Convergence of Supercomputing and Extreme Big Data on the TSUBAME Supercomputer Exascale

松岡聡・東工大
Satoshi Matsuoka
Tokyo Institute of Technology

OpenSFS @ Tokyo
2013/10/17
Themes of the Day...

• How do you respond to the followings?
• “We don’t need to invest in all that supercomputer R&D stuff; we invest into clouds, mobiles, etc., for big data and we will just leverage off those…”
• “Sure, supercomputers are pretty big, but giants Google/Amazon/... will have enough resource in the cloud for big data, so we will just use those…”
The current “Big Data” are not really that Big...

• Typical “real” definition: “Mining people’s privacy data to make money”

• Corporate data are usually in data warehoused silo - > limited volume, in Gigabytes~Terabytes, seldom Petabytes.

• Processing involve simple \( O(n) \) algorithms, or those that can be accelerated with DB-inherited indexing algorithms

• Executed on re-purposed commodity “web” servers linked with 1Gbps networks running Hadoop/HDFS

• Vicious cycle of stagnation in innovations...

• Breaking Down of Corporate Silos⇒ Convergence with Supercomputing with Extreme Big Data
We will have tons of unknown genes

Metagenome analysis

• Directly sequencing uncultured microbiomes obtained from target environment and analyzing the sequence data
  – Finding novel genes from unculturable microorganism
  – Elucidating composition of species/genes of environments

Examples of microbiome

- Human body
- Gut microbiome
- Soil
- Sea

[Slide Courtesy Yutaka Akiyama @ Tokyo Tech.]
Extreme Big Data in Genomics

Impact of new generation sequencers

Sequencing data (bp)/$ becomes x4000 per 5 years
c.f., HPC x33 in 5 years

Lincoln Stein, Genome Biology, vol. 11(5), 2010
Extreme Big Data Example in Social NW rates and volumes are immense

- Facebook:
  - ~1 billion users
  - average 130 friends
  - 30 billion pieces of content shared / month
- Twitter:
  - 500 million active users
  - 340 million tweets / day
- Internet – 100s of exabytes / year
  - 300 million new websites per year
  - 48 hours of video to You Tube per minute
  - 30,000 YouTube videos played per second

Continuous Billion-Scale Social Simulation with Real-Time Streaming Data (Toyotaro Suzumura/IBM-Tokyo Tech)

- Applications
  - Target Area: Planet (Open Street Map)
    - 7 billion people
- Input Data
  - Road Network (Open Street Map) for Planet: 300 GB (XML)
  - Trip data for 7 billion people
    - 10 KB (1 trip) x 7 billion = 70 TB
  - Real-Time Streaming Data (e.g. Social sensor, physical data)
- Simulated Output for 1 Iteration
  - 700 TB

Extreme Big Data in Genomics

Impact of new generation sequencers

Sequencing data (bp)/$ becomes x4000 per 5 years c.f., HPC x33 in 5 years

Future “Extreme Big Data”

- NOT mining Tbytes Silo Data
- Peta~Zetabytes of Data
- Ultra High-BW Data Stream
- Highly Unstructured, Irregular
- Complex correlations between data from multiple sources
- Extreme Capacity, Bandwidth, Compute All Required
“Extreme Big Data” will change everything

• “Breaking down of Silos” (Rajeeb Harza, Intel VP of Technical Computing)

• Already happening in Science & Engineering due to Open Data movement

• More complex analysis algorithms: $O(n \log n)$, $O(m \times n)$, ...

• Will become the NORM for competitiveness reasons.
TSUBAME2.0 Nov. 1, 2010
“The Greenest Production Supercomputer in the World”

TSUBAME2.0: A GPU-centric Green 2.4 Petaflops Supercomputer

Tsubame 2.0: "Tiny" footprint, very power efficient
• Floorspace less than 200m² (2,100 ft²)
• Top-class power efficient machine on the Green 500

TSUBAME 2.0
New Development

System
(42 Racks)
1408 GPU Compute Nodes,
34 Nehalem "Fat Memory" Nodes

Rack
(8 Node Chassis)

Chip
(CPU, GPU)

Intel
NVIDIA

Compute Node
(2 CPUs, 3 GPUs)

Node Chassis
(4 Compute Nodes)

>400GB/s Mem BW
>1.6TB/s Mem BW
>12TB/s Mem BW
>600TB/s Mem BW

220Tbps NW
Bisection BW
1.4MW Max

32nm
40nm

CPU(Westmere EP)
76.8 GFLOPS

GPUs(Tesla M2050)
515 GFLOPS
3 GB

1.6 TFLOPS
55 GB/103 GB

6.7 TFLOPS
220 GB/412 GB

53.6 TFLOPS
1.7 TB/3.2 TB

2.4 PFLOPS
80 TB

~1KW max

Integrated by NEC Corporation
**TSUBAME2.0 Compute Node**

**Thin Node**

- Infiniband QDR x2 (80Gbps)

**1.6 Tflops**

**400GB/s Mem BW**

**80GBps NW**

~1KW max

**Productized as HP ProLiant SL390s**

**HP SL390G7 (Developed for TSUBAME 2.0)**

- GPU: NVIDIA Fermi M2050 x 3
  - 515GFlops, 3GByte memory /GPU
- CPU: Intel Westmere-EP 2.93GHz x2 (12cores/node)
- Multi I/O chips, 72 PCI-e (16 x 4 + 4 x 2) lanes --- 3GPUs + 2 IB QDR
- Memory: 54, 96 GB DDR3-1333
- SSD: 60GBx2, 120GBx2

**Total Perf**

2.4PFlops

Mem: ~100TB

SSD: ~200TB
2010: TSUBAME2.0 as No.1 in Japan

Total 2.4 Petaflops
#4 Top500, Nov. 2010
TSUBAME Wins Awards...

