Requirements for Parallel I/O, Visualization and Analysis

Prabhat\textsuperscript{1}, Quincey Koziol\textsuperscript{2}

\textsuperscript{1}LBL/NERSC
\textsuperscript{2}The HDF Group

NERSC ASCR Requirements for 2017
January 15, 2014
LBNL
1. Project Description

• m636 repo

• LBL Vis Base Program (Bethel PI) [PM: Nowell]
  • Conduct fundamental and applied vis/analytics R&D to address exascale challenges

• ExaHDF5 Project (Prabhat, Quincey PIs) [PM: Nowell]
  • Scale Parallel I/O, and data management technologies for current petascale and future exascale hardware

• MANTISSA Project (Prabhat PI) [PM: Landsberg]
  • Develop scalable statistics and machine learning techniques for data-centric science
1. Project Goals

- Demonstrate successful application of visualization techniques to PB sized output

- Demonstrate HDF5 (and production I/O stack) scaling on current petascale and future exascale platforms

- Demonstrate sophisticated Big Data analytics techniques applied to TB sized complex, multi-modal datasets (simulations, experiments, observations)
2. I/O, Vis, Analysis Strategies (1/2)

- **We approach scaling and performance optimization by:**
  - Domain decomposition (spatial, temporal, etc)
  - Collective buffering, compression, auto-tuning, minimizing synchronization points

- **Libraries and Codes:**
  - Simulation codes: VPIC, Chombo, FLASH, MOAB, SPH, IMPACT-Z, VORPAL, Warp, CAM5
  - I/O: HDF5, NetCDF
  - Vis: VTK, VisIt, Paraview

- **Characterization:**
  - I/O: particle, block structured, unstructured, AMR meshes
  - Vis: volume rendering, ray casting, streamline computation
  - Analysis: Big Data motifs (sparse/dense linear algebra, stochastic optimization, graph analytics)
2. I/O, Vis, Analysis Strategies (2/2)

• Our biggest challenges are:
  • Under-provisioned I/O resources
  • Insufficient funding for I/O R&D
  • Scaling of storage technologies wrt compute and interconnect
  • Characterization of ‘Big Data’ analytics space

• Our parallel scaling is limited by:
  • Optimized hardware and middleware implementations
  • High computational complexity of analytical algorithms

• We expect our approach and/or codes to change by 2017:
  • Utilize Burst Buffer/NVRAM technology
  • Asynchronous and Fault Tolerance in HDF5
  • Novel statistical and machine learning algorithms for analytics
## 3. Current/Projected HPC Usage

<table>
<thead>
<tr>
<th></th>
<th>Compute Hours</th>
<th>Target Concurrency</th>
<th>Data read/ written per run</th>
<th>Memory per node</th>
<th>Required software</th>
<th>Resources used</th>
<th>Data Stored</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current 2014</td>
<td>4M</td>
<td>10K-150K</td>
<td>100GB-30TB</td>
<td>100%</td>
<td>HDF5, NetCDF, MPI, MPI-IO, pthreads, OpenMP, ScalaPACK, BLAS</td>
<td>/scratch/project</td>
<td>250-500 TB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated 2017</td>
<td>30M</td>
<td>10K-5M</td>
<td>100GB-1PB</td>
<td>100%</td>
<td>HDF5, NetCDF, MPI, MPI-IO, MPI+X?? ScalaPACK, BLAS</td>
<td>/scratch/project Burst Buffers</td>
<td>1-5 PB</td>
</tr>
</tbody>
</table>
5. Strategies for New Architectures (1/3)

- **Visualization Research:**
  - Experimenting with hybrid programming models (MPI+X)
  - Reasonable success with pthreads/OpenMP
  - Tests conducted on hopper, titan

- **HDF5:**
  - Production ready on all DOE centers [NERSC, ALCF, OLCF, etc]
  - Programming model question is not as relevant; need OS/runtime to provide dedicated core for processing
  - Additional cores for compression; offload to NVRAM

- **Analytics software:**
  - Extreme levels of concurrency (million-way parallelism) is probably not required
  - Start with MPI+OpenMP, utilize vendor BLAS implementations
  - Examining MIC architecture in collaboration with Intel Research
5. Strategies for New Architectures (2/3)

• Have there been or are there now other funded groups or researchers engaged to help with these activities?
  – Visualization: yes
  – Parallel I/O, Analysis: no

• Explain your strategy for transitioning to energy-efficient, manycore architectures
  – Parallel I/O: hardware/software stack is in flux. Don’t have sufficient funding at the moment
  – Analysis: identify Big Data motifs, apply for DOE/ASCR funding to pursue careful exploration of motifs and energy-efficient/manycore issues
5. Strategies for New Architectures (3/3)

• What role should DOE/ASCR/NERSC play in the transition to these architectures?
  – Additional resources needed at NERSC to systematically explore issues related to energy efficiency and manycore
  – NERSC Staff, User training
  – Fund efforts similar to petascale post-doc program
  – Fund industry + researchers (i.e. Intel/Whamcloud fastforward program) to explore I/O middleware stack

• Other considerations:
  – Analysis algorithms will rely on dense/sparse linear algebra. Optimized implementations for manycore architectures will be important.
5. Special I/O Needs

• Collaborators (VPIC, Chombo, etc) use checkpoint/restart
  • Fast checkpoint/restart performance is key
  • Ideally, flush system memory to storage in 15-30 minutes

• QoS on shared resources (interconnect, I/O)

• Burst Buffer use cases:
  • Definitely relevant to accelerating Parallel I/O operations (reads and writes)
  • BUT we need software stack to intelligently use hardware
    • HDF5, Lustre/GPFS filesystem, etc
  • Relevant to In-situ/In-Transit vis. (GLEAN, Paraview, VisIt)
6. Summary

• New science results:
  • Facilitating multiple science code teams to store, analyze and visualize output

• Recommendations on NERSC services
  • Generally happy with professional quality of services rendered by NERSC, and Cray/NERSC staff collaboration
  • Lagging behind other HPC centers in terms of I/O hardware provisioning
    • Mira: 240 GB/s, Sequoia: ~1TB/s, BlueWaters: 1 TB/s, Edison: 70 GB/s
  • Risk of entering vicious cycle wherein Hero-scale runs are never attempted at NERSC

• “Expanded HPC resources”
  • I/O hardware, bandwidth, QoS
  • Burst Buffer technology