Hundred Million Cores in Commodity---
Why not? (or, will `custom'*finally* prevail?)
or
“The first Exa Machine shall be a Cloud”

Satoshi Matsuoka, Prof., Dr. Sci.
GSIC Center, Tokyo Institute of Technology /
The NAREGI Project, National Institute of Informatics

CCGSC 2008 @ NC, USA Sep. 14-17 2008
Rise of the Commodity Clusters: “The Scenario”

High Performance Commodity Computing
- High Performance x86 CPUs
- Fast Commodity Interconnect
- Cluster Software

Rise and spread of Commodity Clusters and increase in their size

Widespread Use of Clusters:
Small to very large (e.g. TSUBAME, Ranger)

Real-time tracking of technology curve
SC Technology Curve (x1.68 per Year)

Myrinet, Infiniband, etc.

Rapid Increase in the Top500 => RoadRunner
1st Peta in 2008
And this went to Petascale, Despite all the Skepticism

• TACC Ranger
  – The largest x86 Linux Cluster
    ~50,000 x86 cores
  – Half a Petaflop
• RR: the first #1 "commodity" cluster on Top500
  – The first #1 machine to use IB
  – The first #1 Linux machine
• The first #1 “heterogeneous” SC (Cell and Opteron)
“Why HPC Architecture Must be Custom Built in the Exascale Era”

2007-11-28
Slides Courtesy of Hisa Ando
(Former) Senior Architect
Fujitsu Ltd.

(Abridged and Translated by Satoshi Matsuoka)
Exaflop HPC Energy Consumption

- In SC07, Ray Orbach “Exascale by 2016”
- Energy Consumption at 90nm
  - FPU: \( \sim 500 \text{ pJ/DPFOP} \)
    - (GRAPE-DR: 65W/256GFlops=250pJ/DP FOP)
  - General Purpose CPU: \( 20 \text{ nJ/Cycle} \)
    - 4FOP/Cycle => x10 power over FPU

- 1 Exa Flops power requirement
  - Circa 2006-7: 90nm technology: \( 500\text{pJ} \times 10^{18} = 500 \times 10^6 \text{ W} \)
  - Circa 2016- (conservatively) suppose 22nm technology
    - Gate capacitance \( 22/90 = 0.24 \)
    - \( \text{Vcc} 0.8V/1V = 0.8 \)
    - Power \( \propto CV^2 = 0.24 \times 0.8^2 = 0.15 \)
    - 1 ExaFlop FPU array: \( 0.15 \times 500\text{pJ} \times 10^{18} = 75\text{MW}(!) \)
Why Special Architecture for Exascale?

- **Total System Power** $\approx 1.5\text{GW} \sim 2\text{GW}$ (!?)
  - Extrapolate to gen. purpose CPU: $75\text{MW} \times 10 = 750\text{MW}$
  - Memory, power delivery loss, cooling, I/O and storage… incur additional x2~x3 overhead
  - $>100$ million in Utility Bill(!)

- **Save Power, save power, and save power:**
  - **Objective:** 1/30 power reduction
    - Energy reduction of FPUs---low power design
    - SIMD-parallel control of massive FMA FPUs
      + Powerful scalar processor---beat Amdahl’s law

- **Claim (by Ando) such a processor cannot be general-purpose** (= for Commercial Apps)

- **I.e., Exascale machine must be (made of) special-purpose HPC architecture**
**Special Purpose Processor for Exascale circa 2016**

- **Server Processor**
  - 65nm technology
  - AMD 4 Core Opteron
  - Chip 283mm²
  - Core 26mm²
  - 128Fopx5GHz
  - 640GFlops

- **22nm technology**
  - 32 Cores
  - Chip 283mm²
  - Core 3.5mm²
  - 90nm technology
  - 32 Cores
  - Chip 283mm²
  - Core 3.5mm²

- **Custom HPC processor**
  - 8 Core (28mm²)
  - + 2048FMA (128mm²)
  - + 16MB L2$/LM (35mm²)
  - + I/O (60mm²)
  - Suppose: 16FMA/mm²
  - 4096Fopx5GHz
  - = 20TFlops