“Greenest Production Supercomputer in the World”
the Green 500
Nov. 2010, June 2011
(#4 Top500 Nov. 2010)
ACM Gordon Bell Prize 2011

Special Achievements in Scalability and Time-to-Solution

“Peta-Scale Phase-Field Simulation for Dendritic Solidification on the TSUBAME 2.0 Supercomputer”
Commendation for Sci & Tech by Ministry of Education 2012

Prize for Sci & Tech, Development Category
Development of Greenest Production Peta-scale Supercomputer

Satoshi Matsuoka, Toshio Endo, Takayuki Aoki
TSUBAME2.0 Storage Overview

Infiniband QDR Network for LNET and Other Services

- SFA10k #1
- SFA10k #2
- SFA10k #3
- SFA10k #4
- SFA10k #5

Parallel File System Volumes

- Lustre 3.6 PB
- GPFS with HSM

Home Volumes 1.2PB

- 2.4 PB HDD + ~4PB Tape
- 130 TB => 500TB~1PB

Grid Storage

- 250 TB, 300TB/s
- Scratch

11PB (7PB HDD, 4PB Tape)
**TSUBAME2.0 Storage Overview**

**TSUBAME2.0 Storage 11PB (7PB HDD, 4PB Tape)**

Infiniband QDR Network for LNET and Other Services

- **QDR IB (×4) × 20**
- **QDR IB (×4) × 8**
- **10GbE × 2**

**Concurrent Parallel I/O** (e.g. MPI-I/O)
- SFA10k #1
- SFA10k #2
- SFA10k #3
- SFA10k #4
- SFA10k #5

**Fine-grained R/W I/O** (checkpoints, temporary files, Big Data processing)
- Home storage for computing nodes
- Cloud-based campus storage services

**Data transfer service** between SCs/CCs

- **250 TB, 300GB/s**
- **Long-Term Backup**
- **2.4 PB HDD + ~4 PB Tape**
- **Thin node SSD**
- **Fat/Medium node SSD**

**Home Volumes** 1.2PB

**HPCI Storage**

- **130 TB => 500TB~1PB**

- **Home storage for computing nodes**
- **Cloud-based campus storage services**

**Read mostly I/O** (data-intensive apps, parallel workflow, parameter survey)

**GPFS with HSM**

**Parallel File System Volumes**

**Scratch**

**GPFS with HSM**

**Home Volumes** 1.2PB
TSUBAME2.0 Storage Usage

- **home0**
- **nest2**
- **lustre0**
- **lustre1**
- **gpfs0**
- **total**

Total 1.4 Petabytes

- **TSUBAME1.0 Data**

GPFS Failure

GPFS Operations

Lustre Expansion

Dates:
- 2010/7/3
- 2010/10/11
- 2011/1/19
- 2011/4/29
- 2011/8/7
- 2011/11/15
- 2012/2/23
- 2012/6/2
- 2012/9/10
- 2012/12/19
Hadoop on TSUBAME (Tsudoop)

- **Script-based invocation**
  - acquire computing nodes via PBS Pro
  - deploy a Hadoop environment on the fly (incl. HDFS)
  - execute a user MapReduce jobs

- **Various FS support**
  - HDFS by aggregating local SSDs
  - Lustre, GPFS (to appear)

- **Customized Hadoop for executing CUDA programs (experimental)**
  - Hybrid Map Task Scheduling
    - Automatically detects map task characteristics by monitoring
    - Scheduling map tasks to minimize overall MapReduce job execution time
    - Extension of Hadoop Pipes features
Towards TSUBAME 3.0
Interim Upgrade TSUBAME2.0 to 2.5
(Early Fall 2013)

- Upgrade the TSUBAME2.0s GPUs
  NVIDIA Fermi M2050 to Kepler K20X

  SFP/DFP peak from 4.8PF/2.4PF => 17PF/5.7PF
c.f. The K Computer 11.2/11.2
  Acceleration of Important Apps
  Considerable Improvement
  Summer 2013

TSUBAME2.0 Compute Node
Fermi GPU 3 x 1408 = 4224 GPUs

Significant Capacity Improvement at low cost & w/o Power Increase

TSUBAME3.0 2H2015
TSUBAME2.0⇒2.5 Thin Node Upgrade

**Thin Node**

- Infiniband QDR x2 (80Gbps)

**Peak Perf.**

- 4.08 Tflops
- ~800GB/s Mem BW
- 80GBps NW
- ~1KW max

**HP SL390G7 (Developed for TSUBAME 2.0, Modified for 2.5)**

- **GPU:** NVIDIA Kepler K20X x 3
  - 1310GFlops, 6GByte Mem (per GPU)

- **CPU:** Intel Westmere-EP 2.93GHz x2

- **Multi I/O chips:** 72 PCI-e (16 x 4 + 4 x 2) lanes --- 3GPUs + 2 IB QDR

- **Memory:** 54, 96 GB DDR3-1333
- **SSD:** 60GBx2, 120GBx2

---

**NVIDIA Fermi**

- M2050
- 1039/515 GFlops

**NVIDIA Kepler**

- K20X
- 3950/1310 GFlops
<table>
<thead>
<tr>
<th></th>
<th>TSUBAME2.0</th>
<th>TSUBAME2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Node Machine</strong></td>
<td>HP Proliant SL390s</td>
<td>← No Change</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
<td>Intel Xeon X5670 (6core 2.93GHz, Westmere) x 2</td>
<td>← No Change</td>
</tr>
<tr>
<td><strong>GPU</strong></td>
<td>NVIDIA Tesla M2050 x 3</td>
<td>NVIDIA Tesla K20X x 3</td>
</tr>
<tr>
<td></td>
<td>- 448 CUDA cores (Fermi)</td>
<td>- 2688 CUDA cores (Kepler)</td>
</tr>
<tr>
<td></td>
<td>- SFP 1.03TFlops</td>
<td>- SFP 3.95TFlops</td>
</tr>
<tr>
<td></td>
<td>- DFP 0.515TFlops</td>
<td>- DFP 1.31TFlops</td>
</tr>
<tr>
<td></td>
<td>- 3GiB GDDR5 memory</td>
<td>- 6GiB GDDR5 memory</td>
</tr>
<tr>
<td></td>
<td>- 150GB Peak, ~90GB/s STREAM Memory BW</td>
<td>- 250GB Peak, ~180GB/s STREAM Memory BW</td>
</tr>
<tr>
<td><strong>Node Performance (incl. CPU Turbo boost)</strong></td>
<td>- SFP 3.40TFlops</td>
<td>- SFP 12.2TFlops</td>
</tr>
<tr>
<td></td>
<td>- DFP 1.70TFlops</td>
<td>- DFP 4.08TFlops</td>
</tr>
<tr>
<td></td>
<td>- ~500GB Peak, ~300GB/s STREAM Memory BW</td>
<td>- ~800GB Peak, ~570GB/s STREAM Memory BW</td>
</tr>
<tr>
<td><strong>TOTAL System</strong></td>
<td>- SFP 4.80PFlops</td>
<td>- SFP 17.1PFlops (x3.6)</td>
</tr>
<tr>
<td></td>
<td>- DFP 2.40PFlops</td>
<td>- DFP 5.76PFlops (x2.4)</td>
</tr>
<tr>
<td></td>
<td>- Peak ~0.70PB/s, STREAM ~0.440PB/s Memory BW</td>
<td>- Peak ~1.16PB/s, STREAM ~0.804PB/s Memory BW (x1.8)</td>
</tr>
</tbody>
</table>
2013: TSUBAME2.5 No.1 in Japan in Single Precision FP, 17 Petaflops

