HPC-GPU: Large-Scale GPU Accelerated Windows HPC Clusters and its Application to Advanced Bioinformatics and Structural Proteomics (and Climate/Environment)

Satoshi Matsuoka, Professor/Dr.Sci.
&
Yutaka Akiyama, Professor
with
Toshio Endo, Fumikazu Konishi, Akira Nukada, Naoya Maruyama...

Tokyo Inst. Technology
The TSUBAME 1.0 “Supercomputing Grid Cluster”
April 2006 at Tokyo Tech
80 TFlops, 1400 Users, 200 Industry Users

Voltaire ISR9288 Infiniband 10Gbps x2 (DDR next ver.)
~1310+50 Ports
~13.5Terabits/s (3Tbits bisection)
10Gbps + External Network

Sun Galaxy 4 (Opteron Dual core 8-socket)
10480core/655Nodes
21.4Terabytes
50.4TeraFlops
OS Linux (SuSE 9, 10)
NAREGI Grid MW

“Fastest Supercomputer in Asia” 7th on the 27th Top500@38.18TF

Unified IB network

Storage
1.0 Petabyte (Sun “Thumper”)
0.1 Petabyte (NEC iStore)
Lustre FS, NFS, WebDAV (over IP)
50GB/s aggregate I/O BW

ClearSpeed CSX600
SIMD accelerator
360 boards,
35TeraFlops (Current)
Titech TSUBAME
~76 racks
350m² floor area
1.2 MW (peak)
TSUBAME Upgrades Towards Petaflops

Sustained Acceleration

- Earth Simulator 40TF (2002)
- Titech Campus Grid Clusters
- BlueGene/L 360TF (2005)
- TSUBAME 1.1 103 TF (2002)
- TSUBAME 1.2 ~900TF SFP (2008-2H)
- TSUBAME 1.2 170 TF DFP (2008-2H)
- U-Tokyo, Kyoto-U, Tsukuba T2K 61-140TF (2008)
- TSUBAME 2.0 > 1PF (2010)
- TSUBAME 2.1 > 3PF (2011-2H)
- TSUBAME 2.1 > 10TF (2011~12?)
- Japanese NLP >10PF (2012-1Q)
- US >10P (2011~12?)

- LANL Roadrunner (Jun 2008)
- TSUBAME 85TF 1.1 PB (2006)
# Biggest Problem is Power...

<table>
<thead>
<tr>
<th>Machine</th>
<th>CPU Cores</th>
<th>Watts (W)</th>
<th>Peak GFLOPS</th>
<th>Peak MFLOPS/Watt</th>
<th>Watts/CPU Core</th>
<th>Ratio c.f. TSUBAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSUBAME (Opteron)</td>
<td>10480</td>
<td>800,000</td>
<td>50,400</td>
<td>63.00</td>
<td>76.34</td>
<td></td>
</tr>
<tr>
<td>TSUBAME 2006 (w/360CSs)</td>
<td>11,200</td>
<td>810,000</td>
<td>79,430</td>
<td>98.06</td>
<td>72.32</td>
<td></td>
</tr>
<tr>
<td>TSUBAME 2007 (w/648CSs)</td>
<td>11,776</td>
<td>820,000</td>
<td>102,200</td>
<td>124.63</td>
<td>69.63</td>
<td>1.00</td>
</tr>
<tr>
<td>Earth Simulator</td>
<td>5120</td>
<td>6,000,000</td>
<td>40,000</td>
<td>6.67</td>
<td>1171.88</td>
<td>0.05</td>
</tr>
<tr>
<td>ASCI Purple (LLNL)</td>
<td>12240</td>
<td>6,000,000</td>
<td>77,824</td>
<td>12.97</td>
<td>490.20</td>
<td>0.10</td>
</tr>
<tr>
<td>AIST Supercluster (Opteron)</td>
<td>3188</td>
<td>522,240</td>
<td>14400</td>
<td>27.57</td>
<td>163.81</td>
<td>0.22</td>
</tr>
<tr>
<td>LLNL BG/L (rack)</td>
<td>2048</td>
<td>25,000</td>
<td>5734.4</td>
<td>229.38</td>
<td>12.21</td>
<td>1.84</td>
</tr>
<tr>
<td>Next Gen BG/P (rack)</td>
<td>4096</td>
<td>30,000</td>
<td>16384</td>
<td>546.13</td>
<td>7.32</td>
<td>4.38</td>
</tr>
<tr>
<td>TSUBAME 2.0 (2010Q3/4)</td>
<td>160,000</td>
<td>810,000</td>
<td>1,024,000</td>
<td>1264.20</td>
<td>5.06</td>
<td>10.14</td>
</tr>
</tbody>
</table>