- **Grape-DR**
  - 90nm technology
  - 512FM+FA
  - 22nm technology
  - Chip ~250mm²
  - 25(FM + FA)s/mm²

---

**FUJITSU**

_The possibilities are infinite_
But wait, we now have this in commodity... the GPUs (Tesla, FireStream, Larrabee, ClearSpeed)

- **Thread Processor Cluster (TPC)**
- **Thread Processor Array (TPA)**
- **Thread Processor**

- x20 transitors (30 bil)
- 20TF FMA SFP
- 10TF FMA DFP

**nVidia Tesla T10**: 65nm, 600m², 1.4 bil Tr

1.08TF SFP

“Massive FMA FPUs”

“Powerful Scalar”

x86
16 cores
2.4Ghz
80GFlops

240 Cores
1.5Ghz
1.08TFlops SFP
90GFlops DFP

20GBs/s

102 GBytes/s

Tesla Accelerator

PCI-e
TSUBAME 1.2 Evolution (Oct. 2008)
The first “Petascale” SC in Japan

- Storage
  - 1.5 Petabyte (Sun x4500 x 60)
  - 0.1 Petabyte (NEC iStore)
- Lustre FS, NFS, CIF, WebDAV (over IP)
- Netowrk
  - 10Gbps+External NW
- Unified Infiniband network
- Sun x4600 (16 Opteron Cores)
  - 32~128 GBytes/Node
  - 10480core/655Nodes
  - 21.4 Terabytes
  - 50.4 TeraFlops
- OS Linux (SuSE 9, 10)
- NAREGI Grid MW

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

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

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

Nvidia Tesla T10P-one card per node, ~680 cards
High Performance in Many BW-Intensive Apps
10% power increase over TSUBAME 1.0 (130TF SFP / 80TF DFP)
TSUBAME Upgrades Towards Petaflops

Sustained Acceleration

US >10P (2011~12?)
Japanese NLP >10PF (2012-1Q)

LANL Roadrunner (Jun2008)
1PF

BlueGene/L
1.6 PB, 128GB nodes (2007)
360TF (2005)

Earth Simulator
40TF (2002)

Titech Campus Grid Clusters
1.3TF

TSUBAME 85TF
1.1 PB (2006)

TSUBAME 1.2 (?)
~900TF SFP
170 TF DFP (2008-2H)

U-Tokyo, Kyoto-U,
Tsukuba T2K 61-140TF (2008)

TSUBAME 2.0
>1 PF (2010)

TSUBAME 2.1
3PF (2011-2H)

Titech Campus Grid

IDC Servers and Cluster SC---the differences (or are there?)

• The Same---the nodes
  - Processors (x86)
  - Memory (DDR DRAM)
  - I/O (PCI-e)
  - OS (Linux/Windows), MW

• Differences
  - Network (IB vs GbE) (< 10% of machine cost)
    - Parallel Storage
    - Power Density
  - Parallel SW Stack: (MPI, OpenMP, BQ, …)
    - Operations as a SC

• Accelerators?
TSUBAME Network: \( \sim 1400 \) port

**Fat Tree, IB-RDMA & TCP-IP**

External Ether

Bisection BW = \( 2.88 \text{Tbps} \times 2 \)

Voltair ISR9288

IB 4x 10Gbps x 2

Single mode fiber for cross-floor connections

X4600 x 120 nodes (240 ports) per switch => 600 + 55 nodes, 1310 ports, 13.5 Tbps

X4500 x 60 nodes (60 ports) => 60 ports 600 Gbps
A Cloud Exaflops Machine

• Suppose 100MW power capacity, 1TFlop / 100W
• Hmm, this is easy, as my Nvidia GTX280 or AMD Radeon 4870 is ~200-250W incl. system overhead in SFP circa 2008...
• Need 10x => Expect 1TF DFP @ 100W system possible in 2012~13... (22nm)
• Network? Approx 10%-25% of system power and cost, so this is not a problem
• Rest are software and operational issues
• If Cloud centers can charge x2 standard fee, then it is a good business case
Mar 2012… the Japanese “> 10 PF” SC to be Online

25~30MW Power
~1mil ft2 floorspace
$1 bil construction

Kobe, Japan
“the site”
In fact we can build a Exaflop Cloud SC in 2012(!)