Total
17.1 Petaflops SFP
5.76 Petaflops DFP

K Computer
11.4 Petaflops SFP/DFP

All University Centers
COMBINED 9 Petaflops SFP
Linpack Benchmark

• Linpack: Dense matrix solver by LU decomposition with pivotting
  – Used in Top500/Green500 supercomputer ranking!
• On TSUBAME2.5, we adopted “In-core” algorithm, where the whole matrix data are placed on GPU device memory
  – K20X on T2.5 has 2x larger memory than M2050 on T2.0
  – PCIe communication has relatively larger effects

<table>
<thead>
<tr>
<th></th>
<th>TSUBAME2.0</th>
<th>TSUBAME2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>Out-of-core</td>
<td>In-core</td>
</tr>
<tr>
<td>N (matrix size)</td>
<td>2,490,368</td>
<td>1,760,000</td>
</tr>
<tr>
<td>NB (block size)</td>
<td>1024</td>
<td>192</td>
</tr>
<tr>
<td>Speed (PFlops)</td>
<td><strong>1.192</strong></td>
<td><strong>2.843</strong></td>
</tr>
<tr>
<td>Rank in Top500</td>
<td>No. 4 in 11/2010</td>
<td>TBA in 11/2013</td>
</tr>
<tr>
<td>Power (MWatt)</td>
<td>1.244</td>
<td>0.958</td>
</tr>
<tr>
<td>Speed/Power (GFlops/Watt)</td>
<td><strong>0.958</strong></td>
<td>&gt;2.50x</td>
</tr>
<tr>
<td>Rank in Green500</td>
<td>No. 2 in 11/2010</td>
<td>TBA in 11/2013</td>
</tr>
</tbody>
</table>
High-Performance General Solver for Extremely Large-scale Semidefinite Programming Problems [Fujisawa]

1. Mathematical Programming: one of the most important mathematical programming
2. Many Applications: combinatorial optimization, control theory, structural optimization, quantum chemistry, sensor network location, data mining, etc.

SDPARA is a parallel implementation of the interior-point method for Semidefinite Programming
Parallel computation for two major bottleneck parts
- **ELEMENTS** ⇒ Computation of Schur complement matrix (SCM)
- **CHOLESKY** ⇒ Cholesky factorization of Schur complement matrix (SCM)

SDPARA could attain high scalability using 16,320 CPU cores on the TSUBAME 2.5 supercomputer and some techniques of processor affinity and memory interleaving when the computation of SCM (ELEMENTS) constituted a bottleneck.

With 4,080 NVIDIA GPUs on the TSUBAME 2.0 & 2.5 supercomputer, our implementation achieved 1.019 PFlops (TSUBAME 2.0) & 1.713 PFlops (TSUBAME 2.5) in double precision for a large-scale problem (CHOLESKY) with over two million constraints.

Parallel Algorithm of Cholesky Factorization
GPU computation, PCI-e communication, and MPI communication are overlapped

1.713 PFLOPS (DP) with 4080 GPUs!!
Phase-field simulation for Dendritic Solidification
[Shimokawabe, Aoki et. al.] Gordon Bell 2011 Winner

- Peta-Scale phase-field simulations can simulate the multiple dendritic growth during solidification required for the evaluation of new materials.
- 2011 ACM Gordon Bell Prize Special Achievements in Scalability and Time-to-Solution

Developing lightweight strengthening material by controlling microstructure
Low-carbon society

Weak scaling on TSUBAME (Single precision)
Mesh size (1GPU+4 CPU cores): 4096 x 162 x 130

- TSUBAME 2.0
  - 2,000 PFlops
  - (4,000 GPUs+16,000 CPU cores)
  - 4,096 x 5,022 x 16,640

- TSUBAME 2.5
  - 3,444 PFlops
  - (3,968 GPUs+15,872 CPU cores)
  - 4,096 x 6,480 x 13,000
The above peta-scale simulations were executed as the TSUBAME Grand Challenge Program, Category A in 2012 fall.

- The LES wind simulation for the area 10km × 10km with 1-m resolution has never been done before in the world.
- We achieved 1.14 PFLOPS using 3968 GPUs on the TSUBAME 2.5 supercomputer.
AMBER pmemd benchmark
Nucleosome = 25,095 atoms

<table>
<thead>
<tr>
<th>Configuration</th>
<th>ns/day</th>
</tr>
</thead>
<tbody>
<tr>
<td>K20X×8</td>
<td>11.39</td>
</tr>
<tr>
<td>K20X×4</td>
<td>6.66</td>
</tr>
<tr>
<td>K20X×2</td>
<td>4.04</td>
</tr>
<tr>
<td>K20X×1</td>
<td>3.11</td>
</tr>
<tr>
<td>M2050×8</td>
<td>3.44</td>
</tr>
<tr>
<td>M2050×4</td>
<td>2.22</td>
</tr>
<tr>
<td>M2050×2</td>
<td>1.85</td>
</tr>
<tr>
<td>M2050×1</td>
<td>0.99</td>
</tr>
<tr>
<td>MPI 4node</td>
<td>0.31</td>
</tr>
<tr>
<td>MPI 2node</td>
<td>0.15</td>
</tr>
<tr>
<td>MPI 1node</td>
<td>0.11</td>
</tr>
</tbody>
</table>

TSUBAME2.0 M2050
TSUBAME2.5 K20X

Dr. Sekijima@Tokyo Tech
GHOSTM: A GPU-Accelerated Homology Search Tool for Metagenomics

Homology search is one of important methods to annotate DNA sequences

Data
- soil metagenomic data (SRR407548)
  150 bp, 100,000 entries
- KEGG GENES amino acid sequences
  4 GB, (May, 2013)

MEGADOCK-GPU

Predicting protein-protein interaction network via protein-protein docking calculations

Docking calculations for 352 pairs

Protein-protein interaction network is very important to understand cell behavior and diseases.

<table>
<thead>
<tr>
<th>Application</th>
<th>TSUBAME2.0 Performance</th>
<th>TSUBAME2.5 Performance</th>
<th>Boost Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top500/Linpack (PFlops)</td>
<td>1.192</td>
<td>2.843</td>
<td>2.39</td>
</tr>
<tr>
<td>Green500/Linpack (GFlops/W)</td>
<td>0.958</td>
<td>&gt; 2.400</td>
<td>&gt; 2.50</td>
</tr>
<tr>
<td>Semi-Definite Programming Nonlinear Optimization (PFlops)</td>
<td>1.019</td>
<td>1.713</td>
<td>1.68</td>
</tr>
<tr>
<td>Gordon Bell Dandrite Stencil (PFlops)</td>
<td>2.000</td>
<td>3.444</td>
<td>1.72</td>
</tr>
<tr>
<td>LBM LES Whole City Airflow (PFlops)</td>
<td>0.600</td>
<td>1.142</td>
<td>1.90</td>
</tr>
<tr>
<td>Amber 12 pmemd 4 nodes 8 GPUs (nsec/day)</td>
<td>3.44</td>
<td>11.39</td>
<td>3.31</td>
</tr>
<tr>
<td>GHOSTM Genome Homology Search (Sec)</td>
<td>19361</td>
<td>10785</td>
<td>1.80</td>
</tr>
<tr>
<td>MEGADOC Protein Docking (vs. 1CPU core)</td>
<td>37.11</td>
<td>83.49</td>
<td>2.25</td>
</tr>
</tbody>
</table>
Graph500 “Big Data” Benchmark

Kronecker graph BSP Problem

\[ \arg \max_{\Theta} P(\Theta) \]

November 15, 2010

Graph 500 Takes Aim at a New Kind of HPC

Richard Murphy (Sandia NL => Micron)

“I expect that this ranking may at times look very different from the TOP500 list. Cloud architectures will almost certainly dominate a major chunk of part of the list.”

