NERSC REQUIREMENTS FOR ENABLING CS RESEARCH IN EXASCALE DATA ANALYTICS

Nagiza F. Samatova
samatovan@ornl.gov
Oak Ridge National Laboratory
North Carolina State University

January 5, 2011
ACKNOWLEDGEMENTS

• DOE ASCR for funding CS research in exascale data analytics

• Arie Shoshani, Alok Choudhary, Rob Ross, etc
  – PI’s and co-PI’s on the projects

• LCF Facilities at ORNL

• Application Scientists:
  – CS Chang, ORNL
  – Stephane, PPPL
  – Fred Semazzi, NCSU
  – Many others
HARDWARE FOR DATA ANALYTICS IS A FOSTER CHILD

• HW configuration for DA is an after-thought:
  – Has been traditionally optimized for running simulations
  – Whatever is left over is what data analyst should live with
  – DA-driven HW must become the first class citizen on the agenda if we are serious about the exascale

• Infrastructure depends on the DA modality:
  – In situ?
  – Distributed or streamline fashion?
  – Local or global context analysis?
  – Shared among a group of collaborators?
  – Linked to experimental and/or other data archives?
In contrast to simulations, Data Analytics requires a different mix of memory, disk storage, & communication trade-offs.
Each step is a combinatorial optimization problem with different data access patterns.

**Pipeline: Ab Initio Prediction of Protein 3-d Structure**

- **Known structures**
  - 100 GB
  - Search, Optimization, Enumeration

- **Finding Common Motifs**
  - Knowledge-based Energy Tables
  - 3 GB – 5 TB
  - ROSETTA Monte Carlo protein folding
  - Energy Optimization

- **Decoy structures**
  - $10^3$-$10^5$
  - (10^4~50TB)
  - Search, Optimization, Enumeration

- **Finding Maximal Cliques**
  - Cliques of structures
  - 10 GB – 500 TB
  - Merging & Scoring
  - Search

- **Native Structures**
EX: SCIENTIFIC DATA FLOW IN MS PROTEOMICS

Twice a month production runs

Archive Data (~2TB) (grows exponentially):
-- 100MB files
-- 100-1000s scans in each file

14-24 hours per file
One scan access at a time

Results (~2TB)

SEQUEST
DOMAIN-SPECIFIC REALIZATION OF THE SW STACK

Science Question Level

Bioenergy • Bioremediation • Carbon sequestration

Modeling Problem Formulation

Biological Application Level

Structural Modeling
• 3-d protein structure
• Protein docking
• Protein-ligand

Mathematical Problem Formulation

Network Modeling
• Metabolic pathways
• Regulatory & signaling networks
• Protein interaction networks

Data-Intensive Methods for:
• Combinatorial search, enumeration, and optimization
• Information and knowledge fusion
• Dimensionality reduction and feature extraction

Computational Science Level

Scalable Parallel Implementation

Memory, I/O, and Communication Management
• Dynamic load balancing
• Latency hiding
• End-to-end data flow optimization
• Heterogenous patterns of data access

Plug-ins

Other Models

Other Methods

January 5, 2011
Nagiza F. Samatova
END-TO-END DATA ANALYTICS SOFTWARE STACK IS COMPLEX: GENERIC (ALL APPLICATIONS) PERSPECTIVE

Exemplars
CDAT
Kepler
pR
RScaLAPACK
FastBit
pnetCDF

Focus of my talk
THE LESSON LEARNED FROM LINEAR ALGEBRA

“The lesson learned from Linear Algebra”

Pre-conditioner

“conditions a given problem into a form that is more suitable for numerical solution.”

“Dwarfs:” Sparse, Dense, etc.

Solver

highly optimized computational kernel

• ScaLAPACK
• PETSc
• etc
SOFTWARE FOR DATA ANALYTICS IS MORE AD HOC

• Should we adopt this approach from Linear Algebra to Data Analytics at extreme scale? If so, then
  – What are the “Dwarfs” for data analytics?
  – What about the “Preconditioners?”
  – What are the “Computational kernels?”

• Do we/should we have a ScaLAPACK-like library for Exascale Data Analytics?

• What NERSC should/could offer for enabling this activity?
“**Computational Kernels**” Concept is Promising

The frequency of kernel operations in illustrative data mining algorithms and applications.

