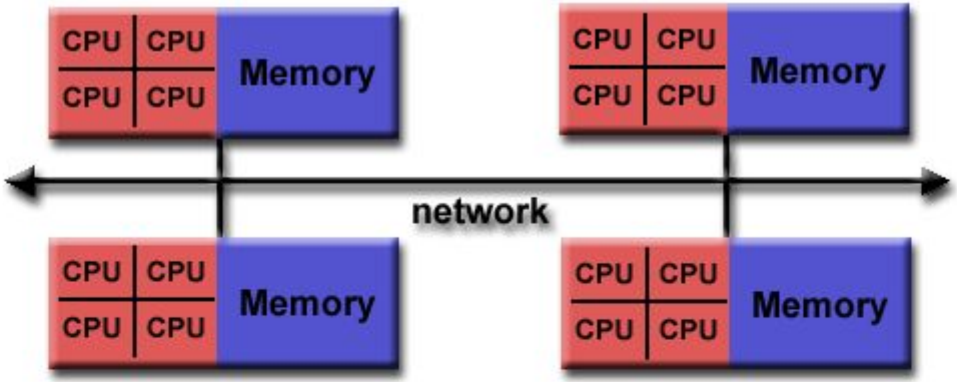
Depicted: 3D rendered meshes

**Models 2D and 3D unstructured meshes**

**Hardware and software flexibility**

https://ceed.exascaleproject.org/miniapps/#laghos-n

# Ember

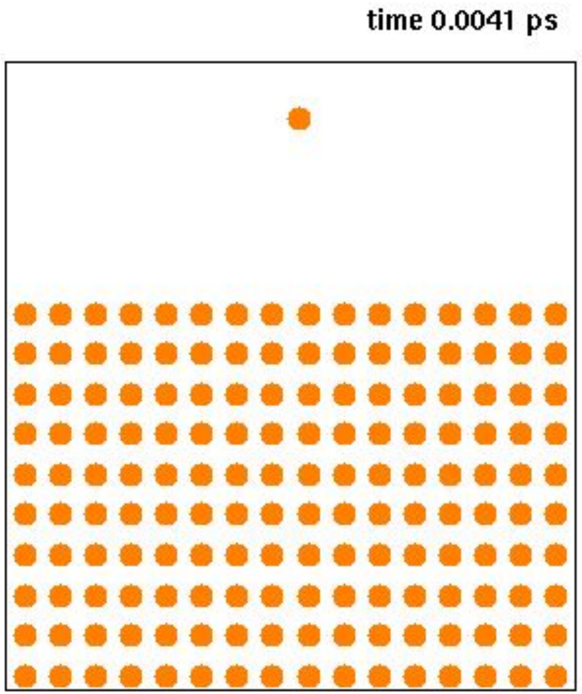


Depicted: Diagram of MPI's method of operating

**Measures interconnect and shared memory performance**

https://computing.llnl.gov/tutorials/mpi/

# ExaMiniMD



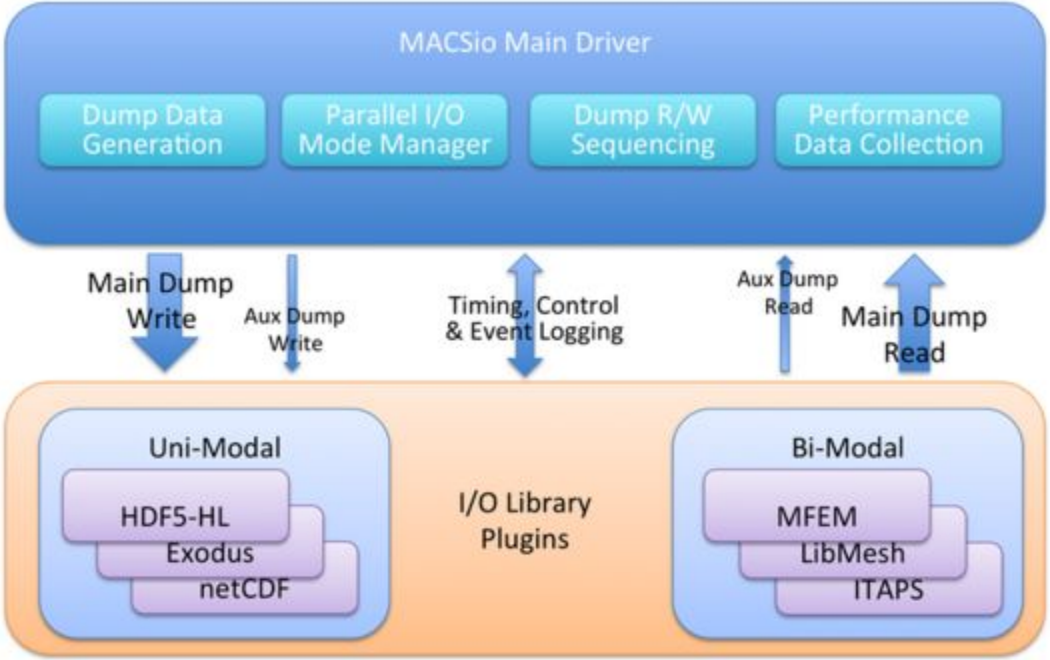
Depicted: A molecular dynamics example

**Kokkos programming model**

**Molecular dynamics simulation**

https://en.wikipedia.org/wiki/Molecular_dynamics

# MACSio

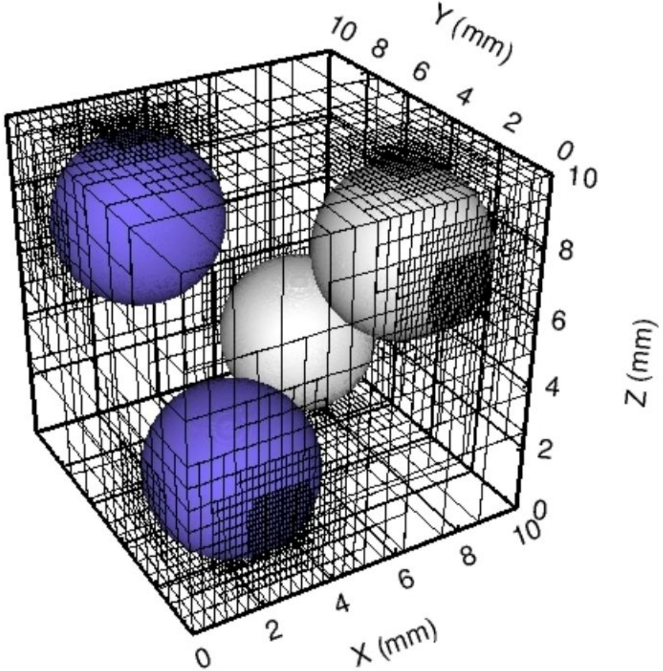


Depicted: MACSio main driver and I/O library plugins

**Measures scalable I/O performance**

https://computing.llnl.gov/projects/co-design/macsio

# MiniAMR

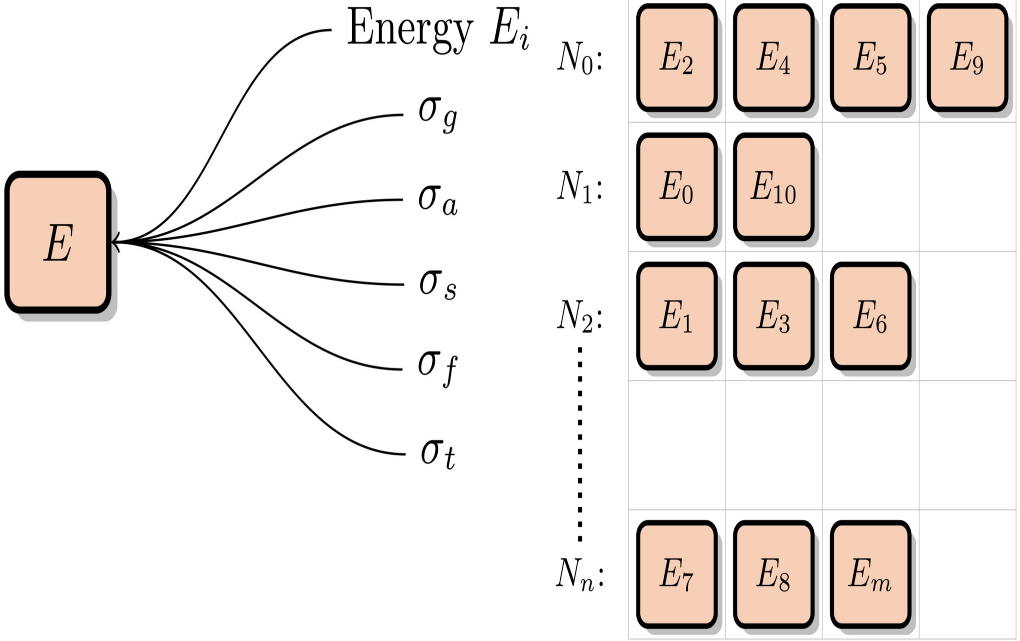


Depicted: Visualization of a four spheres problem MiniAMR ran in a study

**3D stencil calculation on a unit cube**

https://www.osti.gov/servlets/purl/1258271

