Experiences Scaling a Production Arm Supercomputer to Petaflops and Beyond

Kevin Pedretti for the Astra Team

POCs: Jim Laros (Platform), Kevin Pedretti (Software Stack), Si Hammond (Apps)

jhlaros@sandia.gov, ktpedre@sandia.gov, sdammo@sandia.gov
It Takes an Incredible Team…

• DOE Headquarters:
  • Thuc Hoang
  • Mark Anderson
• Sandia Procurement
• Sandia Facilities
• Incredible Sandia Team
• Colleagues at LLNL and LANL
  • Mike Lang
  • Rob Neely
  • Mike Collette
  • Alan Dayton
  • Trent D’Hooge
  • Todd Gamblin
  • Robin Goldstone
  • Anna Pietarila Graham
  • Sam Gutierrez
  • Steve Langer
  • Matt Leininger
  • Matt Legendre
  • Pat McCormick
  • David Nystrom
  • Howard Pritchard
  • Dave Rich
  • And loads more …

• HPE:
  • Mike V. and Nic Dube
  • Andy Warner
  • Erik Jacobson
  • John D’Arcy
  • Steve Cruso
  • Lori Gilbertson
  • Meredydd Evans
  • Cheng Liao
  • John Baron
  • Kevin Jameson
  • Tim Wilcox
  • Charles Hanna
  • Michael Craig
  • Patrick Raymond
  • And loads more …

• Cavium/Marvel:
  • Giri Chukkapalli (now NVIDIA)
  • Todd Cunningham
  • Larry Wikelius
  • Raj Sharma Govindaiah
  • Kiet Tran
  • Joel James
  • And loads more…

• ARM:
  • ARM Research Team!
  • ARM Compiler Team!
  • ARM Math Libraries!
  • And loads more…
It Takes an Incredible Team...
Outline

• Astra Overview
• ATSE Software Stack
• Recent Application Results
• Conclusion – HPC on Arm, are we there yet?
**Vanguard Program: Advanced Technology Prototype Systems**

**ASC Test Beds**
- Small testbeds (~10-100 nodes)
- Breadth of architectures Key
- Brave users

**Vanguard**
- Larger-scale experimental systems
- Focused efforts to mature new technologies
- Broader user-base
- Not production, seek to increase technology and vendor choices
- DOE/NNSA Tri-lab resource

**Production Platforms**
- Leadership-class systems (Petascale, Exascale, ...)
- Advanced technologies, sometimes first-of-kind
- Broad user-base
- Production use

**ATS and CTS Platforms**

---

>Astra is the first Vanguard Platform
Astra Node Architecture

- **2,592** HPE Apollo 70 compute nodes
  - Cavium Thunder-X2 Arm SoC, 28 core, 2.0 GHz
  - 5,184 CPUs, 145,152 cores, 2.3 PFLOPs system peak
  - 128GB DDR Memory per node **(8 memory channels per socket)**
  - Aggregate capacity: 332 TB, Aggregate Bandwidth: 885 TB/s
- Mellanox IB EDR, ConnectX-5
- HPE Apollo 4520 All–flash storage, Lustre parallel file-system
  - Capacity: 990 TB (usable)
  - Bandwidth 244 GB/s

8 GB DDR4-2666 DR

Cavium Thunder-X2 ARM v8.1
28 cores @ 2.0 GHz

Mellanox ConnectX-5 OCP Network Interface

1 EDR link, 100 Gbps
Astra System Architecture

- 36 compute racks (9 scalable units, each 4 racks)
- 2592 compute nodes (5184 TX2 processors)
- 3 IB spine switches (each 540-port)
Astra System Architecture

- 36 compute racks
  (9 scalable units, each 4 racks)
- 2592 compute nodes
  (5184 TX2 processors)
- 3 IB spine switches
  (each 540-port)
Vanguard-Astra: Timeline

725 Facility Build and Final Preparation

5 Months, RFP to Contract Signed

RFP Development
- 20 Dec 2017: RFP Released
- 8 Feb 2018: RFP Responses
- 21 Feb 2018: Review and Selection

Negotiations, SOW Development
- 1 June 2018: Contract Awarded
- 24 August 2018: L2 Milestone Completed
- 4 Sept 2018: 1st Hardware Delivered
- 28 Sept 2018: Final Hardware Delivered

FY Fiscal Milestones Completed

System Integration
- 11 Oct 2018: Installation Completed
- 23 Oct 2018: First Large-scale Runs (1022)
- 1 Nov 2018: HPL Submission 1.529 PF (2238)
- 15 Oct 2018: Facility Completed

Cont. System Integration and Stabilization
- Friendly Users: 8 Jan. 2019
- Restricted Network Access: 6 May 2019
- L2 Milestone: Sept 4th 2019

June 2019 Top500:
- HPL: #156 @ 1.758 PFlops/s on 2428 nodes (up from #204)
- HPCG: #29 @ 90.92 TFlop/s on 2385 nodes (up from #38)
Real-Time System Monitoring Has Been Key