TSUBAME 2.0  x24 improvement in 4.5 years…? ➔ ~ x1000 over 10 years
NVIDIA CUDA Architecture

240 SPs / Chip (Tesla G200 GPU)
Great Relative Power/Performance Possibilities for Scientific Codes
GPUs (Tesla, FireStream, Larrabee, ClearSpeed) as Extreme Many-Core and Solution to This Problem

nVidia Tesla T10: 65nm, 600m2, 1.4 bil Tr
1.08TF SFP

"Massive FMA FPUs"
65~55nm(2008)
=> 15 nm (2016)
x20 transitors (30 bil)
5000 Cores
20TF FMA SFP
10TF FMA DFP

"Powerful Scalar"

x86
16 cores
2.4Ghz
80GFlops

240 Cores
1.5Ghz
1.08TFlops SFP
90GFlops DFP

20GBytes/s
102 GBytes/s
PCI-e

Thread Processor Cluster (TPC)
Thread Processor Array (TPA)
Thread Processor
Overview of HPC-GPGPU Project (sponsored by Microsoft Research)

Advanced Bioinformatics/Proteomics

Bioinformatics Acceleration
- e.g., 3-D All-to-All Protein Docking

GPGPU-CPU
- Hybrid Massively Parallel
- “Adaptive” Solvers + GPGPU FFT and other Acceleration Kernels
- Improving GPGPU Programmability w/ Library/Languages e.g. MS Accelerator
- High Dependability w/ large-scale GPGPU Cluster
- CPU-GPU Heterogeneity
- Low Energy Computing w/GPUs

Need x1000 acceleration over standard PCs

Towards Next Gen Petascale Personal Clusters and Desksides

Prototype Cluster
- 32-64 nodes, 50-100 TFlops
Unlike the conventional accelerators, GPUs have high memory bandwidth.

Since latest high-end GPUs support double precision, GPUs also work as commodity vector processors.

The target application area for GPUs is very wide.

Restrictions: Limited non-stream memory access, PCI-express overhead, etc.

How do we utilize them easily?
Prototype HPC-GPGPU Cluster

Current: TSUBAME Production
“Supercomputing Grid Cluster”
@Tokyo Tech 2006-2010

Node Architecture and Specs
- Single Socket, Quad Core x64
- Latest GPGPUs from NVidia/AMD
- Small Memory (4GB DDR2)
- 600W Power
- 1~2 TFlops, $2500/node

Whole Cluster Specs (~50 nodes)
- 50-100 TeraFlops
- 20K-30KW Power
- Massive FFT Engine
- $100,000-$200,000 MSRP

Windows HPC-GPU Cluster

50~100 TFlops Peak, fast FFT

Voltaire ISR9288 Infiniband 10Gbps
x2 (DDR next ver.)
~1310+50 Ports
~13.5 Teraflops/s (3 TFlops peak)
Unified IB network

Sun Galaxy 4 (Opteron Dual core 8-socket)
10480 core/655 Nodes
21.7 Terabytes
50.4 TeraFlops
OS Linux (SuSE 9, 10)
NAREGI Grid MW

Storage
1.5PB+ (Sun “Thumper”)
0.1PB (NEC iStore)
 Lustre FS, NFS, CIF, WebDAV (over IP)
50GB/s aggregate I/O BW

ClearSpeed CSX600
SIMD accelerator
648 boards, 52 TeraFlops (Current)

~100 TFlops
80+ racks
350m2 floor
1.2 MW (peak)
~50 tons
$70~80 mil MSRP
Fastest SC in Japan

x500 C/P improvement
Protein docking is a central issue both in computational biology and current drug design technique in pharmaceutical industry.

It is becoming a new trend to design an “inhibitor” drug compound which controls specified protein-protein interaction as a target.