# XSBench



|  |  |  |  |  |
|---|---|---|---|---|
| $N_0$: | $E_2$ | $E_4$ | $E_5$ | $E_9$ |
| $N_1$: | $E_0$ | $E_{10}$ |  |  |
| $N_2$: | $E_1$ | $E_3$ | $E_6$ |  |
|  |  |  |  |  |
| $N_n$: | $E_7$ | $E_8$ | $E_m$ |  |

Depicted: Nuclide Grid used as a grid search type by XSBench

**Represents main computational kernel of the Monte Carlo neutronics application OpenMC**

https://github.com/ANL-CESAR/XSBench

# LANL’s Pavilion HPC Test Harness

❑ Acceptance Testing!

❑ Python3 based framework

❑ YAML based configuration

❑ Evaluate functionality, usability, and performance of HPC systems

# Our Project

Goal:

❏ Develop an application test suite to run ECP Proxy Applications Suite using Pavilion Test Harness

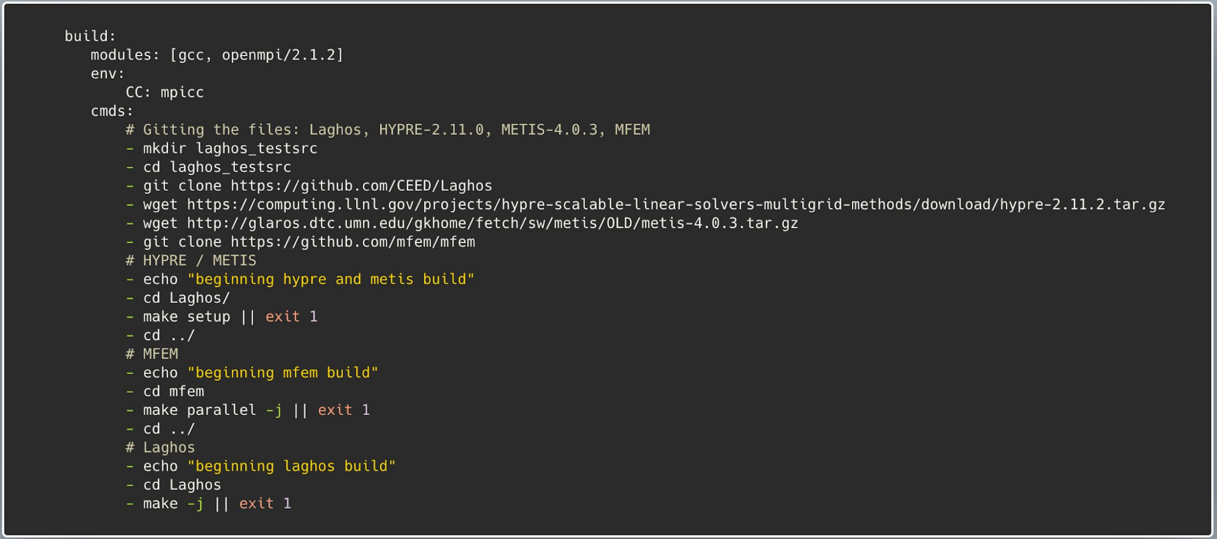
Generalize build commands

Runtime inputs

Capture test results

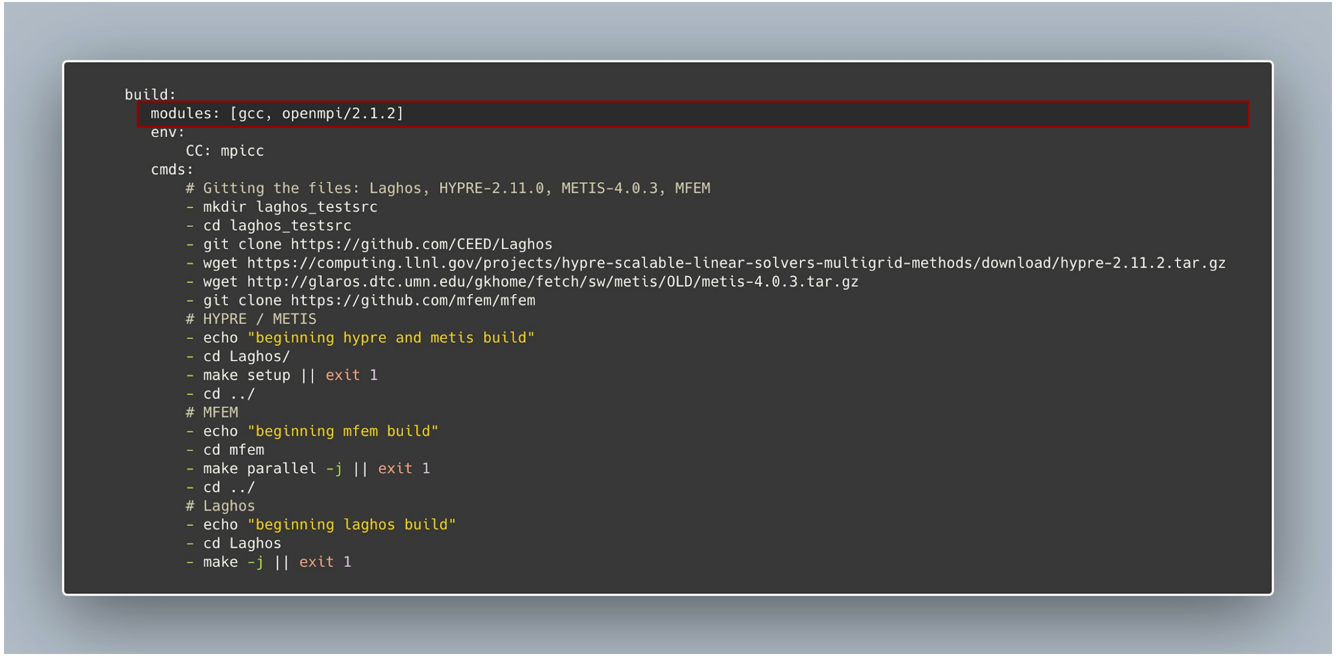
https://www.pngjoy.com/pngs/27/660351_shooting-star-gold-glitter-shooting-stars-png-download.png