The 4th Graph500 List (Jun2012) TSUBAME #4 w/GPUs

Toyotaro Suzumura, Koji Ueno, Tokyo Institute of Technology

Reality: Top500 Supercomputers Dominate

No Cloud IDCs at all

TSUBAME2.0 #3(Nov.2011) #4(Jun.2012)
3500 Fiber Cables > 100Km w/DFB Silicon Photonics
End-to-End 7.5GB/s, > 2us Non-Blocking 220Tbps Bisection
Supercomputer Tokyo Tech. Tsubame 2.0 #4 Top500 (2010)

- Advanced Silicon Photonics 40G single CMOS Die
  1490nm DFB 100km Fiber

- ~1500 nodes compute & storage
  Full Bisection Multi-Rail Optical Network
  Injection 80GBps/Node
  Bisection 220Terabps


- 8 zones, Total 5600 nodes,
  Injection 1GBps/Node
  Bisection 160Gigabps

- Juniper EX4200
  Zone (700 nodes)
- Juniper EX4200
  Zone (700 nodes)
- Juniper MX480
  2 zone switches (Virtual Chassis)
  10GbE

- LACP
- 10GbE
- Juniper EX8208
- Juniper EX8208
- Juniper EX4200
  Zone (700 nodes)
- Juniper EX4200
  Zone (700 nodes)
But what does “220Tbps” mean?

### Global IP Traffic, 2011-2016 (Source Cisco)

<table>
<thead>
<tr>
<th>By Type (PB per Month / Average Bitrate in Tbps)</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
<th>2015</th>
<th>2016</th>
<th>CAGR 2011-2016</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fixed Internet</strong></td>
<td>23,288</td>
<td>32,990</td>
<td>40,587</td>
<td>50,888</td>
<td>64,349</td>
<td>81,347</td>
<td>28%</td>
</tr>
<tr>
<td></td>
<td>71.9</td>
<td>101.8</td>
<td>125.3</td>
<td>157.1</td>
<td>198.6</td>
<td>251.1</td>
<td></td>
</tr>
<tr>
<td><strong>Managed IP</strong></td>
<td>6,849</td>
<td>9,199</td>
<td>11,846</td>
<td>13,925</td>
<td>16,085</td>
<td>18,131</td>
<td>21%</td>
</tr>
<tr>
<td></td>
<td>21.1</td>
<td>28.4</td>
<td>36.6</td>
<td>43.0</td>
<td>49.6</td>
<td>56.0</td>
<td></td>
</tr>
<tr>
<td><strong>Mobile data</strong></td>
<td>597</td>
<td>1,252</td>
<td>2,379</td>
<td>4,215</td>
<td>6,896</td>
<td>10,804</td>
<td>78%</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>3.9</td>
<td>7.3</td>
<td>13.0</td>
<td>21.3</td>
<td>33.3</td>
<td></td>
</tr>
<tr>
<td><strong>Total IP traffic</strong></td>
<td>30,734</td>
<td>43,441</td>
<td>54,812</td>
<td>69,028</td>
<td>87,331</td>
<td>110,282</td>
<td>29%</td>
</tr>
<tr>
<td></td>
<td>94.9</td>
<td>134.1</td>
<td>169.2</td>
<td>213.0</td>
<td>269.5</td>
<td>340.4</td>
<td></td>
</tr>
</tbody>
</table>

**TSUBAME2.0 Network has TWICE the capacity of the Global Internet, being used by 2.1 Billion users**
Global Server Shipments are Flat – ~40%
Capacity Growth Rate (~30% for non-HPC)

Service rate proportional to North-South bandwidth
IDC Capacity Growth CAGR thus ~30%
“Convergence” of Supercomputing and Big Data with supercomputing leadership

HPC: x1000 in 10 years

CAGR ~= 100%

IDC: x30 in 10 years
Server unit sales flat
(replacement demand)

CAGR ~= 30-40%

Attendees:
US: 25
Europe: 11
Japan 9

Next meeting
Fukuoka, Japan
Feb. 27-28
Adjacent Big Data Workshop
Feb. 26

Charleston, South Carolina, USA, April 30- May 1

Exec Committee
Pete Beckman
Jean-Yves Berthou
Jack Dongarra
Yutaka Ishikawa
Satoshi Matsuoka
Philippe Ricoux

http://www.exascale.org/bdec/
TSUBAME Evolution

Graph 500
No. 3 (2011)

K-Computer
10.5PF
1TB/s

100PF
10PF
1PF

100TF
10TF

25-30PF
2.5PF

5.7PF
3.0PF

1.1
1.2
2.0
2.5

49.5TF
109.7TF
163.2TF
2287.6TF

2007
2009
2011
2013
2015H2

No.7
No.9
No.14
No.4
No.5
No.5
14

No.500 line in the world

Phase 1
Fast I/O
250TB
300GB/s
30PB/Day

Phase 2
Fast I/O
5~10PB
10TB/s
1ExaB/Day

Awards
Focused Research Towards Tsubame 3.0 and Beyond towards Exa

- **Green Computing**: Ultra Power Efficient HPC
- **High Radix Bisection Networks**: HW, Topology, Routing Algorithms, Placement...
- **Fault Tolerance**: Group-based Hierarchical Checkpointing, Fault Prediction, Hybrid Algorithms
- **Scientific “Extreme” Big Data**: Ultra Fast I/O, Hadoop Acceleration, Large Graphs => Convergence
- **New memory systems**: Pushing the envelops of low Power vs. Capacity vs. BW, exploit the deep hierarchy with new algorithms to decrease Bytes/Flops
- **Post Petascale Programming**: OpenACC and other many-core programming substrates, Task Parallel
- **Scalable Algorithms for Many Core**: Apps/System/HW Co-Design
“Software Technology that Deals with Deeper Memory Hierarchy in Post-petascale Era”

2012-2017, PI: Toshio Endo, Tokyo Tech

Growing “Memory wall” will be an obstacle to larger and fast simulations in post-petascale era