Protein Docking is important for:
1) Enzymatic reaction
2) Signal Transduction
3) Formulating protein scaffold, etc.
All-to-all 3-D Protein Docking Challenge (by Y. Akiyama, in collaboration w/AIST CBRC and other pharma companies)

1,000 x 1,000 all-to-all docking fitness evaluation will take only 1-2 months (15 deg. pitch) with a 32-node HPC-GPGPU cluster (128 GPGPUs).

cf.

~ 500 years with single CPU (sus. 1+GF)

> 2 years with 1-rack BlueGene/L

Fast PPI candidate screening with FFT

Blue Protein system
CBRC, AIST
(4 rack, 8192 nodes)
Rigid Docking vs. Flexible Docking

Rigid Docking
- Protein is regarded as a rigid body
- Shape complementarity and only simple physicochemical potentials.

No biologist think proteins are rigid.
However,
- Flexible docking is prohibitively expensive due to energy local minima.
- Flexible docking usually needs good initial docking structure.
- Some proteins are almost rigid.

Porcine Pancreatic Trypsin (PDB:1AVX).

Single Rigid docking has only limited validity.
But all-to-all Rigid Docking screening can be a good basis for quick survey of potential pairs, and for flexible docking study.
Calculation flow and Complexity

Calculation for a single protein-protein pair: \( \sim 200 \) Tera ops.

3-D complex convolution \( O(N^3 \log N) \), typically \( N = 256 \)

\( \times \)

Possible rotations \( R = 54,000 \) (6 deg. pitch) \( \sim 200 \) Exa \( (10^{20}) \) ops
Rigid Docking Example

- Implemented 6 clustering methods for Post-Docking analysis
- Developed a confidence level evaluation procedure

Receptor protein (Trypsin) is shown in center as a ribbon diagram. Top 2,000 candidate ligand docking sites are shown by small dots. Final prediction site is shown by a red sphere.

If significant candidate is found, the system outputs predicted docking structure with a confidence level, otherwise reports as “No docking”.

**Trypsin and inhibitor (PDB: 1AVX)**
- prediction
- other clusters
- real

clusters vs. significance
Bandwidth Intensive 3-D FFT on NVIDIA CUDA GPUs [SC08]

By Akira Nukada, Tokyo Tech.

Our 3-D FFT algorithm consists of the following two algorithms to maximize the memory bandwidth:

1. optimized 1-D FFTs for dimension X,
2. multi-row FFT for dimension Y & Z.

The multi-row FFT computes multiple 1-D FFTs simultaneously.

Adapted from vector algorithms, assuming high memory bandwidth.

This algorithm accesses multiple streams, but each of them is successive. Since each thread computes independent set of small FFT, thousands of registers are required. Solution: for 256-point FFT, use two-pass 16-point FFT kernels.
Performance of 1-D FFT

GFLOPS

Note: An earlier sample with 1.3GHz is used for Tesla S1070.
Comparison with CPUs

X3 Faster than FFTW on GPUs

GFLOPS

Phenom 9500
Core2Quad Q6700
Opteron 16core
CELLBE (PS3)
8800GT
8800GTS
8800GTX
GTX280

140 GFLOPS
Performance in Double Precision

The bottleneck is floating-point operation in double precision.

Both GPUs are running at 1.3GHz, but product version of S1070 will come with higher clock improve the performance.

Performance is competitive with that of a single-node vector supercomputer NEC SX-6 (64 GFLOPS peak).

3-D FFT of size $256^3$
Performance of DP 3-D FFT on CUDA and TSUBAME

GFLOPS

GTX 280 1 node 2 node 4 node 8 node 16 node 32 node 64 node

Tokyo Tech TSUBAME

OpenMP ver.

MPI version is used for computation with multiple nodes
The Worst Case:
GPU computes only FFT, and CPU computes all the others.

Ex) simply replacing CPU library by GPU
Ex) data come from I/O devices

We have to transfer data between host and device using PCI-Express bus.

In the best case, the host CPU is used only to control GPUs.

only support PCI-e 1.1
Eliminating the Bandwidth Bottleneck---Entire docking in GPU

- Read geometry data of atoms of Proteins from PDB, preprocess, and transfer both to the GPGPU card.
- Generate 3D Grid data on the card. For ligands rotate by 6 or 15 degrees increments.
- Conduct two forward transforms, one backward transform, and an element-wise multiplication.
- Find the best docking position with statistical clustering.