# Pavilion Integration: Build



```
build:
   modules: [gcc, openmpi/2.1.2]
   env:
       CC: mpicc
   cmds:
       # Gitting the files: Laghos, HYPRE-2.11.0, METIS-4.0.3, MFEM
       - mkdir laghos_testsrc
       - cd laghos_testsrc
       - git clone https://github.com/CEED/Laghos
       - wget https://computing.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods/download/hypre-2.11.2.tar.gz
       - wget http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/OLD/metis-4.0.3.tar.gz
       - git clone https://github.com/mfem/mfem
       # HYPRE / METIS
       - echo "beginning hypre and metis build"
       - cd Laghos/
       - make setup || exit 1
       - cd ../
       # MFEM
       - echo "beginning mfem build"
       - cd mfem
       - make parallel -j || exit 1
       - cd ../
       # Laghos
       - echo "beginning laghos build"
       - cd Laghos
       - make -j || exit 1
```

# Pavilion Integration: Build

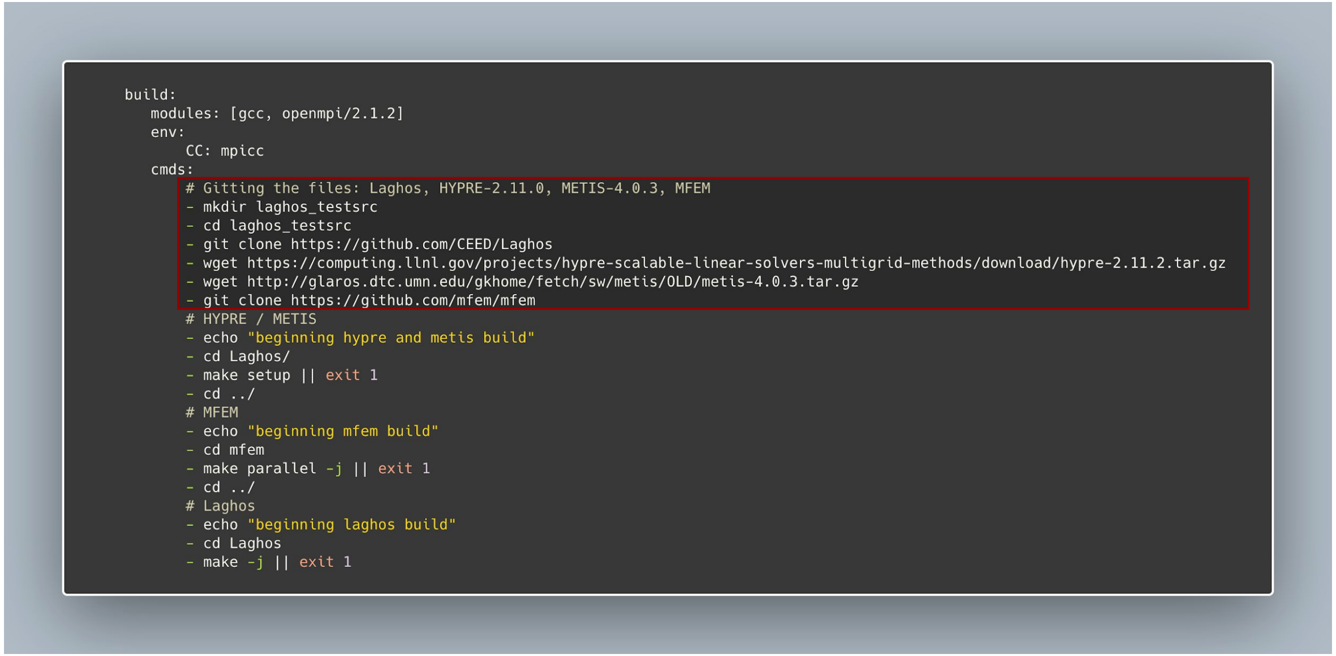


```
build:
   modules: [gcc, openmpi/2.1.2]
   env:
       CC: mpicc
   cmds:
       # Gitting the files: Laghos, HYPRE-2.11.0, METIS-4.0.3, MFEM
       - mkdir laghos_testsrc
       - cd laghos_testsrc
       - git clone https://github.com/CEED/Laghos
       - wget https://computing.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods/download/hypre-2.11.2.tar.gz
       - wget http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/OLD/metis-4.0.3.tar.gz
       - git clone https://github.com/mfem/mfem
       # HYPRE / METIS
       - echo "beginning hypre and metis build"
       - cd Laghos/
       - make setup || exit 1
       - cd ../
       # MFEM
       - echo "beginning mfem build"
       - cd mfem
       - make parallel -j || exit 1
       - cd ../
       # Laghos
       - echo "beginning laghos build"
       - cd Laghos
       - make -j || exit 1
```

Load build modules:

- ❏ gcc
- ❏ openmpi

# Pavilion Integration: Build



```
build:
   modules: [gcc, openmpi/2.1.2]
   env:
       CC: mpicc
   cmds:
       # Gitting the files: Laghos, HYPRE-2.11.0, METIS-4.0.3, MFEM
       - mkdir laghos_testsrc
       - cd laghos_testsrc
       - git clone https://github.com/CEED/Laghos
       - wget https://computing.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods/download/hypre-2.11.2.tar.gz
       - wget http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/OLD/metis-4.0.3.tar.gz
       - git clone https://github.com/mfem/mfem
       # HYPRE / METIS
       - echo "beginning hypre and metis build"
       - cd Laghos/
       - make setup || exit 1
       - cd ../
       # MFEM
       - echo "beginning mfem build"
       - cd mfem
       - make parallel -j || exit 1
       - cd ../
       # Laghos
       - echo "beginning laghos build"
       - cd Laghos
       - make -j || exit 1
```

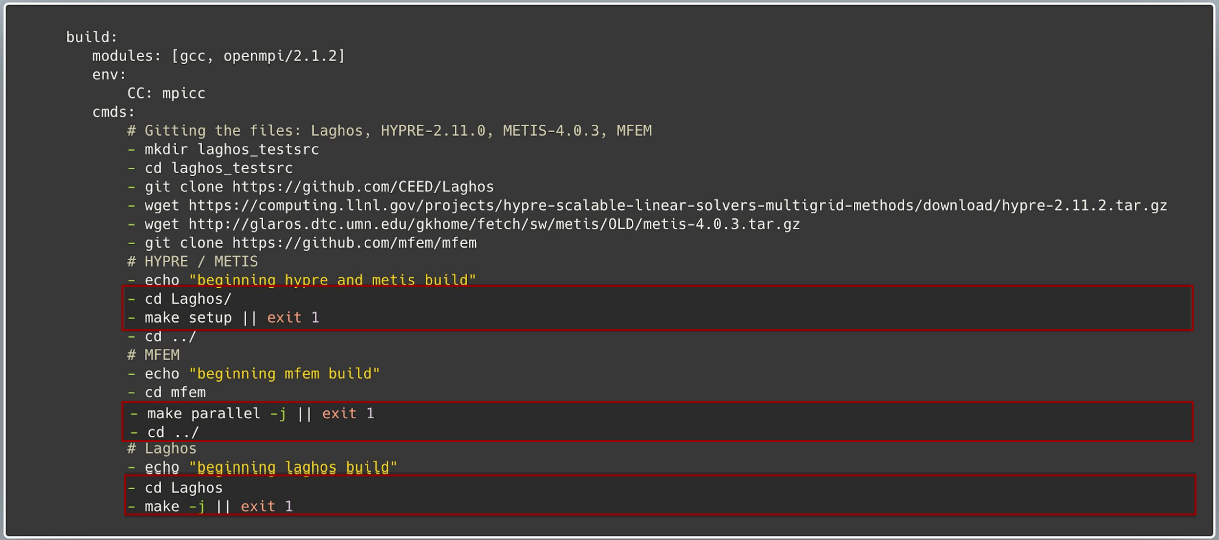
Load build modules:

- □ gcc
- □ openmpi

Clone repositories:

- □ Laghos
- □ HYPRE
- □ METIS
- □ MFEM

# Pavilion Integration: Build



```
build:
   modules: [gcc, openmpi/2.1.2]
   env:
       CC: mpicc
   cmds:
       # Gitting the files: Laghos, HYPRE-2.11.0, METIS-4.0.3, MFEM
       - mkdir laghos_testsrc
       - cd laghos_testsrc
       - git clone https://github.com/CEED/Laghos
       - wget https://computing.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods/download/hypre-2.11.2.tar.gz
       - wget http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/OLD/metis-4.0.3.tar.gz
       - git clone https://github.com/mfem/mfem
       # HYPRE / METIS
       - echo "beginning hypre and metis build"
       - cd Laghos/
       - make setup || exit 1
       - cd ../
       # MFEM
       - echo "beginning mfem build"
       - cd mfem
       - make parallel -j || exit 1
       - cd ../
       # Laghos
       - echo "beginning laghos build"
       - cd Laghos
       - make -j || exit 1
```

Load build modules:

❑ gcc
❑ openmpi

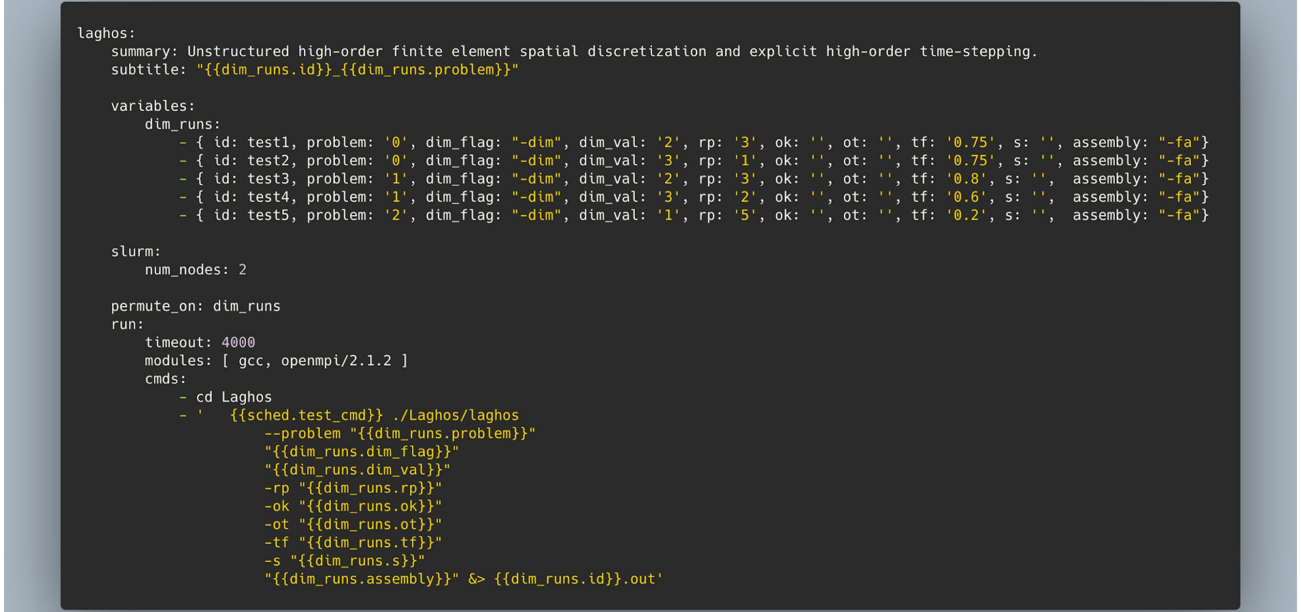
Clone repositories:

❑ Laghos
❑ HYPRE
❑ METIS
❑ MFEM

Run “make” commands

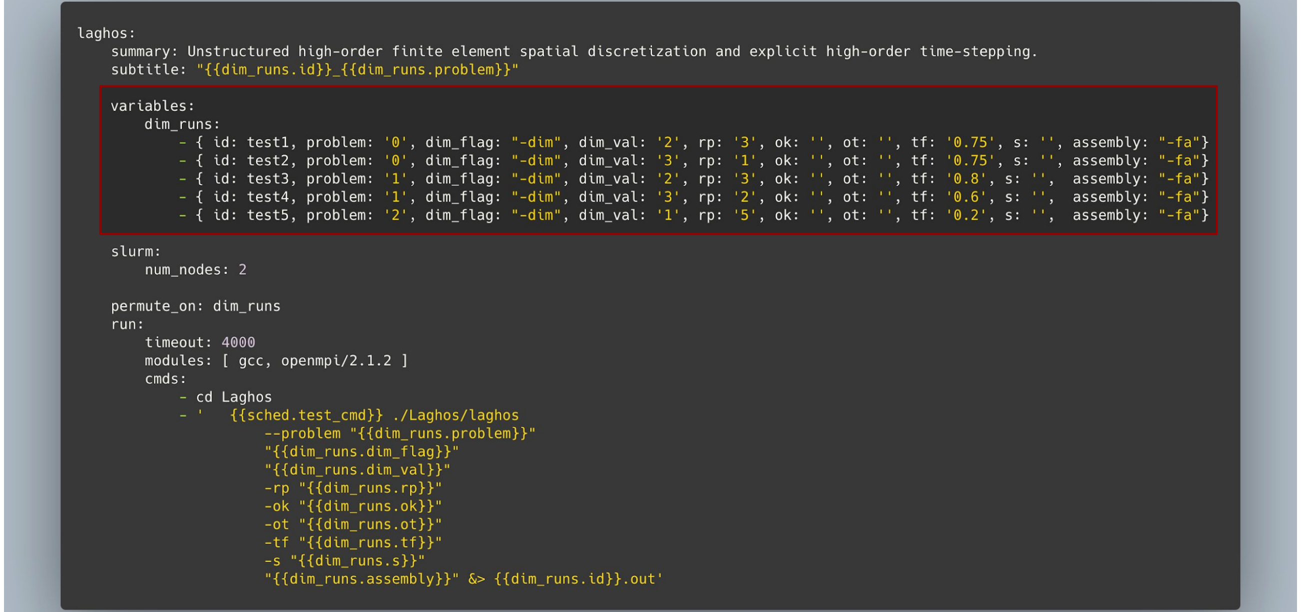
❑ HYPRE/METIS
`make setup`
❑ MFEM
`make parallel`
❑ Laghos
`make -j`

# Pavilion Integration: Run



```
laghos:
    summary: Unstructured high-order finite element spatial discretization and explicit high-order time-stepping.
    subtitle: "{{dim_runs.id}}_{{dim_runs.problem}}"

    variables:
        dim_runs:
            - { id: test1, problem: '0', dim_flag: "-dim", dim_val: '2', rp: '3', ok: '', ot: '', tf: '0.75', s: '', assembly: "-fa"}
            - { id: test2, problem: '0', dim_flag: "-dim", dim_val: '3', rp: '1', ok: '', ot: '', tf: '0.75', s: '', assembly: "-fa"}
            - { id: test3, problem: '1', dim_flag: "-dim", dim_val: '2', rp: '3', ok: '', ot: '', tf: '0.8', s: '',  assembly: "-fa"}
            - { id: test4, problem: '1', dim_flag: "-dim", dim_val: '3', rp: '2', ok: '', ot: '', tf: '0.6', s: '',  assembly: "-fa"}
            - { id: test5, problem: '2', dim_flag: "-dim", dim_val: '1', rp: '5', ok: '', ot: '', tf: '0.2', s: '',  assembly: "-fa"}

    slurm:
        num_nodes: 2

    permute_on: dim_runs
    run:
        timeout: 4000
        modules: [ gcc, openmpi/2.1.2 ]
        cmds:
            - cd Laghos
            - '   {{sched.test_cmd}} ./Laghos/laghos
                      --problem "{{dim_runs.problem}}"
                      "{{dim_runs.dim_flag}}"
                      "{{dim_runs.dim_val}}"
                      -rp "{{dim_runs.rp}}"
                      -ok "{{dim_runs.ok}}"
                      -ot "{{dim_runs.ot}}"
                      -tf "{{dim_runs.tf}}"
                      -s "{{dim_runs.s}}"
                      "{{dim_runs.assembly}}" &> {{dim_runs.id}}.out'
```