- Tools: {BMC, PDU, Syslog, TX2MON} + TimescaleDB + Grafana

HPL ~1.2 MW
STREAM ~1 MW
Outline

• Astra Overview

• ATSE Software Stack

• Recent Application Results

• Conclusion – HPC on Arm, are we there yet?
Astra Collaboration with HPE’s HPC Software Stack

HPE’s HPC Software Stack

• HPE:
  • HPE Cluster Manager
  • HPE MPI (+ XPMEM)

• Arm:
  • Arm HPC Compilers
  • Arm Math Libraries
  • Allinea Tools

• Mellanox-OFED & HPC-X
• RedHat 7.x for aarch64
HPCM Provides Scalable System Management for Astra

- **HPCM**: HPE Performance Cluster Manager
  - Merger of HPE CMU with SGI Icebox stack
  - New product at time of Astra deployment
- **Collaboration resulted in new capabilities**
  - Support for hierarchical leader nodes for non-Icebox clusters (aka “Flat Clusters”)
    - **Demonstrated boot of 2592 nodes in < 10 min**
  - Resilient leader node failover
  - Scalable BIOS upgrades and configuration
  - Ability to deploy TOSS images (Tri-lab Operating System Stack)

Gluster image store, NFSroot exported by Leaders to Compute Nodes
TOSS Provides Robust Base OS for Astra / ARM

- **TOSS**: Tri-lab Operating System Stack  
  (Lead: LLNL, LANL, SNL)
  - Targets commodity technology systems (model: vendors provide HW, labs provide SW)
  - Red Hat 7 based; x86_64, ppc64le, and aarch64
  - ~4K packages on all archs, 200+ specific to TOSS
  - Partnership with RedHat with direct support

- **Astra-related activities**
  - Lustre enablement and bringup
  - Added support for Mellanox OFED InfiniBand stack, needed for advanced IB features
  - Debugged Linux Kernel issues on Arm, scale of Astra revealed bugs not previously seen
    - Kworker CPU hang – fix was in Linux upstream, but not in RedHat. Patch added to TOSS Linux kernel.
    - Sys_getdents64 oops – rare hang at job cleanup / exit.
      Actively debugging with RedHat + Marvell + HPE + Mellanox
ATSE is an Integrated Software Environment for ASC Workloads

- **Advanced Tri-lab Software Environment**
  - User-facing programming environment co-developed with Astra
  - Provides a common set of libraries and tools used by ASC codes
  - Integrates with TOSS and the vendor software stack
  - Derived from OpenHPC package recipes, similar look and feel (add uarch optimizations, static libraries, -fPIC, add missing libs)

- **FY19 Accomplishments**
  - Deployed TOSS + ATSE at transition to SRN (May’19)
  - Developed ATSE 1.2 with support for 2x compilers and 2x MPIs: {GNU7, ARM} x {OpenMPI3, HPE-MPI}
  - Built Trilinos and many ASC apps with ATSE
  - Packaged ATSE containers and tested up to 2048 nodes

- **Future Directions**
  - Migrate to Spack Stacks build
  - Add support for SNL adv. arch testbeds
  - Collaboration with RIKEN on McKernel
Containerized SPARC HIFiRE-1 on Astra

In job script:

```bash
mpirun \\
    --map-by core \\
    --bind-to core \\
    singularity exec atse-astra-1.2.1.simg \\
    container_startup.sh
```

carrier_startup.sh

```bash
#!/bin/bash
module purge
module load devpack-gnu7
./sparc
```

Early Results: SPARC on Astra, 56 MPI processes per node

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Trials</th>
<th>Native (seconds)</th>
<th>Container (seconds)</th>
<th>% Diff vs. Native</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>2</td>
<td>8164</td>
<td>8169</td>
<td>+ 0.1%</td>
</tr>
<tr>
<td>256</td>
<td>3</td>
<td>4473</td>
<td>4505</td>
<td>+ 0.7%</td>
</tr>
<tr>
<td>512</td>
<td>3</td>
<td>2634</td>
<td>2636</td>
<td>+ 0.1%</td>
</tr>
<tr>
<td>1024</td>
<td>1*</td>
<td>1827</td>
<td>1762</td>
<td>- 3.6%</td>
</tr>
<tr>
<td>2048</td>
<td>2</td>
<td>1412</td>
<td>1429</td>
<td>+ 1.2 %</td>
</tr>
</tbody>
</table>

Points:
- Supporting SPARC containerized build & deployment on Astra
- Enables easy test of new or old ATSE software stacks
- Near-native performance using a container
- Testing HIFiRE-1 Experiment (MacLean et al. 2008)
Outline

• Astra Overview

• ATSE Software Stack

• Recent Application Results

• Conclusion – HPC on Arm, are we there yet?
Applications ported during open and restricted phases:

- SNL: SPARC, EMPIRE, SPARTA, Xyce, NALU, HOMME-X, LAMMPS, CTH, Zapotec
- LANL: FLAG, PARTISN, VPIC
- LLNL: ALE3D, Ares, PF3D