MAIN LOOP on GPGPU Card
=> almost no data transfer between CPU and GPU
Raccoon: Accelerated WinHPC GPU Prototype Cluster

- 32 compute nodes
- 128 8800GTS GPGPUs
- one head node.
- Gigabit Ethernet network
- Three 40U rack cabinets.
- Linux or Windows HPC 2008
- Visual Studio 2005 SP1
- nVidia CUDA 2.x

- New nodes for GTX280, Infiniband, ...
## Performance Estimation for 3D PPD

### Single Node

<table>
<thead>
<tr>
<th></th>
<th>Power (W)</th>
<th>Peak (GFLOPS)</th>
<th>3D-FFT (GFLOPS)</th>
<th>Docking (GFLOPS)</th>
<th>Nodes per 40 U rack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue Gene/L</td>
<td>20</td>
<td>5.6</td>
<td>-</td>
<td>1.8</td>
<td>1024</td>
</tr>
<tr>
<td>TSUBAME</td>
<td>1000 (est.)</td>
<td>76.8 (DP)</td>
<td>18.8 (DP)</td>
<td>26.7 (DP)</td>
<td>10</td>
</tr>
<tr>
<td>8800 GTS *4</td>
<td>570</td>
<td>1664</td>
<td>256</td>
<td>207</td>
<td>10~13</td>
</tr>
</tbody>
</table>

### System Total

<table>
<thead>
<tr>
<th></th>
<th># of nodes</th>
<th>Power (kW)</th>
<th>Peak (TFLOPS)</th>
<th>Docking (TFLOPS)</th>
<th>MFLOPS/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue Gene/L (Blue Protein @ AIST)</td>
<td>4096 (4racks)</td>
<td>80</td>
<td>22.9</td>
<td>7.0</td>
<td>87.5</td>
</tr>
<tr>
<td>TSUBAME</td>
<td>655 (~70 racks)</td>
<td>~700</td>
<td>50.3 (DP)</td>
<td>17.5 (DP)</td>
<td>25</td>
</tr>
<tr>
<td>8800 GTS</td>
<td>32 (3racks)</td>
<td>18</td>
<td>53.2</td>
<td>6.5</td>
<td>361 (x4 BG!)</td>
</tr>
</tbody>
</table>

Can compute 1000x1000 in 1 month (15 deg.) or 1 year (6 deg.)

*Only CPUs for TSUBAME. DP=double precision.*
DNA Giga Sequencing Using GPUs

- Personalized DNA-based Diagnostics and Medicine => Giga Sequencers
- Proposal: GPUs for giga-size short fragment DNA sequencing, for 50,000,000 fragments
  - Rigorous Smith-Watherman gapped alignment
  - Months on a conventional cluster, just 17 hours on our test cluster

<table>
<thead>
<tr>
<th>Type of GPU</th>
<th># of GPUs</th>
<th>Pruning mode</th>
<th># of Probes</th>
<th>total CPU (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8800GTS</td>
<td>128</td>
<td>off</td>
<td>10,000,000</td>
<td>22.6</td>
</tr>
<tr>
<td>Tesla10</td>
<td>128</td>
<td>off</td>
<td>10,000,000</td>
<td>12.2</td>
</tr>
<tr>
<td>8800GTS</td>
<td>128</td>
<td>on</td>
<td>50,000,000</td>
<td>16.7</td>
</tr>
<tr>
<td>Tesla 10</td>
<td>128</td>
<td>on</td>
<td>50,000,000</td>
<td>8.9</td>
</tr>
</tbody>
</table>
CFD, esp. Climate and Disaster Prevention on GPUs

(Material from Prof. Takayuki Aoki, Tokyo Tech.)

Safety

Nuclear (Cooling)

Weather/Environmental

Civil Engineering
Rayleigh-Taylor Instability

Heavy fluid lays on light fluid and unstable.