# Pavilion Integration: Run

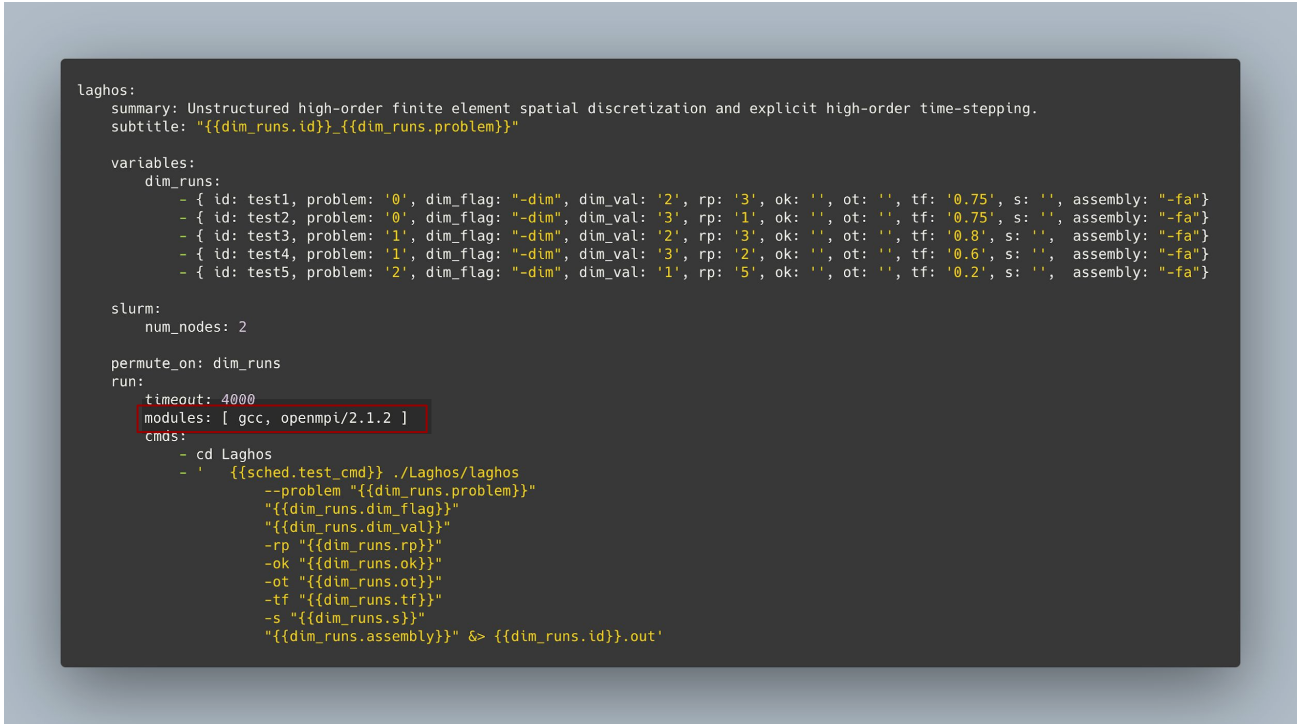


```
laghos:
    summary: Unstructured high-order finite element spatial discretization and explicit high-order time-stepping.
    subtitle: "{{dim_runs.id}}_{{dim_runs.problem}}"

    variables:
        dim_runs:
            - { id: test1, problem: '0', dim_flag: "-dim", dim_val: '2', rp: '3', ok: '', ot: '', tf: '0.75', s: '', assembly: "-fa"}
            - { id: test2, problem: '0', dim_flag: "-dim", dim_val: '3', rp: '1', ok: '', ot: '', tf: '0.75', s: '', assembly: "-fa"}
            - { id: test3, problem: '1', dim_flag: "-dim", dim_val: '2', rp: '3', ok: '', ot: '', tf: '0.8', s: '',  assembly: "-fa"}
            - { id: test4, problem: '1', dim_flag: "-dim", dim_val: '3', rp: '2', ok: '', ot: '', tf: '0.6', s: '',  assembly: "-fa"}
            - { id: test5, problem: '2', dim_flag: "-dim", dim_val: '1', rp: '5', ok: '', ot: '', tf: '0.2', s: '',  assembly: "-fa"}

    slurm:
        num_nodes: 2

    permute_on: dim_runs
    run:
        timeout: 4000
        modules: [ gcc, openmpi/2.1.2 ]
        cmds:
            - cd Laghos
            - '   {{sched.test_cmd}} ./Laghos/laghos
                      --problem "{{dim_runs.problem}}"
                      "{{dim_runs.dim_flag}}"
                      "{{dim_runs.dim_val}}"
                      -rp "{{dim_runs.rp}}"
                      -ok "{{dim_runs.ok}}"
                      -ot "{{dim_runs.ot}}"
                      -tf "{{dim_runs.tf}}"
                      -s "{{dim_runs.s}}"
                      "{{dim_runs.assembly}}" &> {{dim_runs.id}}.out'
```

variables:

- ❏ Permute each id
- ❏ Variables used to hold values for each test.

# Pavilion Integration: Run



```
laghos:
    summary: Unstructured high-order finite element spatial discretization and explicit high-order time-stepping.
    subtitle: "{{dim_runs.id}}_{{dim_runs.problem}}"

    variables:
        dim_runs:
            - { id: test1, problem: '0', dim_flag: "-dim", dim_val: '2', rp: '3', ok: '', ot: '', tf: '0.75', s: '', assembly: "-fa"}
            - { id: test2, problem: '0', dim_flag: "-dim", dim_val: '3', rp: '1', ok: '', ot: '', tf: '0.75', s: '', assembly: "-fa"}
            - { id: test3, problem: '1', dim_flag: "-dim", dim_val: '2', rp: '3', ok: '', ot: '', tf: '0.8', s: '',  assembly: "-fa"}
            - { id: test4, problem: '1', dim_flag: "-dim", dim_val: '3', rp: '2', ok: '', ot: '', tf: '0.6', s: '',  assembly: "-fa"}
            - { id: test5, problem: '2', dim_flag: "-dim", dim_val: '1', rp: '5', ok: '', ot: '', tf: '0.2', s: '',  assembly: "-fa"}

    slurm:
        num_nodes: 2

    permute_on: dim_runs
    run:
        timeout: 4000
        modules: [ gcc, openmpi/2.1.2 ]
        cmds:
            - cd Laghos
            - '   {{sched.test_cmd}} ./Laghos/laghos
                    --problem "{{dim_runs.problem}}"
                    "{{dim_runs.dim_flag}}"
                    "{{dim_runs.dim_val}}"
                    -rp "{{dim_runs.rp}}"
                    -ok "{{dim_runs.ok}}"
                    -ot "{{dim_runs.ot}}"
                    -tf "{{dim_runs.tf}}"
                    -s "{{dim_runs.s}}"
                    "{{dim_runs.assembly}}" &> {{dim_runs.id}}.out'
```