- @Tokyo---One of the Largest IDC in the World (in Toyosu, Tokyo... Built in 2003)
- Can fit a 10PF easy, 1 Exaflop in 2012
- On top of a 55KV/6GW Substation
- 150m diameter, 1,400,000 ft2 IDC floorspace
- 70+70 MW = 140MW power
- Can fit both Google/MS IDC or Any DOE center
- Remember interconnect cost 20% at most
- And can run Linux, Cloud/Grid interfaces, and HPC languages for accelerated & hybrid programming

- Merger of “SC Centers” & “Cloud”
Overview of HPC-GPGPU Project
(work w/MS Research TCI)

Research Focus

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
- Model-based GPGPU-CPU Load Balancing

X86 Scalar
GPGPU SIMD-Vector Acceleration
Scalar Multi-Core

Prototype Cluster
32-64 nodes, 50-100TFlops

Towards Next Gen Petascale
Personal Clusters and Desksides

Need x1000 acceleration over standard PCs
All-to-all 3-D Protein Docking Challenge

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

Blue Protein system
CBRC, AIST
(4 rack, 8192 nodes)
Calculation Flow for 3-D AA docking

Protein 1
3-D voxel data
NVIDIA CUDA
3-D complex convolution
candidate docking sites
cluster analysis
Host CPUs
Docking Confidence level

Protein 2
rotation (6 deg., 15 deg.)
3-D voxel data

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 Ops for $1000 \times 1000$
High Performance 3-D FFT on NVIDIA CUDA GPUs
[SC08 paper preview]

Akira Nukada
Tokyo Institute of Technology, GSIC.
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 an 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

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

X3 Faster than FFTW on GPUs

GFLOPS
Heavily Accelerated Prototype Cluster System Configuration

- 32 compute nodes
- 128 8800GTS GPGPUs
- one head node.
- Gigabit Ethernet network
- Three 40U rack cabinets.
- Windows Compute Cluster Server 2003 SP1, planned 2008 migration
- Visual Studio 2005 SP1
- nVidia CUDA 2.x
## 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 (4 racks)</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 (3 racks)</td>
<td>18</td>
<td>53.2</td>
<td>6.5</td>
<td>361</td>
</tr>
</tbody>
</table>

Can compute 1000x1000 in 1 month (15 deg.) or 1 year (6 deg.)\textsuperscript{24}
CFD on GPUs
(Material from Prof. Takayuki Aoki, Tokyo Tech.)

Safety

Nuclear (Cooling)

Weather/Environmental

Civil Engineering

Animations Courtesy Prof. Takayuki Aoki @ Tokyo Tech.
Riken Himeno Benchmark
(Prof. Takayuki Aoki, Tokyo Tech)

RIKEN Himeno CFD Benchmark

Poisson Equation: \( \nabla \cdot (\nabla p) = \rho \)

(Generalized coordinate)

\[
\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} + \alpha \frac{\partial^2 p}{\partial xy} + \beta \frac{\partial^2 p}{\partial xz} + \gamma \frac{\partial^2 p}{\partial yz} = \rho
\]

Discretized Form:

\[
\frac{p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k}}{\Delta x^2} + \frac{p_{i,j+1,k} - 2p_{i,j,k} + p_{i,j-1,k}}{\Delta y^2} + \frac{p_{i,j,k+1} - 2p_{i,j,k} + p_{i,j,k-1}}{\Delta z^2} \\
+ \alpha \frac{p_{i+1,j+1,k} - p_{i+1,j-1,k} - p_{i-1,j+1,k} + p_{i-1,j-1,k}}{4\Delta x\Delta y} \\
+ \beta \frac{p_{i+1,j,k+1} - p_{i+1,j,k-1} - p_{i+1,j-1,k} + p_{i+1,j-1,k}}{4\Delta x\Delta z} \\
+ \gamma \frac{p_{i,j+1,k+1} - p_{i,j+1,k-1} - p_{i,j-1,k+1} + p_{i,j-1,k-1}}{4\Delta y\Delta z} = p_{i,j,k}
\]