- Deeper memory hierarchy and locality improvement are keys
- Goals: ~100PB/s and ~100PB simulations on Exaflops system

Towards deeper hierarchy

Locality Improvement of Stencil Computations

In dev mem Larger than dev mem

Towards deeper hierarchy

Trade off

Capacity (GB)

Bandwidth (GB/s)

100000

10000

1000

100

10

105.63

110.86

116.25

104.58

109.39

114.46

105.93

98.66

89.81

75.16

104.95

104.13

92.06

87.18

90.53

67.64

57.96

50.88

41.62

67.45

63.3

63.37

59.26

57.57

47.42

41.69

36.34

29.88

29.26

4.49

3.85

3.99

2.81

4.63

4.11

4.13

100000

10000

1000

100

Capacity (GB)

Bandwidth (GB/s)

size of each dimension in single precision (float) format.
JST CREST “System Software for Post Petascale Data Intensive Science” (FY2011-15)

<table>
<thead>
<tr>
<th>Co-PI</th>
<th>Institute</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Osamu Tatebe</td>
<td>University of Tsukuba</td>
<td>Project Leader</td>
</tr>
<tr>
<td>Yoshihiro Oyama</td>
<td>University of Electro-Communications</td>
<td></td>
</tr>
</tbody>
</table>

- **Objective**
  - R&D based on **scale-out file system architecture**
  - Target snapshot, 100 TB/s

- **Research topics**
  - Scale-out distributed file system
    - Scale out to $O(10K)$ I/O servers by utilizing access locality
    - Metadata server clustering to scale the performance out
  - Compute node OS
    - Kernel driver, process scheduling, client caching, operation offload
  - Runtime for Data-Intensive Computing
    - Efficient runtime of workflow execution, MapReduce, and MPI-IO for the scale-out distributed file system
JST CREST: Advanced Computing and Optimization Infrastructure for Extremely Large-Scale Graphs on Post Peta-Scale Supercomputers

- Innovative Algorithms and implementations
  - Optimization, Searching, Clustering, Network flow, etc.
  - Extreme Big Graph Data for emerging applications
    - $2^{30} \sim 2^{42}$ nodes and $2^{40} \sim 2^{46}$ edges
    - Over 1M threads are required for real-time analysis
  - Many applications on post peta-scale supercomputers
    - Analyzing massive cyber security and social networks
    - Optimizing smart grid networks
    - Health care and medical science
    - Understanding complex life system

Example: Symbolic Network
- **Human Brain Project** [http://www.humanbrainproject.eu/](http://www.humanbrainproject.eu/)
- Understanding the human brain is one of the greatest challenges facing 21st century science
- **89 billion neurons** (nodes)
- **1 trillion connections** (edges)
- Over $10^{17}$ bytes memory (storage) and $10^{18}$ Flops for brain simulator
Human Brain Project

Symbolic Network

USA Road Network

Graph500 (Toy)
Graph500 (Mini)
Graph500 (Small)
Graph500 (Medium)
Graph500 (Large)
Graph500 (Huge)

K computer: 65536 nodes
Graph500: 5524 GTEPS

Android tablet
Tegra3 1.7GHz: 1GB RAM
0.15 GTEPS: 64.12 MTEPS/W
Our achievements (Super computer) : **Graph500**

<table>
<thead>
<tr>
<th>Rank</th>
<th>1st 2010/11</th>
<th>2nd 2011/06</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SCALE36 / 7.0G / 8192 node</td>
<td>SCALE38 / 18.47 G / 32768 node</td>
</tr>
<tr>
<td>2</td>
<td>SCALE32 / 5.6G / 9544 node</td>
<td>SCALE38 / 18.36 G / 32768 node</td>
</tr>
<tr>
<td>3</td>
<td>SCALE29 / 1.3G / 128 node</td>
<td>SCALE37 / 43.38 G / 4096 node</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rank</th>
<th>3rd 2011/11</th>
<th>4th 2012/06</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SCALE32 / 253G / 4096 node</td>
<td>SCALE38 / 3541G / 32768 node</td>
</tr>
<tr>
<td>2</td>
<td>SCALE37 / 113G / 1800 node</td>
<td>SCALE35 / 508G / 1024 node</td>
</tr>
<tr>
<td>3</td>
<td>SCALE37 / 103G / 4096 node</td>
<td>SCALE38 / 358G / 4800 node</td>
</tr>
<tr>
<td>4</td>
<td>SCALE36 / 100G / 1366 node</td>
<td>SCALE35 / 317G / 1366 node</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rank</th>
<th>5th 2012/11</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SCALE40 / 15363G / 65536 node</td>
</tr>
<tr>
<td>2</td>
<td>SCALE39 / 10461G / 32768 node</td>
</tr>
<tr>
<td>3</td>
<td>SCALE38 / 5848G / 15384 node</td>
</tr>
<tr>
<td>4</td>
<td>SCALE40 / 5524G / 65536 node</td>
</tr>
</tbody>
</table>
Twitter network (Application of Graph500 Benchmark)

Follow-ship network 2009

User i

(i, j)-edge

User j

41 million vertices and 2.47 billion edges

Our NUMA-optimized BFS on 4-way Xeon system

69 ms / BFS

⇒ 21.28 GTEPS

Six-degrees of separation

Frontier size in BFS

with source as User 21,804,357

<table>
<thead>
<tr>
<th>Lv</th>
<th>Frontier size</th>
<th>Freq. (%)</th>
<th>Cum. Freq. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>6,188</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>3</td>
<td>510,515</td>
<td>1.23</td>
<td>1.24</td>
</tr>
<tr>
<td>4</td>
<td>29,526,508</td>
<td>70.89</td>
<td>72.13</td>
</tr>
<tr>
<td>5</td>
<td>11,314,238</td>
<td>27.16</td>
<td>99.29</td>
</tr>
<tr>
<td>6</td>
<td>282,456</td>
<td>0.68</td>
<td>99.97</td>
</tr>
<tr>
<td>7</td>
<td>11536</td>
<td>0.03</td>
<td>100.00</td>
</tr>
<tr>
<td>8</td>
<td>673</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>9</td>
<td>68</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>10</td>
<td>19</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>41,652,230</td>
<td>100.00</td>
<td>-</td>
</tr>
</tbody>
</table>
Extreme Big Data (EBD)
Next Generation Big Data Infrastructure Technologies Towards Yottabyte/Year

Principal Investigator
Satoshi Matsuoka

Global Scientific Information and Computing Center
Tokyo Institute of Technology
Extreme Big Data not just traditional HPC!!!