<table>
<thead>
<tr>
<th>Application</th>
<th>Top 3 Kernels (%)</th>
<th>Sum %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Kernel 1 (%)</td>
<td>Kernel 2 (%)</td>
</tr>
<tr>
<td>K-means</td>
<td>Distance (68)</td>
<td>Center (21)</td>
</tr>
<tr>
<td>Fuzzy K-means</td>
<td>Center (58)</td>
<td>Distance (39)</td>
</tr>
<tr>
<td>BIRCH</td>
<td>Distance (54)</td>
<td>Variance (22)</td>
</tr>
<tr>
<td>HOP</td>
<td>Density (39)</td>
<td>Search (30)</td>
</tr>
<tr>
<td>Naïve Bayesian</td>
<td>probCal (49)</td>
<td>Variance (38)</td>
</tr>
<tr>
<td>ScalParC</td>
<td>Classify (37)</td>
<td>giniCalc (36)</td>
</tr>
<tr>
<td>Apriori</td>
<td>Subset (58)</td>
<td>dataRead (14)</td>
</tr>
<tr>
<td>Eclat</td>
<td>Intersect (39)</td>
<td>addClass (23)</td>
</tr>
<tr>
<td>SVMlight</td>
<td>quotMatrix(57)</td>
<td>quadGrad (38)</td>
</tr>
</tbody>
</table>

Alok Choudhary, NWU, *NU-Minebench*
WHAT ABOUT “PRECONDITIONERS” FOR DATA ANALYTICS?

• How to define a “preconditioner” for data analytics?

Solve a Problem $P_{\text{hard}}$

Directly     Indirectly (via “Preconditioner”):
Reduce a Hard Problem $P_{\text{hard}}$ to a “Better” Problem $P_{\text{better}}$

$P_{\text{hard}} \rightarrow \text{Preconditioner} \rightarrow P_{\text{better}}$

“Better” in terms of:
• Increased throughput
• Faster time-to-solution
• More accurate solution
• Higher data compression rate
• Approximate but real-time solution
IF WE ARE LUCKY...

- and Jack Dongara did most of the work for us:
  - Some data analysis routines call linear algebra functions
  - In R, they are built on top of LAPACK library
- **RScaLAPACK is an R wrapper library to ScaLAPACK**

\[
\begin{align*}
    A &= \text{matrix}(\text{rnorm}(256),16,16) \\
    b &= \text{as.vector}(\text{rnorm}(16))
\end{align*}
\]

Using RScaLAPACK:

- library (RScaLAPACK)
- sla.solve (A,b)
- sla.svd (A)
- sla.prcomp (A)

Using R:

- solve (A,b)
- La.svd (A)
- prcomp (A)

WHAT IF WE ARE NOT THAT LUCKY?
IN SITU PRECONDITIONERS FOR SCIENTIFIC DATA COMPRESSION

• Myth: “Scientific data is almost uncompressible.”

GTS Fusion Simulation Data (Stephane, PPPL)

C&R Data
• ~2TB per C&R
• Every 1 hour
• Two copies
• Keep the last copy

Analysis Data
• ~2TB per run (now)
• Every 10\textsuperscript{th} time step
• Cannot afford storing all b/s of
• Analysis routines and I/O reads
• Matlab analysis routines

V&V Data
• Small
• Every 2\textsuperscript{nd} time step

Expected: 10-fold increase by 2012-2014
# Computing and Storage Resources

<table>
<thead>
<tr>
<th></th>
<th>GYRO</th>
<th>GTS</th>
<th>XGC1*</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Facilities</strong></td>
<td>NERSC/OLCF</td>
<td>NERSC/OLCF</td>
<td>NERSC/OLCF</td>
</tr>
<tr>
<td><strong>Architectures</strong></td>
<td>XT5, Power, Cluster</td>
<td>XT5</td>
<td>XT5</td>
</tr>
<tr>
<td><strong>Years</strong></td>
<td>Present</td>
<td>In 5 yrs</td>
<td>Present</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In 5 yrs</td>
<td></td>
</tr>
<tr>
<td><strong>Hrs used/year</strong></td>
<td>30M</td>
<td>50M</td>
<td>24M</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NERSC’09 used</strong></td>
<td>1.2Mhrs</td>
<td>~2Mhrs</td>
<td>~8M hrs</td>
</tr>
<tr>
<td><strong>#Cores per run</strong></td>
<td>512</td>
<td>512</td>
<td>8-98K</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Wall clock/run</strong></td>
<td>12</td>
<td>24</td>
<td>72 Hrs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Memory/run</strong></td>
<td>512GB</td>
<td>1.024TB</td>
<td>16-100T</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Min Memory/core</strong></td>
<td>1GB</td>
<td>2GB</td>
<td>1GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Read/Write data</strong></td>
<td>2.5TB</td>
<td>8TB</td>
<td>5TB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Checkpoint size</strong></td>
<td>4GB</td>
<td>8GB</td>
<td>1-8GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data in/out nersc</strong></td>
<td>5GB/run</td>
<td>10GB/run</td>
<td>10GB/day</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>On-line storage</strong></td>
<td>4TB/10K</td>
<td>8TB/10K</td>
<td>4TB/3K</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Off-line storage</strong></td>
<td>25GB</td>
<td>100GB</td>
<td>1TB/30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Unstructured mesh

---

From C.S. Chang’s Talk at NERSC
**S-preconditioner for Analysis Data Compression**

- Analysis data is stored every *N*-th time step:
  - Lossy data reduction
  - Data is almost random—hard/impossible to compress; <10% lossless
  - *N* is defined ad hoc (*N*=10 for GTS, *N*=100 for Supernova)

While Compression Ratio is growing up from 87.06% to 87.44%, the Pearson Correlation dropped from 0.994 to 0.937.