18 neighbor point access

Himeno for CUDA

1 block = 16x16x8 compute region
Block has 256 threads
Total 256 blocks = 65536 threads
Block shared mem = 16kB
Boundary region used for transfer
4 GPU node parallelization

Host

Open MP Parallel

64x64x32
64x64x32
64x64x32
64x64x32
64x64x128

Data exchange

PCI Express

GPUs

GeForce 8800 Ultra x 4

Data exchange

Data exchange

Data exchange
## Parallel Performance

### S Model [65x65x129]

<table>
<thead>
<tr>
<th>GPU Configuration</th>
<th>GFLOPS</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GPU (no data transfer)</td>
<td>30.6</td>
<td>0.269</td>
</tr>
<tr>
<td>2 GPU (16kB transfer)</td>
<td>42.5</td>
<td>0.193</td>
</tr>
<tr>
<td>4 GPU (32kB transfer)</td>
<td>51.9</td>
<td>0.158</td>
</tr>
</tbody>
</table>

x53.1 acceleration  

Reference:  

<table>
<thead>
<tr>
<th>GPU Configuration</th>
<th>GFLOPS</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GPU (no data transfer)</td>
<td>29.4</td>
<td>2.328</td>
</tr>
<tr>
<td>2 GPU (66kB transfer)</td>
<td>53.7</td>
<td>1.275</td>
</tr>
<tr>
<td>4 GPU (131kB transfer)</td>
<td>83.6</td>
<td>0.819</td>
</tr>
</tbody>
</table>

### M Model [129x129x257]

<table>
<thead>
<tr>
<th>GPU Configuration</th>
<th>GFLOPS</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GPU (no data transfer)</td>
<td>29.4</td>
<td>2.328</td>
</tr>
<tr>
<td>2 GPU (66kB transfer)</td>
<td>53.7</td>
<td>1.275</td>
</tr>
<tr>
<td>4 GPU (131kB transfer)</td>
<td>83.6</td>
<td>0.819</td>
</tr>
</tbody>
</table>

### L Model [257x257x512]

<table>
<thead>
<tr>
<th>GPU Configuration</th>
<th>GFLOPS</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GPU (no data transfer)</td>
<td>……….</td>
<td>……….</td>
</tr>
<tr>
<td>2 GPU (262kB transfer)</td>
<td>……….</td>
<td>……….</td>
</tr>
<tr>
<td>4 GPU (524kB transfer)</td>
<td>93.6</td>
<td>5.974</td>
</tr>
</tbody>
</table>

C.f. NEC SX-8 6 CPU (96GF Peak) 38.3GFLOPS Size XL
Two-dimensional Burgers Equation

\[
\begin{align*}
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= \kappa \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \\
\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= \kappa \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right)
\end{align*}
\]

GeForce 8800 GTS
40 GFLOPS

\[ u_{i,j}, v_{i,j} \]

\[ 1024 \times 1024 \]

velocity \( u \) at the \( v \)-point

\[ u_s = \frac{u_{i,j} + u_{i+1,j} + u_{i,j-1} + u_{i+1,j-1}}{4} \]
Homogeneous Isotopic Turbulence

Burgers equation

Poisson equation

\[
\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = \frac{1}{\Delta t} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)
\]

Correction

\[
\frac{\partial u}{\partial t} = -\frac{1}{\rho} \frac{\partial p}{\partial x} \quad \frac{\partial v}{\partial t} = -\frac{1}{\rho} \frac{\partial p}{\partial y}
\]

\[u_{i, j}, \quad p_{i, j}, \quad v_{i, j} \]

1024 × 1024
Two-Stream Instability in Plasma Physics

Vlasov-Poisson Equation:

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

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

\[
\begin{cases}
E = -\frac{\partial \phi}{\partial x}, & n_e = \int f dv
\end{cases}
\]

\(f\) : electron distribution function

\(n\) : electron number density
120 GFLOPS using 8800GTS

× 130
Rayleigh-Taylor Instability

Heavy fluid lays on light fluid and unstable.