--- Analysis of required system properties ---

[Slide courtesy Alok Choudhary, Northeastern U]
EBE Research Scheme

Future Non-Silo Extreme Big Data Apps

Large Scale Metagenomics

Ultra Large Scale Graphs and Social Infrastructures

Massive Sensors and Data Assimilation in Weather Prediction

Co-Design

EBD System Software incl. EBD Object System

Exascale Big Data HPC

Convergent Architecture (Phases 1~4)

Large Capacity NVM, High-Bisection NW

Cloud IDC

Very low BW & Efficiency

Supercomputers

Compute&Batch-Oriented
Extreme Big Data (EBD) Team
Co-Design EHPC and EDB Apps

• Satoshi Matsuoka (PI), Toshio Yutaka Akiyama, Ken Endo, Hitoshi Sato (Tokyo Tech.) (Tasks 1, 3, 4, 6)

• Osamu Tatebe (Univ. Tsukuba) (Tasks 2, 3)

• Michihiro Koibuchi (NII) (Tasks 1, 2)

• Toyotaro Suzumura (Tokyo Tech. and IBM Lab, 5-2)

• Takemasa Miyoshi (Riken AICS, 5-3)
In most living organisms, genetic instructions used in their development are stored in the long polymeric molecule called DNA. DNA consists of two long polymers of simple units called nucleotides. The four bases found in DNA are adenine (abbreviated A), cytosine (C), guanine (G), and thymine (T).

To decipher the information contained in DNA, we need to determine the order of nucleotides. This task is important for many emerging areas of science and medicine. Modern sequencing techniques split the DNA molecule into pieces (called reads) which are processed separately to increase the sequencing throughput. Reads must be aligned to the reference sequence to determine their position in the molecule. This process is called read alignment.
In most living organisms genetic instructions used in their development are stored in the long polymeric molecule called DNA. DNA consists of two long polymers of simple units called nucleotides. The four bases found in DNA are adenine (abbreviated A), cytosine (C), guanine (G) and thymine (T).
E/0 - I/O
(Many-core I/O)

Preliminary I/O Evaluation on GPU and NVRAM

How to design local storage for next-gen supercomputers?
- Designed a local I/O prototype using 16 mSATA SSDs

• Capacity: 4TB
• Read bandwidth: 8 GB/s

〜 320K IOPS (3 μ sec)

I/O performance of multiple mSATA SSD

〜 7.39 GB/s from 16 mSATA SSDs (Enabled RAID0)

I/O performance from GPU to multiple mSATA SSDs

〜 3.06 GB/s from 8 mSATA SSDs to GPU

Matrix Size [GB]
Target C/R strategies & Storage designs

- **Single-level**
- **Multi-level**

**Synchronous**
- Computation
- Sync ckpt
- Sync ckpt

**Asynchronous**
- Computation
- Background process
- Async ckpt
- Async ckpt

- **Coordinated**
  - P0
  - P1
  - P2
  - P3

- **Uncoordinated**
  - P0
  - P1
  - P2
  - P3

**Flat buffer**
- Compute node 1
- Compute node 2
- Compute node 3
- Compute node 4
- SSD 1
- SSD 2
- SSD 3
- SSD 4
- PFS (Parallel file system)

**Burst buffer**
- Compute node 1
- Compute node 2
- Compute node 3
- Compute node 4
- SSD 1
- SSD 2
- SSD 3
- SSD 4
- PFS (Parallel file system)
Multi-level Asynchronous C/R Model

- Compute checkpoint/restart "Efficiency" for C/R strategy comparison
  - **Efficiency**: Fraction of time an application spends only in computation in optimal checkpoint interval

\[ f : (L_{i=1...N}, O_{i=1...N}, R_{i=1...N}) \]

\[
\begin{align*}
\text{Efficiency} &= \frac{\text{ideal runtime}}{\text{expected runtime}} \\
\text{ideal runtime} \text{ : No failure and No checkpoint} \\
\text{expected runtime} \text{ : Computed by the models}
\end{align*}
\]

\[
\begin{align*}
\text{Interval} & : t \\
\text{c}_c & : c \cdot \text{level checkpoint time} \\
\text{r}_c & : c \cdot \text{level recovery time} \\
\lambda_i & : i \cdot \text{level checkpoint time}
\end{align*}
\]

Recursive Structured Storage Model
(Collaboration with DoE LLNL)

- Generalization of storage architectures with "context-free grammar"
  - A tier $i$ hierarchical entity ($H_i$), has a storage ($S_i$) shared by ($m_j$) upper hierarchical entities ($H_{i-1}$)
  - $H_{i=0}$ is a compute node
  - $H_N \{m_1, m_2, \ldots, m_N\}$

**Example**

<table>
<thead>
<tr>
<th>$r_i$</th>
<th>Sequential read throughput from compute nodes ($H_{i=0}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_i$</td>
<td>Sequential write throughput from compute nodes ($H_{i=0}$)</td>
</tr>
<tr>
<td>$m_i$</td>
<td>The number of a upper hierarchical entities ($H_{i-1}$) sharing $S_i$</td>
</tr>
</tbody>
</table>

**Storage Model:** $H_N \{m_1, m_2, \ldots, m_N\}$

$\frac{<\text{# of C/R nodes per } S_i>}{K^*} \leq \frac{<\text{# of } S_i>}{\Pi_{k=i+1}^{N} m_k}$

$K^*$: C/R cluster size
Experimental Setup

1 Compute node

Read: 500 MB/s
Write: 260 MB/s

Node 1

Node 2

Node 1088

Flat buffer system: $H_2 \{1, 1088\}$

Aggregate Read: 544 GB/s
Aggregate Write: 283 GB/s

Read: 10 GB/s
Write: 10 GB/s

Checkpoint size: 5 GB/node
Logging cluster size: 16 nodes *

32 Compute node

Read: 16 GB/s
Write: 8.32 GB/s

Node 1

... Node 32

... Node 1088

Burst buffer system: $H_2 \{32, 34\}$

The system sizes are based on the Coastal cluster at LLNL (88.5TFLOPS)

* Guermouche, A., Ropars, T., Snir, M. and Cappello, F.: HydEE: Failure Containment without Event Logging for Large Scale Send-Deterministic MPI Applications
Efficiency with Increasing Failure Rates and Checkpoint Costs

- Assuming message logging overhead is 0
- The burst buffer system always achieves a higher efficiency
  ⇒ Stores checkpoints on fewer nodes

- All systems work equally well up to x10 ⇒ TSUBAME4.0 (2020) can go exascale

- Uncoordinated checkpointing: 70% efficiency on systems two orders of magnitude larger (if logging overhead is 0)
  ⇒ Partial restart exploit the bandwidth of both burst buffers and the PFS
Phase 3 Scaling up to Petabyte/s I/O EBD 2017-18
DRAM+Flash(+Processor) 100 ExaB/Day, 30 ZetaB/Year

Rack
4 cabinets/64 nodes
25TB DRAM
786TB Flash
50 TB/s DRAM BW
1.54TB/s Flash BW
1.28TB/s NW BW
384TFLops
30.7KW, $1 mil

IDC/SC
650 Racks (~ES)
41,600 nodes
16PB DRAM
511PB Flash
25.6PB/s DRAM BW
1PB/s Flash BW
(x1000 K-comp HDD)
250PFlops DFP
500PFlops SFP
830TB/s NW BW
20MW, $700 million

25.6GB/s DDR4 channels
Embedded 100Gbps (~10GB/s)
Memor y switch
Intel
Post-Skymont, NVIDIA Post Volta, Fujitsu fx-XX, etc.