Euler equation:

\[ \frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \]

\[
Q = \begin{bmatrix}
\rho \\
\rho u \\
\rho v \\
e
\end{bmatrix}
E = \begin{bmatrix}
\rho u \\
\rho u^2 + p \\
\rho uv \\
eu + pu
\end{bmatrix}
F = \begin{bmatrix}
\rho v \\
\rho uv \\
\rho v^2 + p \\
ev + pv
\end{bmatrix}
\]

42 GFLOPS using GTX280
Two-Stream Instability
in Plasma Physics

Vlasov-Poisson Equation:

\[
\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} - \frac{eE}{m_e} \frac{\partial f}{\partial v} = 0
\]

\[
\frac{\partial^2 \phi}{\partial x^2} = \frac{e(n_e - n_i)}{\varepsilon_0}
\]

\[
\left( E = -\frac{\partial \phi}{\partial x}, \quad n_e = \int f dv \right)
\]

\( f \) : electron distribution function

\( n \) : electron number density

120 GFLOPS using 8800GTS
3-D Computation of Phase Separation

Nanoscience Structure: 158 GFLOPS using GTX280

Cahn-Hilliard equation:

\[
\frac{\partial \psi}{\partial t} = L \nabla^2 \left( \frac{\partial H}{\partial \psi} - C \nabla^2 \psi \right)
\]

\(H\): free energy

256 x 256 x 256
Real-time Tsunami Simulation

Collaboration with ADPC (Asian Disaster Preparedness Center) and Japan Meteorological Agency

Early Warning System:

Data Base ➔ Real-time CFD

high accuracy

Shallow-Water Eq.

Conservative Form:

\[
\frac{\partial h}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} = 0
\]

Assuming hydrostatic balance in the vertical direction,

3D ➔ 2D equation

\[
\frac{\partial hu}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial huv}{\partial y} = -gh \frac{\partial z}{\partial x}
\]

\[
\frac{\partial hv}{\partial t} + \frac{\partial huv}{\partial x} + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) = -gh \frac{\partial z}{\partial y}
\]
Numerical Methods of Tsunami Simulation

- 2-dimensional Problem: Directional-Splitting Fractional Method
- Point Value Comp.: Characteristic-based Method using Multi-moment Interpolation
- Integral Value Comp.: Conservative Semi-Lagrangian CIP + IDO
- Run-up to dry area: thin water layer and artificial viscosities
GPU Performance

Speed Comparison

x-direction : y-direction = 10 : 7

Current Speed-up

GPU : CPU = 62 : 1

GPU – GeForce GTX280 (sp = 240, clock 1.3Ghz)
CPU – Xeon 2.4GHz 6MB Cache Memory
Multi-GPU Estimation

3000km x 3000km (500m mesh)

17280 sec (1 GPU)

32 GPU

540 sec < 10 min

covering Indian Ocean

Colioris Force
Tidal Potential
Wind effect
TSUBAME 1.2 Evolution (Oct. 2008)
The world's first GPU-based SC in the World

Voltaire ISR9288 Infiniband x8
10Gbps x2  ~1310+50 Ports
~13.5Terabits/s
(3Tbits bisection)

10Gbps+External NW
Unified Infiniband network

NEW Deploy:
GCOE TSUBASA
Harpertown-Xeon
90Node 720CPU
8.2TeraFlops

NEW: co-TSUBAME
90Node 720CPU (Low Power)
~7.2TeraFlops

10,000 CPU Cores
300,000 SIMD Cores
> 3 Million Threads
~900TFlops-SFP, ~170TFlops-DFP
80TB/s Mem BW (1/2 ES)

Storage
1.5 Petabyte (Sun x4500 x 60)
0.1Petabyte (NEC iStore)
Lustre FS, NFS, CIF, WebDAV (over IP)
60GB/s aggregate I/O BW

Sun x4600 (16 Opteron Cores)
32~128 GBytes/Node
10480core/655Nodes
21.4TeraBytes
50.4TeraFlops
OS Linux (SuSE 9, 10)
NAREGI Grid MW

ClearSpeed CSX600
SIMD accelerator
360 648 boards,
35 52.2TeraFlops

170 Nvidia Tesla 1070, ~680 Tesla cards
High Performance in Many BW-Intensive Apps
10% power increase over TSUBAME 1.0
NVIDIA’s Launch Partner
680 Unit Tesla Installation…
While TSUBAME in Production Service (!)
TSUBAME 1.2. The most Heterogeneous Supercomputer in the world

- Three node configurations with four different processors → >30,000 cores, ~170TFlops system

**SunFire X4600+ 2 TESLAs + ClearSpeed**
- Opteron 2.4GHz 16 cores
- TESLA S1070 (30cores) 2boards
- ClearSpeed X620 (2cores) 1board
→ 78 cores, 330 Gflops peak