**Euler equation:**

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

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

88 GFLOPS using GTX280

90× 512 x 512
Phase Separation

Phase transition dynamics is described by the Phase Field Model.

Cahn-Hilliard equation:

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

\[H : \text{free energy}\]

\[
\frac{\partial H}{\partial \psi} = \tau \psi - \nu \psi^3
\]

Discretization:

\[
\frac{\partial^4 \psi}{\partial x^4} = \frac{\psi_{i+2,j} - 4\psi_{i+1,j} + 6\psi_{i,j} - 4\psi_{i-1,j} + \psi_{i-2,j}}{\Delta x^4}
\]

\[
\frac{\partial^4 \psi}{\partial x^2 \partial y^2} = \left( \psi_{i+1,j+1} - 2\psi_{i,j+1} + \psi_{i-1,j+1} \right)
\]

\[
-2\psi_{i+1,j} + 4\psi_{i,j} - 2\psi_{i-1,j} + \psi_{i+1,j-1} - 2\psi_{i,j-1} + \psi_{i-1,j-1}
\]
3-D Computation of Phase Separation

Mixture of Oil and Water:

Used register number = 46
※ nvcc option –maxrregcount 32
for G80, 92

158 GFLOPS using GTX280

× 160 256 x 256 x 256
Can we make 100 million cores scale in “non-capacity capability app”?  

- Capability --- latency matters, strong scaling  
- \(~1\sim10\text{KB} 1\text{us} \) messages to be efficient \(\Rightarrow\) computation loop less than 1 us.  
  \(\Rightarrow\) Can only tolerate 1/1000 fluctuation i.e. both loop and communication will be 1ns, c.f. strong scaling code on a petaflops machine  
  \(\Rightarrow\) Even with 3-D stencils expect 1/30\sim1/100 i.e. 10-30ns  

Are we being hypocritical just to get money?
A Typical “Weak Scaling Capability App”
- Capacity App in Disguise -

```
Initialize;
Loop until computation gets done {
    MPI_AllScatter();
    Do work within node for seconds, minutes, hours…;
    MPI_AllGather();
}
Finalize;
```

--- And is grossly inefficient compared to say simple workstealing parameter-sweep esp. if load is unbalanced
So the world will mostly go ensemble --- capability at core, capacity at large ---

Barotropic S-Model
Ensemble climate simulation

QM/MM Molecular Simulation

“How are we to judge sciences, in that using 100,000 cores in a single MPI app has more scientific significance than 100,000 single-threaded app, as they both require system scalability in the design?”

BTW, MW may need to scale better for “capacity” e.g. BQ systems
TSUBAME Job Statistics
Dec. 2006-Aug. 2007 (#Jobs)
1400 SC Users

• 1400 SC Users, 797,886 Jobs (~3270 daily)
• 597,438 serial jobs (74.8%)
• 121,108 <=8p jobs (15.2%) 90%
• 129,398 ISV Application Jobs (16.2%)

However, >32p jobs account for 2/3 of cumulative CPU usage

Coexistence of ease-of-use in both - short duration parameter survey - large scale MPI (Both are hard for physically large-scale distributed grid)
Issues, Issues, Issues

- Large Scale Fault Tolerance
- Programming Models
- Storage (terabytes of bandwidth)

- Being Green: Power consumption and cooling
  - Not just the money but CO2 emission
- Sustainability

- $$$ - Subsidy Necessary

<table>
<thead>
<tr>
<th>Resources</th>
<th>Yens/CPU hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon E2C Single Instance</td>
<td>10.5</td>
</tr>
<tr>
<td>Tokyo Tech. GSIC SLA Svc.</td>
<td>0.543</td>
</tr>
<tr>
<td>Tokyo Tech. GSIC Best Effort Svc.</td>
<td>0.171</td>
</tr>
<tr>
<td>Univ. Tokyo Dedicated Q</td>
<td>0.89</td>
</tr>
</tbody>
</table>
Biggest Problem is Power...