25.6GB/s x 3~4 =80~100GB/s (DDR4-3200)
4~6 channels=>320~600GB/s
(12~24 DIMMS per socket)
4.8 Teraflops 10W, $500?
Phase 4: 2019-20 DRAM+NVM+CPU with 3D/2.5D Die Stacking - The Ultimate Convergence of BD and EC -

2Tbps HBM
4~6HBM Channels
1.5TB/s DRAM & NVM BW

30PB/s I/O BW Possible
1 Yottabyte / Year
A scalable MapReduce-based large scale graph processing algorithm using multi-GPU [CCGrid 2013]

How to utilize multi-GPU for large-scale graph processing?

- Implemented a graph processing algorithm on multi-GPU
- Confirmed speedup on multi-GPU using PageRank application

Performance evaluation on TSUBAME 2.0 (SCALE 27, 128 nodes)

![Graph processing performance graph]

186.6x Speedup over Hadoop
1. Introduction

- Large scale graph processing in various domains: DRAM resources has increased

- Spread of Flash Devices
  - Prof: Price per bit, Energy consumption
  - Cons: Latency, Throughput

Using NVRAMs for large scale graph processing has possibilities of minimum performance degradation

2. Hybrid-BFS

Switch two approaches

Top-down

\[ n_{\text{frontier}} < \frac{n_{\text{all}}}{\beta} \]

- \( n_{\text{frontier}} \): number of frontiers
- \( n_{\text{all}} \): total number of vertices
- \( \beta \): parameter

Bottom-up

\[ n_{\text{frontier}} > \frac{n_{\text{all}}}{\alpha} \]

- Switch two approaches

3. Proposal

1. Offload small accesses data
2. BFS with reading data from NVRAM

4. Evaluation (Offload Top-down Graph: we could reduce half the size of DRAM [128GB -> 64 GB] at Scale 27)

- GTEPS

Switching Parameter

- \( \beta = 10\alpha \)
- \( \beta = 0.1\alpha \)
- \( \alpha = 1.E+04 \)
- \( \alpha = 1.E+05 \)
- \( \alpha = 1.E+06 \)
- \( \alpha = 1.E+07 \)

- 4.1GTEPS (79.4%)
- 5.2GTEPS
- 2.8GTEPS (52.9%)

- DRAM Only
- DRAM+ioDrive2
- DRAM+Intel SSD
5. Current Work

In Bottom-up approach, all un-visited vertex have to do is find a edge which is connected to frontier’s vertex.

A lot of edges are not accessed

Each vertex allocate only a few edges to DRAM

Vertices ID

<table>
<thead>
<tr>
<th>DRAM</th>
<th>NVRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_0</td>
<td></td>
</tr>
<tr>
<td>V_1</td>
<td></td>
</tr>
<tr>
<td>V_2</td>
<td></td>
</tr>
</tbody>
</table>

: outgoing edges form V_n

Simulation : Reduce Bottom-up Graph(BG), Scale 27

Max Number of Edges Which are Allocated in DRAM

Better

<table>
<thead>
<tr>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.0%</td>
<td>30.0%</td>
<td>20.0%</td>
<td>10.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

EBD Algorithm Kernels

6. Related Work and Summary

- Pearce, et al. : 1 TB DRAM and 12 TB NVRAM(Fusion-io ioDrive)
  52 MTEPS [Scale 36 : 69G vertices, 1100G edges]

- We could reduce half the size of DRAM with 20.6% performance degradation (4.1 GTEPS)  [ Scale 27 : 130M vertices, 2.1G edges ]

Roger Pearce, Maya Gokhale, Nancy M. Amato, "Scaling Techniques for Massive Scale-Free Graphs in Distributed (External) Memory" Parallel and Distributed Processing Symposium, International, 2013 IEEE 27th International Symposium on Parallel and Distributed Processing
High Performance Sorting

Fast algorithms:

- Distribution vs Comparison-based
- MSD radix sort
  - variable length / long keys
  - high efficiency on small alphabets

Efficient implementation

GPUs are good at counting numbers

Computational Genomics

(A,C,G,T)

Scalability

N log(N) classical sorts
(quick, merge etc)

LSD radix sort
(THRUST)

short length / fixed-length keys

Integer sorts

Apple apricot banana kiwi
don’t have to examine all characters

Bitonic sort

Comparison of keys

Map-Reduce
Hadoop easy to use but not that efficient

Hybrid approaches/

Best to be found

Good for GPU nodes

Balancing IO / computation

Algorithm Kernels on EBD

Efficient implementation

Scalability
R&D of EDB Distributed Object Store
(co-PI: Osamu Tatebe, U-Tsukuba)

- Key design issues for Scaled-out IOPS and I/O bandwidth
  - Scalable distributed MDS (1M IOPS)
  - High Performance local object store
  - Efficient parallel access (100 TB/s) and parallel query
R&D of EDB Distributed Object Store

• Early distributed MDS design achieves 270K IOPS using 15 MDSs. [not published yet]

<table>
<thead>
<tr>
<th></th>
<th>Ours</th>
<th>GIGA+</th>
<th>skyFS</th>
<th>Lustre</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOPS</td>
<td>270K</td>
<td>98K</td>
<td>100K</td>
<td>80K</td>
</tr>
<tr>
<td>#servers (#cores)</td>
<td>15 (240)</td>
<td>32 (256)</td>
<td>32 (512)</td>
<td>1 (16)</td>
</tr>
</tbody>
</table>

Distributed MDS Performance

• Local Object Storage design achieves 200K IOPS using FusionIO IOdrive. [SWoPP 2013]
This year’s goal

• Conceptual Design of object store
  – Distributed metadata server for O(100K) clients
  – Local object store for NVRAM/Flash
  – Parallel query to maximize data locality
**EBD Interconnect (Koibuchi Group)**

Typical Data Centers
- Poor scalability
- 1GbE + 10GbE
- TCP/IP basis

Our current technology:
- Cabling Layout[HPCA13]  Virtual routing method[IPDPS09]

- Low-latency write/read ~10 μs for 4KB
- EBD interconnect
  - Low-jitter topology w/ random shortcuts
  - TCP/IP bypassing direct comm. to flash