Stephane, PPPL: “*With this data quality and data reduction rate, I can test many more hypothesis using my analysis tools.*”

**PhD students:** Ye Jin and Sriram Lakshminaraisimhan
BFA-PRECONDITIONER FOR C&R DATA COMPRESSION

• C&R Data Compression:
  – Must be lossless
  – Must be fast

• Impact of BFA-preconditioner:
  – 8x throughput increase for bzip
  – 4x throughput increase for gzip
  – 1.41 compression ratio (CR) for zpaq with BFA-precond
  – 1.33 vs. 1.17 CR for bzip2 with vs. w/o BFA-precond.
  – 1.32 vs. 1.19 CR for zlib with vs. w/o BFA-precond.
**BC-preconditioner for Underdetermined Classification Problem (BENCH)**

- Accuracy increase by 13%-16%
- Across different classifiers
- On data with <100 samples >d=4,000-7,000 dimensions—underdetermined problems
- When applied to seasonal hurricane prediction (d>35K), correlation with observed improved from 0.64 to 0.92-0.96

---

**Bayesian Belief Network (BBN)**

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Single classifier</th>
<th>BENCH ensemble</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBN</td>
<td>82.856</td>
<td>97.142</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>82.856</td>
<td>95.714</td>
</tr>
<tr>
<td>SVM</td>
<td>91.426</td>
<td>97.142</td>
</tr>
</tbody>
</table>

PhD students: Ann Pansombut, William Hendrix, Zhengzhang Chen, Doel Gonzalez
SE-preconditioner for Contrasting Frequent Subgraph Mining (NIBBS-Search)

Exact Algorithm versus NIBBS-Search
(98 Genome-Scale Metabolic Networks, 49 Positive, 49 Negative)

Runtime of exact algorithm grows exponentially (unable to complete run)

The NIBBS-Search algorithm completes in a matter of seconds

Empirical tests show that the NIBBS-Search subgraphs are significantly close approximations of maximally-biased subgraphs

PhD students: Matt Schmidt, Andrea Rocha
# Dark Fermentative Bio-Hydrogen Production Pathways are Identified with NIBBS-Search

## Acetate Pathway

<table>
<thead>
<tr>
<th>EC Number</th>
<th>Enzyme Name</th>
<th>T-Test</th>
<th>NIBS</th>
<th>Mutual Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.7.2.1</td>
<td>acetate kinase;</td>
<td></td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>2.3.1.8</td>
<td>phosphotransacetylase</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>4.2.1.55</td>
<td>crotonase</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>2.3.1.54</td>
<td>pyruvate formate lyase</td>
<td></td>
<td>TRUE</td>
<td></td>
</tr>
</tbody>
</table>

## Butyrate Pathway

<table>
<thead>
<tr>
<th>EC Number</th>
<th>Enzyme Name</th>
<th>T-Test</th>
<th>NIBS</th>
<th>Mutual Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3.99.2</td>
<td>butyryl-CoA dehydrogenase;</td>
<td></td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>2.7.2.7</td>
<td>butyrate kinase</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>1.1.1.157</td>
<td>3-hydroxybutyryl-CoA dehydrogenase;</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>2.3.1.19</td>
<td>phosphate butyryltransferase;</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>2.3.1.9</td>
<td>acetyl-CoA C-acetyltransferase;</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>2.3.1.54</td>
<td>pyruvate formate lyase</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>4.2.1.55</td>
<td>crotonase</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
</tbody>
</table>

## Formate Pathway

<table>
<thead>
<tr>
<th>EC Number</th>
<th>Enzyme Name</th>
<th>T-Test</th>
<th>NIBS</th>
<th>Mutual Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.12.1.2</td>
<td>formate dehydrogenase</td>
<td>TRUE</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>1.2.7.1</td>
<td>pyruvate ferredoxin oxidoreductase</td>
<td></td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>1.12.7.2</td>
<td>ferredoxin hydrogenase</td>
<td></td>
<td>TRUE</td>
<td></td>
</tr>
</tbody>
</table>

---

**PhD students:** Andrea Rocha and Kanchana; **Undergrad:** Katie Shpanskaya
CS DATA ANALYSIS RESEARCH “WORKFLOW” — ITERATIVE PROCESS w/ SIGNIFICANT RESOURCE NEEDS

User (e.g., Stephane, PPPL)

Problem
Sample Data

Generate Idea(s)

Prototype

Evaluate

HPC Implementation

GBs

R, pR, Matlab, Python
Performance Tools, Cross-Validation

MPI, CUDA-GPU, etc.

• Many data sets (from different Apps)
• SUPER-production runs (~200 TBs)
• Many parameter sensitivity analysis runs

Publish
Put into production