(318nodes)

**SunFire X4600+ClearSpeed**
- Opteron 2.4GHz 16 cores
- ClearSpeed X620 (2cores) 1board
→ 18 cores, 157 Gflops peak

(330nodes)

**SunBlade X6250** (TSUBASA cluster)
- Xeon 2.83GHz 8 cores
→ 8 cores, 90.7 Gflops peak

(90nodes)
TSUBAME in Top500 Ranking w/our Hetero-Linpack

<table>
<thead>
<tr>
<th></th>
<th>Jun 06</th>
<th>Nov 06</th>
<th>Jun 07</th>
<th>Nov 07</th>
<th>Jun 08</th>
<th>Nov 08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed(TF)</td>
<td>38.18</td>
<td>47.38</td>
<td>48.88</td>
<td>56.43</td>
<td>67.70</td>
<td>77.48</td>
</tr>
<tr>
<td>Rank</td>
<td>7</td>
<td>9</td>
<td>14</td>
<td>16</td>
<td>24</td>
<td>29</td>
</tr>
</tbody>
</table>

- Six consecutive improvements (world’s first)
- The 2nd fastest heterogeneous supercomputer in the world (No.1 is RoadRunner)
- Achieved through extremely intricate hetero-algorithmic advances and tuning
Portfolio of our GPU Computing Base Technologies for HPC & eScience

- TSUBAME 1.2 (680 Teslas) & 2.0
- Kernels (FFT, Dense/Sparse Matrix)
- Parallel Algorithms (Large FFT, LINPACK, CG)
- Task & Resource Mgmt (Heterogeneity, Scheduling, BQ Scheduling, etc.)
- Fault Tolerance (ECC, redundant computation, GPU checkpointing)
- Languages (OpenMP on GPU, Accelerator, MP)
- GPU Low Power computing (power modeling, measurement, optimization)

GPU Leadership from research to deployment(!)
Software-Based ECC for GPUs (N. Maruyama)
Possible Collab. w/MSR Vivian Sewelsen and HPC Cluster

GPU computing reliability
No ECCs on GPUs yet => Bit flip errors?
Large scale clusters will be quite problematic

Our CUDA version memtest86+
60 GPUs on Raccoon

Number of GPUs

0 10 20 30 40 50 60

Number of Bit-Flip Errors

Only 7% overhead w/ECC

X50 exec. time over CPUs

Software-based ECC on GPUs
Read : Read ECC data and check
Write : Generate ECC and store alongside data

Global Memory

User Area

Code Area

Data

Write

Read

Write

Read

Code Gen.

Error checking

N body Problem

Elapsed time

CPU

GPU

GPU Parity

GPU ECC

1024 2048 4096 8192 1638

0 500 1000 1500 2000 2500

0 500 1000 1500 2000 2500

0 500 1000 1500 2000 2500

0 500 1000 1500 2000 2500

0 500 1000 1500 2000 2500
Fast Conjugate Gradient Solver on Multi-GPU Systems (Ali Cevahir)

Iterative CG on multi-CUDA GPUs w/mixed precision
Level 1 BLAS => OK for GPUs
Sparse Matrix*Vector => incurs random access NG for GPUs

Previous Methods
CRS Method (Cache-based CPUs)
- Small # of memory accesses
JDS Method (for vector CPUs)
- Continuous memory accesses

GPU-friendly Implementation
1) JDS-like Storage Format
   - Efficient memory access
   - Adjust alignment for coalescing
2) CRS-like computing order
   - Reduce # of memory accesses

Utilize various memory on CUDA GPUs
1) Matrix element and index (accessed only once) => global memory
2) JDS method offset table (accessed several times) => constant memory (cached)
3) Vector element (accessed many times) => texture memory (cached)

14.5GFlops 4 GPUs (nVidia 8800 GTS) vs. 0.54GFlops 4 Core CPU (Phenom 2.5Ghz)
(Sparse Matrix collection from UFlorida)
Large Scale Hybrid CPUs-GPUs 3-D FFT (Y.Ogata)

Intel Skulltrail
- FFTW on Dual QuadCore Xeon E5420
- CUFFT library on Dual GeForce GTX 280