Utilized ATSE provided software stack and modules

- Early work on ATSE using testbeds helped to iron out some initial issues
- Performance results vary, in some cases Trinity Haswell/CTS-1 are faster, others are slower
- Astra shows good scalability out to > 2,048 nodes
- Early indications are that still room for improvement in compilers and math libraries (subject of continuing Astra projects)
## Peak System Performance

<table>
<thead>
<tr>
<th>LINPACK FLOP Rates (per Node)</th>
<th>CTS1</th>
<th>Trinity</th>
<th>Sierra</th>
<th>Astra</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Perf</strong> 1.09 TF/s</td>
<td>~0.86 TF/s</td>
<td>~2.06 TF/s</td>
<td>~1 TF/s</td>
<td>~21.91 TF/s</td>
</tr>
<tr>
<td><strong>Rel</strong> 1.00X</td>
<td>0.79X</td>
<td>1.89X</td>
<td>0.91X</td>
<td>20.01X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory Bandwidth (STREAM) (per Node)</th>
<th>CTS1</th>
<th>Trinity</th>
<th>Sierra</th>
<th>Astra</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Perf</strong> ~136 GB/s</td>
<td>~120 GB/s</td>
<td>~90 GB/s / ~350 GB/s</td>
<td>~270GB/s</td>
<td>~850 GB/s x 4 = ~3.4 TB/s</td>
</tr>
<tr>
<td><strong>Rel</strong> 1.00X</td>
<td>0.88X</td>
<td>0.66X / 2.57X</td>
<td>1.99X</td>
<td>25.00X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power (Max TDP, per Node)</th>
<th>CTS1</th>
<th>Trinity</th>
<th>Sierra</th>
<th>Astra</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Watts</strong> 120W x 2 = 240W</td>
<td>135W x 2 = 270W</td>
<td>~250W</td>
<td>190W x 2 = 380W</td>
<td>~300W x 4 = 1.2kW</td>
</tr>
<tr>
<td><strong>Rel</strong> 1.00X</td>
<td>1.13X</td>
<td>1.04X</td>
<td>1.58X</td>
<td>5.00X</td>
</tr>
</tbody>
</table>

Guidance figures, used peak values for benchmarks and TDP
• SPARC is Sandia’s latest CFD modeling code
  • Developed under NNSA ATDM Program
  • Written to be threaded and vectorized
  • Uses Kokkos programming abstractions
  • Approximately 2-3M lines of code for binary (including Trilinos packages, mostly C++, tiny bit of Fortran)

• Mixture of assembly and solve phases

• Successfully compiles with GCC and Arm HPC compilers on Astra

• Results show performance with Arm HPC compiler varies from 0.5% faster than GCC to 10% slower
  • This seems to be consistent across our code portfolio at present
Similar performance of Trinity-Haswell and Astra (MPI Only, performance is within 10% except for XL blob meshes which were run on fewer nodes for Astra)

• Similar scaling behavior between platforms

Work by Paul Lin and EMPIRE Team
• Arm has OpenMP runtime optimized for larger thread counts (see this often in benchmarks).
  GNU and Arm comparable at smaller thread counts
• Figure left is initial results on Astra, results on right are with a platform-agnostic improvement to the sort routines

Work by Paul Lin and EMPIRE Team
CTH (Hydrodynamics, Fortran)

CTH Shape Charge Multi-Material Problem, Weak Scaled

- CTH uses significant number of Fortran features (mixture from FORTRAN-IV to Fortran-90)
  - Large complex code which is extremely well trusted by analysts, known to be challenging with Fortran compilers on new platforms
- Successfully compiles with Arm Flang (used for these results) and ATSE-GCC installs

Work by Courtenay Vaughan
• Climate modeling code which is partially developed at Sandia (ASCR)
  • Known to drive components and third parties libraries very hard (frequently the first to find issues during porting)
  • Strong driver for improvements in Trillinos solver libraries across DOE platforms
• Good scalability (want to see near straight lines if possible)
• Recent SMT-2 results are around 10% faster
Outline

• Astra Overview

• ATSE Software Stack

• Recent Application Results

• Conclusion – HPC on Arm, are we there yet?
HPC on Arm, are we there yet? ... Yes

• Basic HPC components supported and demonstrated @ scale
  • InfiniBand, UCX, MPI, Lustre, Linux, SLURM, ...
• Compilers and math libraries work sufficiently well to get codes running
• Performance competitive with leading CPU alternatives
• Offerings from a range of integrators + technology providers
HPC on Arm, are we there yet? ... Yes

- Basic HPC components supported and demonstrated @ scale
  - InfiniBand, UCX, MPI, Lustre, Linux, SLURM, ...
- Compilers and math libraries work sufficiently well to get codes running
- Performance competitive with leading CPU alternatives
- Offerings from a range of integrators + technology providers

- SVE not proven yet, lack of accelerator options (changing)
- Performance not tuned in many packages / kernels yet
  - Need threaded and vectorized versions of kernels
- Still need work on profilers, debuggers, and memory correctness
- Lots of room for HPC specialization + improved energy efficiency
Questions?
Exceptional Service in the National Interest