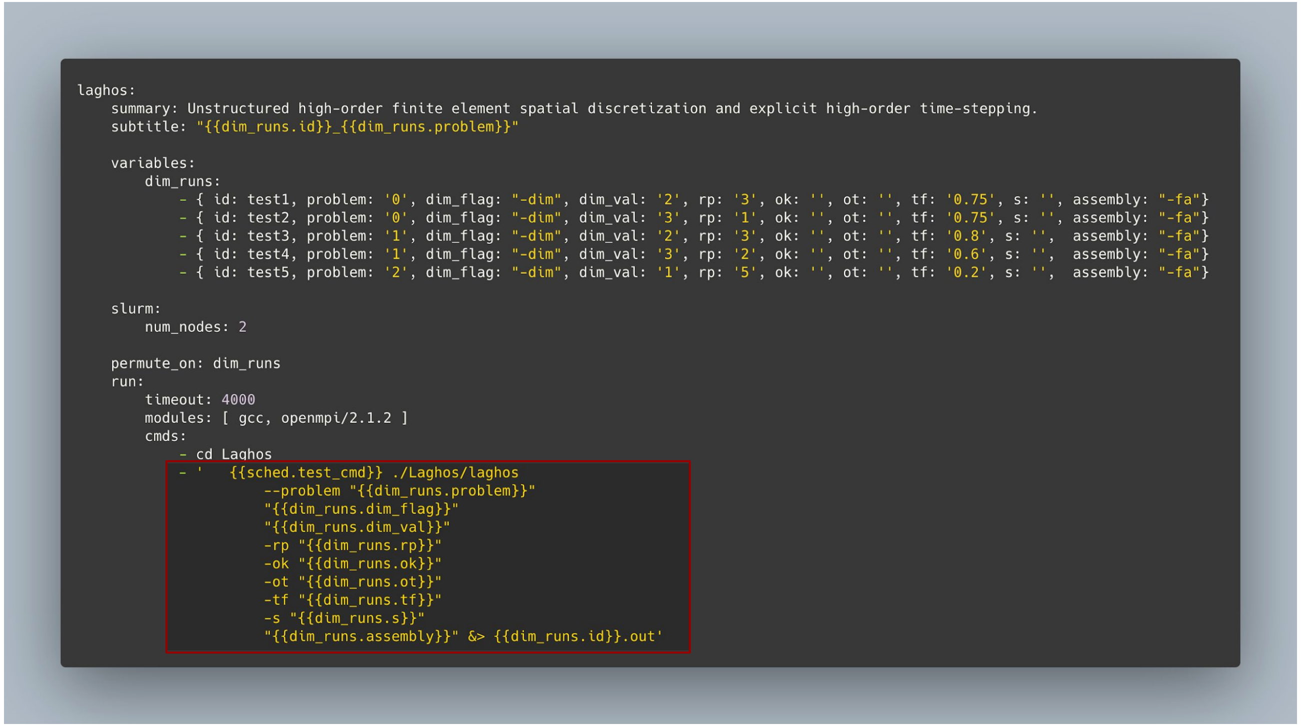
variables:

- ❑ Permute each id
- ❑ Variables used to hold values for each test

Load Modules

- ❑ gcc
- ❑ openmpi

# Pavilion Integration: Run



```
laghos:
    summary: Unstructured high-order finite element spatial discretization and explicit high-order time-stepping.
    subtitle: "{{dim_runs.id}}_{{dim_runs.problem}}"

    variables:
        dim_runs:
            - { id: test1, problem: '0', dim_flag: "-dim", dim_val: '2', rp: '3', ok: '', ot: '', tf: '0.75', s: '', assembly: "-fa"}
            - { id: test2, problem: '0', dim_flag: "-dim", dim_val: '3', rp: '1', ok: '', ot: '', tf: '0.75', s: '', assembly: "-fa"}
            - { id: test3, problem: '1', dim_flag: "-dim", dim_val: '2', rp: '3', ok: '', ot: '', tf: '0.8', s: '',  assembly: "-fa"}
            - { id: test4, problem: '1', dim_flag: "-dim", dim_val: '3', rp: '2', ok: '', ot: '', tf: '0.6', s: '',  assembly: "-fa"}
            - { id: test5, problem: '2', dim_flag: "-dim", dim_val: '1', rp: '5', ok: '', ot: '', tf: '0.2', s: '',  assembly: "-fa"}

    slurm:
        num_nodes: 2

    permute_on: dim_runs
    run:
        timeout: 4000
        modules: [ gcc, openmpi/2.1.2 ]
        cmds:
            - cd Laghos
            - '   {{sched.test_cmd}} ./Laghos/laghos
                      --problem "{{dim_runs.problem}}"
                      "{{dim_runs.dim_flag}}"
                      "{{dim_runs.dim_val}}"
                      -rp "{{dim_runs.rp}}"
                      -ok "{{dim_runs.ok}}"
                      -ot "{{dim_runs.ot}}"
                      -tf "{{dim_runs.tf}}"
                      -s "{{dim_runs.s}}"
                      "{{dim_runs.assembly}}" &> {{dim_runs.id}}.out'
```

variables:

- ❑ Permute each id
- ❑ Variables used to hold values for each test

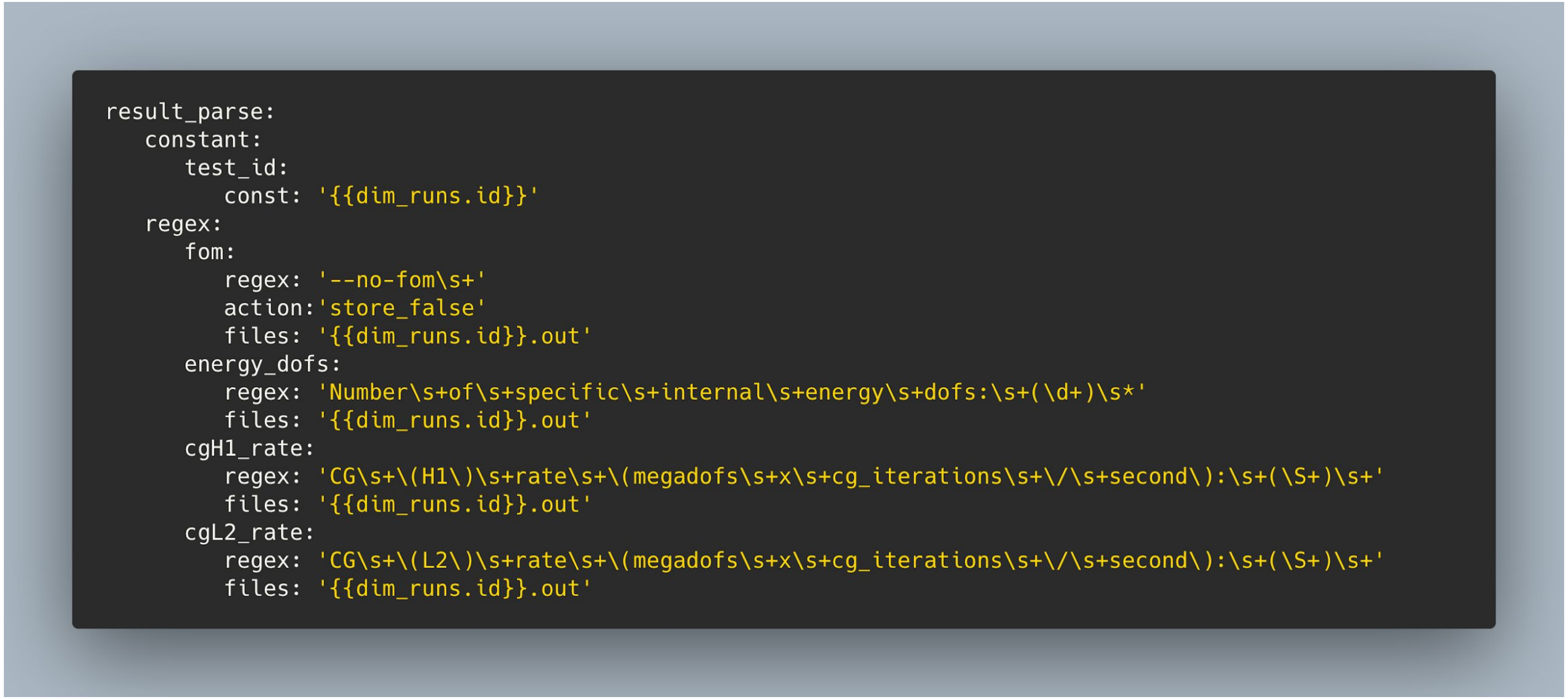
Load Modules

- ❑ gcc
- ❑ openmpi

Scheduling run:

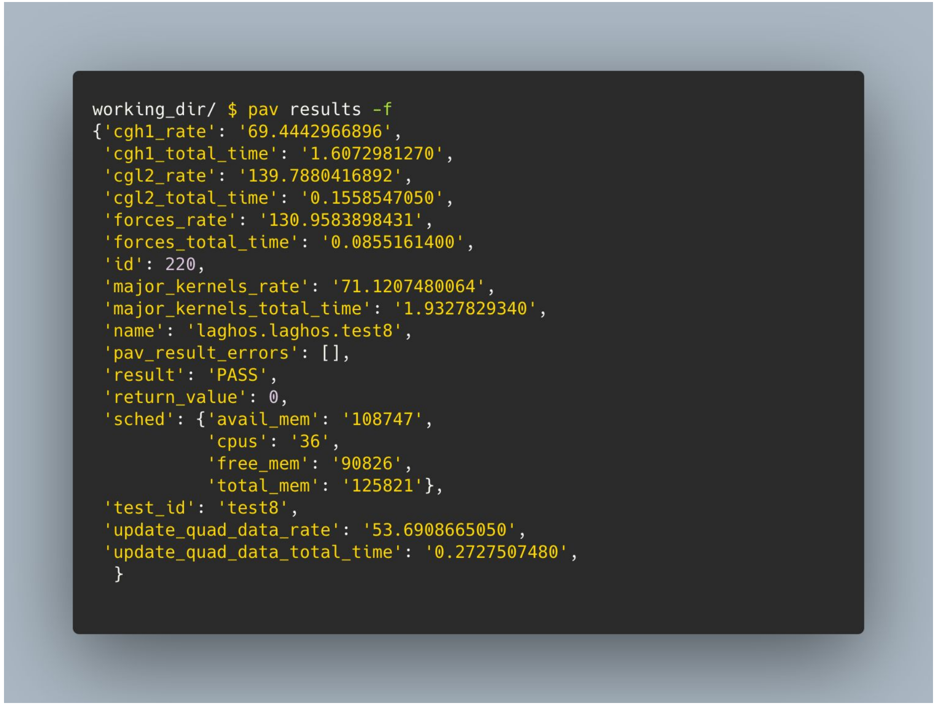
- ❑ 8 tasks per problem
- ❑ Saves results in {dim_runs.id}.out

# Pavilion Integration: Results



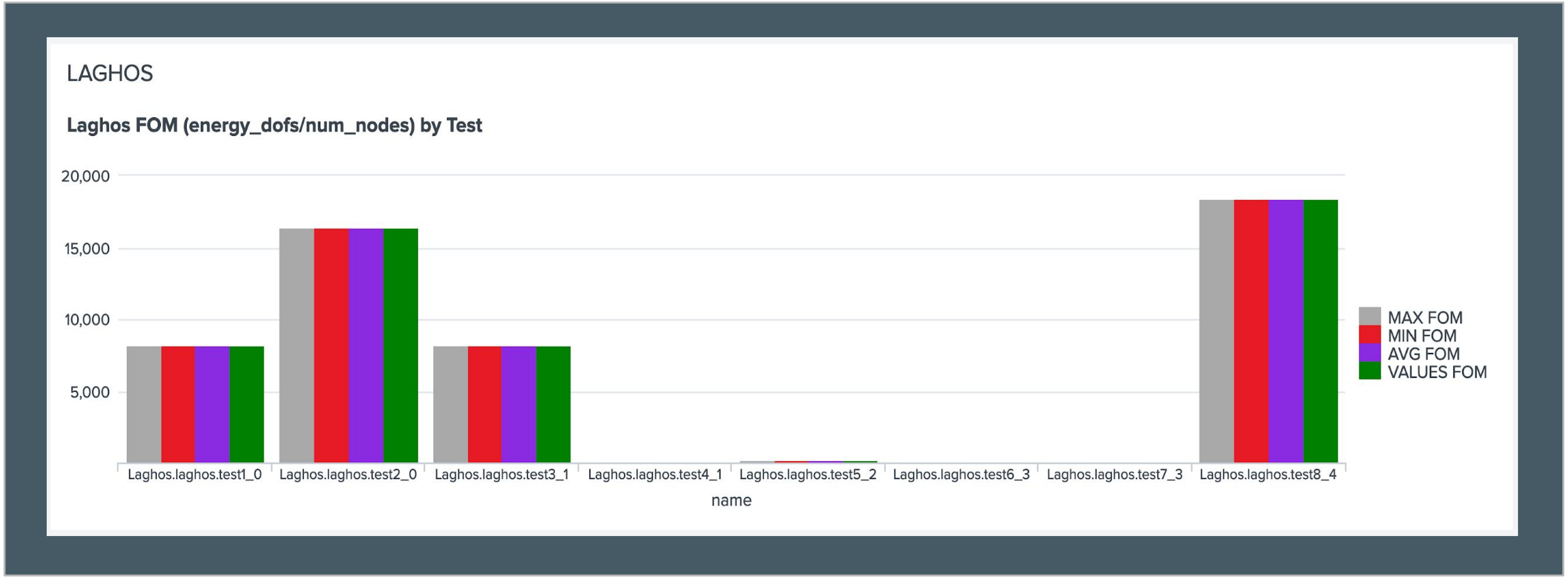
```
result_parse:
   constant:
      test_id:
         const: '{{dim_runs.id}}'
   regex:
      fom:
         regex: '--no-fom\s+'
         action:'store_false'
         files: '{{dim_runs.id}}.out'
      energy_dofs:
         regex: 'Number\s+of\s+specific\s+internal\s+energy\s+dofs:\s+(\d+)\s*'
         files: '{{dim_runs.id}}.out'
      cgH1_rate:
         regex: 'CG\s+\(H1\)\s+rate\s+\(megadofs\s+x\s+cg_iterations\s+\/\s+second\):\s+(\S+)\s+'
         files: '{{dim_runs.id}}.out'
      cgL2_rate:
         regex: 'CG\s+\(L2\)\s+rate\s+\(megadofs\s+x\s+cg_iterations\s+\/\s+second\):\s+(\S+)\s+'
         files: '{{dim_runs.id}}.out'
```

