Acceptance Testing the Chicoma HPE Cray EX Supercomputer

Presented by: Jennifer K. Green, Scientist

Programming & Runtime Environments Team
High Performance Computing Environments
Los Alamos National Laboratory

May 3, 2021
Presented to the CUG 2021 Virtual Conference

LA-UR-21-24209
Background

• Los Alamos National Laboratory (LANL) remains at the forefront of addressing global crises using state-of-the-art computational resources to accelerate scientific innovation and discovery
• LANL is supplying high-performance computing (HPC) resources to contribute to the recovery from the impacts of SARS-CoV-2 (Coronavirus Pandemic)
• Chicoma is an HPE Cray EX Supercomputer recently installed at LANL to specifically serve as a platform to supply molecular dynamics simulation computing cycles for epidemiological modeling, bioinformatics, and chromosome/RNA simulations as part of the 2020 Coronavirus Aid, Relief, and Economic Security (CARES) Act

Overview of Presentation

- Chicoma HPE Cray EX System Description
- Acceptance Testing Approach
- Testing Tools Description
- Test Suite Contents
- Results
- Conclusions
- Future Work
The Chicoma Supercomputer
System Details

- Chicoma is an early deployment of HPE Cray EX
- Has a large-scale system architecture
- Brand new Shasta system software stack
- Features direct-to-chip liquid cooling
- HPE Slingshot interconnect
- AMD EPYC 7H12 processor
- In total
  - more than 73,000 cores
  - 300 TB of system memory
Testing Motivation and Context

● Preparing an Acceptance Testing Plan involves:
  ○ Develop an understanding of the intended workloads for the system
  ○ Identify the specifications and expectations of performance and reliability for supporting the science
  ○ Develop a testing plan to ensure that the final installation has met those requirements

● Chicoma is among the earliest installations of the HPE Cray EX system, running the Shasta Architecture
  ○ Integration and testing activities continue to date
  ○ Continued testing ensures that the system can support a synthetic workload representative of the science for which it is intended
Acceptance Testing Approach

● Integration Testing
  ○ New architecture and progression of system software while developing the test suite required integration testing

● Functionality testing was accomplished during this phase
  ○ Evaluating the readiness of the Cray Programming Environment (CPE) to support workloads
  ○ Testing viability of containerized FEs to host the harness
  ○ Unprivileged container testing using Charliecloud
  ○ Usability of the supplied GROMACS application with COVID-19 study .tpr file
  ○ Scaling tests for MPI applications
  ○ Setting up Pavilion configurations and developing Acceptance Test Suite

● Seven Weeks from Plan Draft to Running Acceptance Tests
  ○ Drafted Testing Plan - July 14, 2020
  ○ Implemented Plan - September 3, 2020
**Test Suite**

**DGEMM** - single node performance

**ExaMiniMD** - proxy MD application

**GROMACS** Covid-19 problem - real world application

**HPCG** - full system benchmark

**HPL** (8 nodes, full system, single node) - various sized benchmark

**LULESH** - proxy application

**MILC7** - Mini app QCD problem

**QuickSilver** - CTS Mini App

**Stream** - Memory benchmark

**SystemConfidence** - network latency benchmark

**VPIC** - Kinetic plasma modeling simulation

**Intel MPI Benchmarks** - MPI-1 benchmark suite
Pavilion2 HPC Test Harness

Pavilion is a Python 3 (3.5+) based framework for running and analyzing tests targeting HPC systems

● Maintained by LANL’s High Performance Computing Environments Group and is open-sourced for community contributions & usage
● Supplies a framework for creating sophisticated YAML configurations to automate the workflow of running jobs on HPC systems
● Plugin components include those for gathering system data, adding additional schedulers, parsing test results, and more
● Pavilion outputs results of every test in a json log file, which then is able to be processed by a number of analysis utilities
● https://github.com/hpc/pavilion2/
Splunk - Data Visualization Tool for Test Results

- Splunk is a flexible data analytics platform that enables searching, monitoring and analyzing machine-generated data.
- Captures and correlates data in searchable database for supporting dashboard and graphical displays of data.
- The monitoring infrastructure is supported by a distributed Splunk instance on every network, gathering large temporal data for analysis.
- LANL’s Splunk distributed monitoring infrastructure indexed Chicoma test result logs to enable automated results analysis and correlation of system events to test underperformance/failures.
Acceptance Testing Results Summary

- **Passing Tests Count**: 13,998
- **Failing Tests Count**: 61
- **Success Percent**: 99.45%

- 1 of 61 failures was legitimate
- HPCG test mis-configuration led to 60 failures
- 225+ TFLOP/s - 256 nodes