Multi-CPU, multi-GPU => Distribute work based on respective performance models

Problem: perf degradation due to multi-GPU data transfer conflict
Meth1: [Exclusive] Mutually exclude host-device transfer
Meth2: [Delay] delay GPU execution for overlaps

Delay and performance dist.
Dynamic Scheduling of Matrix Kernels on Heterogeneous GPUs (Y. Watanabe)

Rapid GPU Performance Progress and Updates ➔ New GPU add-ons will differ in performance

Dynamic Load distribution to heterogeneous GPUs

**Thesis:** Dynamic task allocation works effectively for load distribution

**Characteristics**

- **Good:** No a-priori info necessary
- **Bad:** GPU control API Overhead
- **Good:** Natural distribution of Host-GPU transfers, less collision

94% efficiency c.f. summation of single GPU performances

Better performance c.f. static allocation
Problem: To our knowledge there are no tools to predict the power consumption of large-scale parallel program in the algorithm design stage.

Approach: BSP model provides parallel framework; BM Model incorporates physical constraints. Extending them with power-relevant characters, and, power model is built, and performance is examined.

Results: The power measurement results of matrix computation agrees with the estimated result. Further refinement to the model is needed.
Incompressible Fluid Application
Power Measurements thru Sleep Insertions
(T. Aoki)

INT MAIN(INT ARGC, CHAR *ARGV[]){
    CUDAGETDEVICECOUNT();
    ＜初期化や初期条件導入＞
    ＜"計算"開始＞
    DO {
        FOR(JM = 0; JM < 4; JM++) {
            NODE_VELOCITY();
            ADVECTION_X2D(U,U);
            ADVECTION_Y2D(U,V);
            ADVECTION_X2D(V,U);
            ADVECTION_Y2D(V,V);
            UPDATE_VARIABLE(U,UN);
            UPDATE_VARIABLE(V,VN);
            DIFFUSION2D(U);
            DIFFUSION2D(V);
            UPDATE_VARIABLE(U,UN);
            UPDATE_VARIABLE(V,VN);
            TIME += DT;
        }
        DIVERGENCE();
        POISSON2D_MG();
        GRAD_P();
    } WHILE(ICNT++ < 99999 && TIME + 0.5*DT < 0.7);
    ＜ここまで＞
    終了処理
}

Elapsed time [s]
Power Efficiency in 3-D FFD (A. Nukada)

<table>
<thead>
<tr>
<th>GPU</th>
<th>Computation</th>
<th>Idle</th>
<th>Power</th>
<th>GFLOPS</th>
<th>GFLOPS/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>RIVA128</td>
<td>On CPU</td>
<td>126 W</td>
<td>140 W</td>
<td>10.3</td>
<td>0.074</td>
</tr>
<tr>
<td>8800 GT</td>
<td>On GPU</td>
<td>180 W</td>
<td>215 W</td>
<td>62.2</td>
<td>0.289</td>
</tr>
<tr>
<td>8800 GTS</td>
<td>On GPU</td>
<td>196 W</td>
<td>238 W</td>
<td>67.2</td>
<td>0.282</td>
</tr>
<tr>
<td>8800 GTX</td>
<td>On GPU</td>
<td>224 W</td>
<td>290 W</td>
<td>84.4</td>
<td>0.291</td>
</tr>
</tbody>
</table>

CUDA GPUs have four times higher power efficiency than CPU in high-performance FFT.

RIVA128 is an old, low-power GPU, to measure pure power consumption of host system (CPU, chipset, memory). The interface is legacy PCI.
Onto TSUBAME 2.0 & 2.1

• TSUBAME 2.0 will be
  - Deployed 1H 2010
  - A Petascale machine w/GPU acceleration
  - Will be VERY GREEN (same energy envelope)
  - Will be a Cloudy supercomputer, virtualization/multi-OS dynamic deployment
  - Will be a Supercomputer for Everyone

• TSUBAME will run Windows HPC (!)
  - Users will have a choice along w/Linux
  - The first Petascale Windows HPC Cluster?
Road to Exascale and Personal Petascale

2012 10PF (Japanese NLP SC etc.)
2019 1ExaFlops (Leadership machines)
TSUBAME 4.0 > 100PF
Desktop ~ = 1PF

1ExF

100PF

10PF

1PF

100TF

10TF

1TF


Japanese NLP >10PF(2012)

TSUBAME 4.0

Personal Peta

2016 TSUBAME becomes deskside