# Pavilion Integration: Results

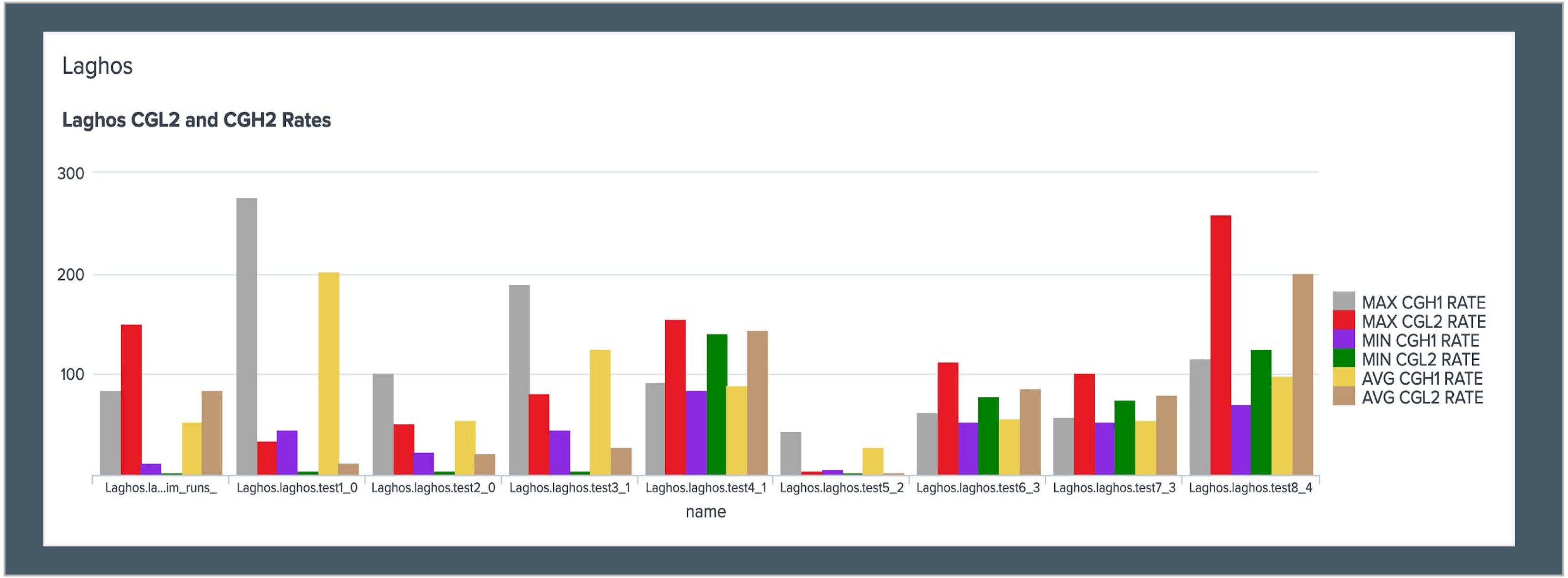


```
working_dir/ $ pav results -f
{'cgh1_rate': '69.4442966896',
 'cgh1_total_time': '1.6072981270',
 'cgl2_rate': '139.7880416892',
 'cgl2_total_time': '0.1558547050',
 'forces_rate': '130.9583898431',
 'forces_total_time': '0.0855161400',
 'id': 220,
 'major_kernels_rate': '71.1207480064',
 'major_kernels_total_time': '1.9327829340',
 'name': 'laghos.laghos.test8',
 'pav_result_errors': [],
 'result': 'PASS',
 'return_value': 0,
 'sched': {'avail_mem': '108747',
           'cpus': '36',
           'free_mem': '90826',
           'total_mem': '125821'},
 'test_id': 'test8',
 'update_quad_data_rate': '53.6908665050',
 'update_quad_data_total_time': '0.2727507480',
  }
```

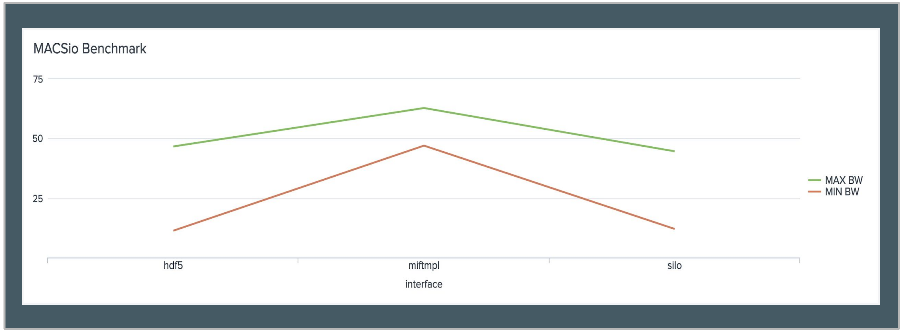
# Pavilion Integration: Results

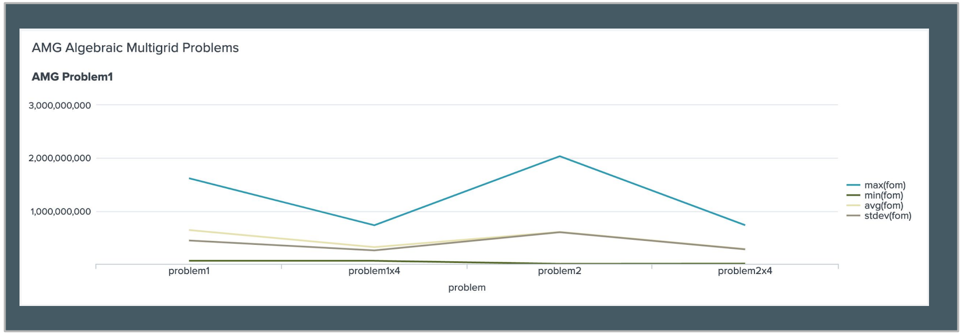


# Pavilion Integration: Results

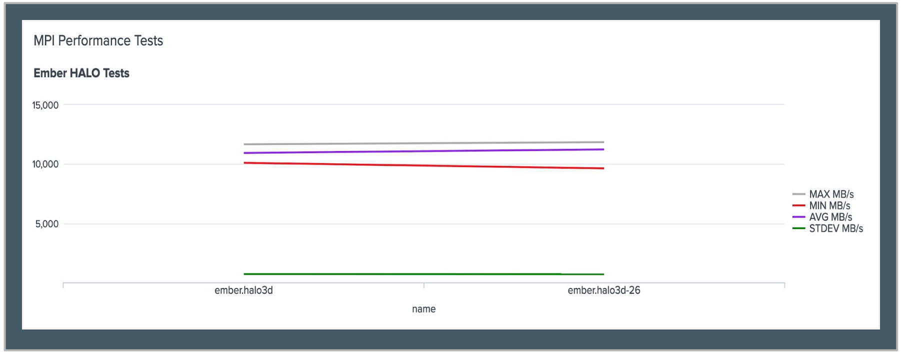


# Pavilion Integration: Results

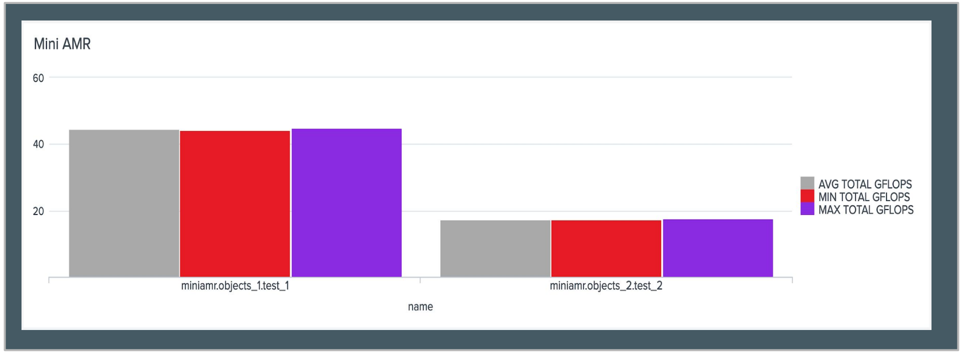

MACSio Benchmark


AMG Algebraic Multigrid Problems

AMG Problem1


MPI Performance Tests

Ember HALO Tests


Mini AMR

# Pavilion Integration: Summary

ECP

EXASCALE COMPUTING PROJECT

Exascale Computing Project

PAVILION

HPC TEST HARNESS

Pavilion

Our Repository

Los Alamos
NATIONAL LABORATORY
EST. 1943

**Anaira Quezada**

quezada.anaira@gmail.com

**Christine Kendrick**

christine.elise.kendrick@gmail.com

**Yolanda Reyes**

w1606799@apps.losrios.edu