- Extreme Big Data Flow

K Computer
Supercomputers
- Dedicated to neighboring and uniform access
EBD Interconnects (Cont’d)

Layout-conscious Random Topology and routing

Topology Graph Analysis

Simulation results under non-uniform shuffle access

Average path length [hops]

Switch degree ≈ Number of shortcuts

1,024-node network

Switch degree

Rand shortcuts reduce path hops

Effect of random shortcuts

Torus/mesh have low throughput

(before) → (after) EBD interconnect

- Torus/Fat tree → Random shortcut topology
- TCP/IP comm. to HDD → TCP/IP bypass direct comm. to flash
- 1G-10/20Gbps tech. → Modern InfiniBand tech.
API co-design for complicated I/O requirements
(co-PI: Yutaka Akiyama, Tokyo Tech)

300% increase per year

**O(n)** Meas. data

**O(m)** Reference Database

**O(m n)** calculation

Correlation, Similarity search

Important issues:
1) Allocation of big Measured data
2) Allocation of big References (DB)

Metagenome sciences

Simple batch of **BLASTX** software

0.18 M Reads / hour
144core Xeon Cluster (2010)

3000-fold speed-up

**GHOST-MP**
OpenMP / MPI load-balancing data dispatcher

572.8 M Reads / hour
82944node K-computer (2012)
1) Novel APIs for supporting abstraction of I/O

**New Idea: “EBD bag” (a kind of large-scale Key-Value Store)**

Because most of results are **write-only**, and **independent** in time order.

It **virtually enables completely distributed I/O programming** (B) efficiently.

2) System Evaluation through real big applications

Ultra-scale metagenome analysis, cancer genome, compound screening, etc.
API co-design for complicated I/O requirements

Plan for H25 (FY2013)

1. Requirement Analysis and Schematic Design for New APIs
   ✓ EBD vs. EBD collective analysis procedures
   ✓ Proposal of the “EBD bag” function

2. Preparation for evaluation through real big applications
   ✓ Ultra-scale Metagenome analysis: data collection and system prototyping (ex. human oral microbiome)
   ✓ Cancer genome analysis
   ✓ Estimation of near-future I/O requirements in related fields (genomics, proteomics, drug design, etc.)
Suzumura Group
EBD Driven Planetary-Scale Social Analytics Infrastructure

EBD-Driven Social Simulation

“Billion-Scale” and 10-Fold Real Time Discrete Event Simulation

- 7 billion human beings on the planet with 3 billion-level road network
- Log data generation speed = 700 Tbps
- Total log size per 1 simulation = 2.2 PB

EBD-Driven Social Analytics

Large-Scale Graph Analysis

- Grand Challenge Problem Size: $2^{42}$ vertices
  (4.4 Trillion Vertices, 1.1 PB Memory)
- Centrality/BC/BFS/RWR/Clustering, etc

Supercomputer with
- 25 PFLOPS, 10PB (DRAM) and 511 PB (Flash), 1 Petabit/s (Comm.)

Co-design

Data Source

10 Tbps
(Streaming Data including Satellite Image)

EBD Object Store
A Study on Scalable Architecture and Optimization Methods for Billion-scale Social Simulation

• Motivation & Goal for 2013 and 2014: Our previous design (ABCA) cannot cope with billion-scale simulation in real-time due to tremendous amount of data and I/O, so this study is to propose the best architecture that can deal with real-time billion-scale social simulation on the future hardware designed for extremely big data processing
• Study the performance characteristics of the agent-based social simulation implementations of each candidate architecture and optimization methods
  – We started investigating from billion-scale traffic simulation
• Current status: we have completed the implementations for the first two architectures and evaluated them in million-scale simulation.
• Plan by the end of this year: Complete implementations of three candidate architectures and evaluate them in billion-scale simulation
• Future: we plan to make the framework more flexible to support more complicated social simulation

- Tokyo Map (~160K cross points, ~230K roads, 46K agents)
- Tsubame S queue machine
Fail-Safe EBD Workflow and Geometrical Search in Big Data Assimilation (co-PI: Takemasa Miyoshi, Riken AICS)

Weather Observation data keep flowing-in every 30s. In case of hardware failure, it is difficult to catch up once delayed.

4D-LETKF enables processing multiple steps at one time. Observations at multiple times are treated simultaneously.

Highly reliable system enabling to catch up in case of delay.

\[ \tilde{x}_{a}(t_{n+1}) = \tilde{x}_{a}(t_{n}) + X_{a}(t_{n})\tilde{w}_{a}(t_{n}) \]
\[ \bar{x}_{a}(t_{n+1}) = X_{a}(t_{n})\tilde{w}_{a}(t_{n}) \]
\[ \tilde{w}_{a} = \tilde{P}_{a}Y_{a}^{T}R^{-1}(y - H(x)) \]
\[ \tilde{w}_{a} = [(K - 1)\tilde{P}_{a}]^{T} \]
Next-generation Data Assimilation System
(Proposed simultaneously to Prof. Tanaka’s CREST)

Phased Array Radar
1GB/30sec/2 radars

A-1. Quality Control
A-2. Data Processing

Himawari
500MB/2.5min

B-1. Quality Control
B-2. Data Processing

Observation data keep flowing-in every 30 sec.

In case of hardware failure

It is hard to catch up once it gets delayed

4D-LETKF enables processing multiple steps at one time

Highly reliable system enabling to catch up in case of delay

Repeat every 30 sec.

4-dimensional Ensemble Kalman Filter
4D-LETKF

\[
\hat{x}_n(t_{n-1}) = \hat{x}_n(t_{n-1}) + X_n(t_{n-1}) W_n(t_n)
\]

\[
\hat{x}_n(t_{n-1}) = X_n(t_{n-1}) W_n(t_n)
\]

\[
W_n = \hat{P}_n Y_n H^{-1}(y - H(x));
\]

\[
W_n = [(K - 1) \hat{P}_n]_{i,j}
\]
LETKF (Local Ensemble Transform Kalman Filter) includes geographical search of nearby observations around each grid point on Earth.

Search $O(10^3)$ nearby observations out of total $O(10^6)$ at each of $O(10^7)$ grid points.

About half of the total LETKF computer time.

**Co-design of Hardware and Software**

Optimizing the spatial search algorithm suitable for the converged EBD architecture.
International Collaborators and Potential Industries

Alok Chaudhary
Professor, Northwestern U
Big data performance and benchmarking

Rick Stevens
Associate Laboratory Director, Argonne National Laboratory
Convergence Architecture

Robert Ross
Math. and Computing Sciences, Argonne National Laboratory
Distributed Big Data Objects

Gabriel Antoniu
Scientific leader of the KerData research team at INRIA Rennes
Distributed Filesystems

Graph and Big Data Benchmarking