Los Alamos National Laboratory
Load Testing

- **59 Hours** with no hardware failures or system related test failures

Only failures were due to oversubscribing or running too large a problem

---

![Load Testing Chart](chart.png)

<table>
<thead>
<tr>
<th>Table Count Tests 1599400800 - 1599699600</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>testname</strong></td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>dgemm</td>
</tr>
<tr>
<td>examini-guai</td>
</tr>
<tr>
<td>gromacs</td>
</tr>
<tr>
<td>hello_mpi</td>
</tr>
<tr>
<td>hpcg-guai</td>
</tr>
<tr>
<td>hpl</td>
</tr>
<tr>
<td>hpl-8</td>
</tr>
<tr>
<td>hpl-full</td>
</tr>
<tr>
<td>imb</td>
</tr>
<tr>
<td>lulesh_single_node</td>
</tr>
<tr>
<td>milc7</td>
</tr>
<tr>
<td>quicksilver</td>
</tr>
<tr>
<td>stream</td>
</tr>
<tr>
<td>sysconfidence</td>
</tr>
<tr>
<td>vpic</td>
</tr>
</tbody>
</table>

- **25,132** TEST SUCCESSES
- **99.10%** Success Percentage
HPL-8 Node Benchmark

62 GFLOP/s diff between fastest/slowest

Problem Specs Selected to Match Integration Team's 8-Node HPL Size and Runtime Configs
HPL Single Node Benchmarks

{ Some interesting HPL spikes in std dev }
Stream Memory Benchmark

Stream Single Node guaje triad Rate

Stream Single Node guaje triad Rate
Full System
High Performance Conjugate Gradient (HPCG)
QuickSilver Proxy Application

QuickSilver

- MPI/MPI-OMP proxy application
- Included in the CTS Benchmarks Suite
- Solves a simplified dynamic Monte Carlo particle transport problem.
- Its performance is bound by poor vectorization potential, latency bound table look-ups and a heavily branching or divergent code path.
VPIC LPI 3D Deck (Lyin-Sequoia)

Vector Particle-In-Cell (VPIC)

- Simulation code for modeling kinetic plasmas on one, two, or three dimensions.
- It employs a second-order, explicit, leapfrog algorithm to update charged particle positions and velocities in order to solve relativistic kinetic equations.
- The input deck, a modified version of lyin_sequoia problem conducted, exercises the problem that Lawrence Livermore National Laboratory used to evaluate their Sequoia system’s potential to model the interaction of realistic fast-ignition-scale lasers with dense plasmas in three dimensions.
Intel MPI Benchmarks

![Graph showing performance metrics for different MPI benchmark tests. The graph compares various performance indicators across different MPI implementations and configurations.]
ExaMiniMD

ECP Proxy Application (Kokkos) Simplified MD Simulation

ExaMiniMD guaje

atomsteps_per_s

atomsteps_per_procs

steps_per_s

ExaMiniMD Time Force

ExaMiniMD Time Neighbor

ExaMiniMD Time Communication

Los Alamos
NATIONAL LABORATORY
The MILC Code is a body of high performance research software written in C for doing SU(3) lattice gauge theory on high performance computers.
DGEMM
● DGEMM was built with the Cray Libsci package and we weren’t able to get the job to spread to both sockets of the nodes.
● We’re working on a DGEMM built with OpenBLAS to see if we can overcome this issue, for improved performance.
GROMACS - COVID-19 Simulation

One major motivation for this effort was to ensure MD problems would run on the system. The repeatability of successful runs with a real GROMACS simulation proves it.
**System Confidence Network Latency Test**

- Captures latency (usecs) and rate (MB/s)
- Consistent ~1 sec latency max measured for all buffer lengths
- Could be a test anomaly
Conclusions

- Chicoma was “accepted” by LANL after demonstrating that it was capable of sustaining a workload and measuring acceptable performance.
- Chicoma was constructed to serve as the IC Program’s Platform for supporting COVID-19 studies.
- Chicoma is currently undergoing an upgrade to Shasta v1.4.
- Tests will be repeated after that upgrade to ensure continued stability and performance of the machine.
- Chicoma is currently running in a pre-production mode at LANL while efforts to fully integrate into production environments are underway.
- Users are using the pre-production system to conduct their research for the IC Program.


**Future Work**

- Pavilion tests are being developed to target unprivileged containerized runtimes on HPC resources at LANL.
- This effort proved that Pavilion was able to satisfy the requirements to conduct Acceptance testing of future procurements.
- Test implementations under Pavilion for Chicoma acceptance will be re-run during the course of transitioning the Chicoma system to full-production.
- Results comparison of the initial baselined results will be rerun with upgrades, including the upgrade to Shasta v1.4, and conducted over the life-cycle of the machine to:
  - identify any performance degradation
  - support optimization of configurations
  - feed into future procurements