ChaNGa
Performance Benchmarking and Profiling

July 2019
• The following research was performed under the HPC Advisory Council activities
  – Compute resource - HPC Advisory Council Cluster Center

• The following was done to provide best practices
  – ChaNGa performance overview over Intel Skylake based platforms
  – Tuning ChaNGa over Charm++ machine layers
  – Understanding ChaNGa profile patterns

• References
  – ChaNGa page: http://charm.cs.uiuc.edu/research/cosmology
  – HowTo Build and Run ChaNGa over Charm++ using UCX/MPI Machine layer
  – ChaNGa on Github
  – Charm++ on Github
  – ChaNGa Benchmarks
Cosmological simulation framework "ChaNGa" is a collaborative project with Prof. Thomas Quinn (University of Washington: N-Body Shop) supported by the NSF.

ChaNGa (Charm N-body GrAvity solver) is a code to perform collisionless N-body simulations.

ChaNGa can perform cosmological simulations with periodic boundary conditions in comoving coordinates or simulations of isolated stellar systems.

ChaNGa can include hydrodynamics using the Smooth Particle Hydrodynamics (SPH) technique. It uses a Barnes-Hut tree to calculate gravity, with hexadecapole expansion of nodes and Ewald summation for periodic forces.

Timestepping is done with a leapfrog integrator with individual timesteps for each particle.

ChaNGa’s uses dynamic load balancing scheme of the Charm++ runtime system in order to obtain good performance on massively parallel systems.
Cluster Configuration

- **Helios Cluster**
  - Supermicro SYS-6029U-TR4 / Foxconn Groot 1A42USF00-600-G 32-node cluster
  - Dual Socket Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz
  - Mellanox ConnectX-6 HDR100 100Gb/s InfiniBand adapters
  - Mellanox Quantum 200Gb/s HDR InfiniBand switch
  - Memory: 192GB DDR4 2677MHz RDIMMs per node
  - 1TB 7.2K RPM SSD 2.5" hard drive per node

- **Software**
  - OS: RHEL 7.6, MLNX_OFED 4.6
  - MPI: HPC-X 2.4
  - Intel Compiler 2018.5
  - Charm++ (git master July 2019)
  - ChaNGa 3.3-409
Testing Details

• **ChaNGa over Charm++ Machine Layers (ML)**
  – UCX ([http://www.openucx.org](http://www.openucx.org))
  – Intel MPI

• **Input file**
  – dwf (dwf1.2048) is a 5 million particle zoom-in simulation
    • It is cosmological, but the particle sampling focuses on a single halo of roughly 1e11 solar masses
  – dwf (dwf1.6144) is a 50 million particle zoom-in simulation

• **ChaNGa Benchmarking**
  – Out of the Box performance (no tuning)
  – AVX Tuning
  – Particles per Bucket Tuning

• **Output**
  – Sum of 10 Big Steps (elapsed)
ChaNGa Out of the Box Performance Results

- UCX Machine provides 49% higher performance at 16 nodes
- UCX Machine provides 161% higher performance at 32 nodes
- Performance reduction demonstrated with Intel MPI beyond of 16 nodes
ChaNGa with AVX Tuning Performance Results

- UCX Machine provides 47% higher performance at 16 nodes
- UCX Machine provides 175% higher performance at 32 nodes
- Performance reduction demonstrated with Intel MPI beyond of 16 nodes

**ChaNGa**
(dwf1.2048.00384, AVX)

Jobs/day

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>Jobs/day</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Intel MPI: 0, UCX: 0</td>
</tr>
<tr>
<td>4</td>
<td>Intel MPI: 1, UCX: 1</td>
</tr>
<tr>
<td>8</td>
<td>Intel MPI: 2, UCX: 3</td>
</tr>
<tr>
<td>16</td>
<td>Intel MPI: 4, UCX: 9</td>
</tr>
<tr>
<td>32</td>
<td>Intel MPI: 8, UCX: 17</td>
</tr>
</tbody>
</table>

Higher is better
ChaNga with AVX, Particles per Bucket Tuning

- UCX Machine provides 49% higher performance at 16 nodes
- UCX Machine provides 171% higher performance at 32 nodes
- Performance reduction demonstrated with Intel MPI beyond of 16 nodes
ChaNGa (using UCX Machine layer) with Tuning

Higher is better

ChaNGa
(dwf1.2048.00384)

jobs/day

Number of Nodes

UCX (out of the box)  UCX (avx)  UCX (avx, b=24)
ChaNGa with Larger Input File (dwf1 50M)

ChaNGa
(dwf1.6144.01472, AVX, -b=24)

Higher is better

Number of Nodes
- 4
- 8
- 16

Jobs/day
- Intel MPI
- UCX

104% 200%
ChaNGa Application Profile with “dwf1” (16 nodes, 640 cores)
ChaNGa Summary

- UCX Machine Layer demonstrates higher performance versus Intel MPI
  - UCX Machine provides up to 49% higher performance at 16 nodes
  - UCX Machine provides up to 175% higher performance at 32 nodes

- Performance reduction demonstrated with Intel MPI
  - beyond of 16 nodes on dwf1.2048 (small input)
  - beyond of 8 nodes on dwf1.6144 (large input)

- AVX and Particles per Bucket Tuning increase performance by 33%
Run Command and Build Example

- **Charmrun commands:**
  - HPC-X/UCX ML (HDR100):
    
    ```
    ./charmrun +p80 -x UCX_NET_DEVICES=mlx5_0:1 ./ChaNGa ./dwf1.2048.param | grep Big | awk '{ SUM += $5} END { print "BigStepTime=" SUM }'
    ```

- **Intel MPI**
  - ```
  ./charmrun +p80 -genv I_MPI_DAPL_PROVIDER=ofa-v2-mlx5_0-1u -genv I_MPI_FABRICS shm:dapl
  ./ChaNGa ./dwf1.2048.param | grep Big | awk '{ SUM += $5} END { print "BigStepTime=" SUM }'
  ```

- **Build walkthrough example**
  - [HowTo Build and Run ChaNGa over Charm++ using UCX/MPI Machine layer](#)
Thank You