<table>
<thead>
<tr>
<th>Machine</th>
<th>CPU Cores</th>
<th>Watts</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
The new JST-CREST "Ultra Low Power (ULP)-HPC" Project 2007-2012

Generalized Autotuning Scheme

Bayesian Merging of Model and Measurement
- Bayes model and distribution
  \[ y_i \sim N(\mu_i, \sigma_i^2) \]
  \[ \mu_i | \beta, \sigma_i^2 \sim N(\beta, 1/\sigma_i^2) \]
- Measured distribution after \( n \) trials
  \[ f(y_{1:n}, \mu_i^2, \beta, \nu_i) \]

Modeled ULP-HPC
- HW, Middleware, etc.
- MRAM
- PRAM
- Flash etc.

Optimize Power/Perf

Optimal Point

x10 Energy Efficiency

2016 TSUBAME becomes 1/1000
Himeno Size M Power Measurement

Jacobi

* CPU
  Av. Power: 160W
  Exec. Time: 62.88s

* GPU
  Av. Power: 170W
  Exec. Time: 3.75s

GPU/CPU
  Power: 106%
  Energy: 1/15.8
  = 6.33%

For extreme memory intensive CFD app GPU uses only 6% of CPU energy
Power Efficiency in 3-D FFD

<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></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></td>
</tr>
<tr>
<td>8800 GTS</td>
<td>On GPU</td>
<td>196 W</td>
<td>238 W</td>
<td>67.2</td>
<td></td>
</tr>
<tr>
<td>8800 GTX</td>
<td>On GPU</td>
<td>224 W</td>
<td>290 W</td>
<td>84.4</td>
<td></td>
</tr>
</tbody>
</table>

CUDA GPUs have four times higher power efficiency than CPU.

RIVA128 is an old, low-power GPU, to measure pure power consumption of host system (CPU, chipset, memory). The interface is legacy PCI.
Dividing the Labor in 2D FFT ([IEEE IPDPS-HCW08])

- Divide the labor for each 1D FFT among the CPUs/GPUs as

  GPU: 65%
  
  CPU1: 25%
  
  CPU2: 10%

E.g.: GPU:CPU1:CPU2 = 65:25:10
Performance Model of Combined CPU/GPU FFT Execution (1)

Predicted Overall Exec. Time

Max. predicted compute time for CPU/GPU Row-wise Computation

Max. predicted compute time for CPU/GPU Col-wise Computation

Execution Flow

CPU Thread

GPU Control Thread

Transpose

Row FFT In CPUs

Row FFT In GPU

Col FFT in-CPUs

Col FFT In GPU

Data Transfer to GPU

FFT planning etc.

FFT Execution

Data Transfer from GPU

Memory Release etc.

Transpose

Transpose

Transpose

Synchronization

Row-wise

Col-wise
Performance Model Accuracy (1)

- Problem Size $8192^2$, 1 CPU + GPU
- Predicted perf. of $8192^2$ derived from our model

![Graph showing the comparison between predicted and measured execution times.](image)

- Good match esp. with higher GPU labor
- Slight difference near optimal performance
Software-Based Error Checking on GPUs

- Protects applications from bit-flip errors in global memory using error-checking code (e.g., parity and ECC)
  - memory allocation → allocates additional global memory for storing error check code
  - write accesses → calculates code for data written
  - read accesses → calculates and compares code for data read

![Diagram of memory allocation and error checking](image)
N-body Problem with Error Checking

- The sample code shipped with the CUDA SDK
- Error checking
  - read accesses to position array
  - write accesses to position and velocity arrays

```c
nbody (float *pos, unsigned *pos_code,
       float *vel, unsigned *vel_code) {
    float4 *mypos = pos[i];
    check_float4(mypos, i);
    for (j in BLOCK) {
        shared float4 block[];
        block[k] = pos[k];
        check_float4(block[k], k);
        for (pt in block_size) {
            acc+=compute_acc(mypos,block[pt]);
        }
    }
    update_pos(pos, vel);
    write_check_code_float4(pos);
    update_vel(vel, acc);
    write_check_code_float4(vel);
}
```
Results of N-body Problem

CPU (4 cores with OpenMP) vs GPU vs GPU+Parity vs GPU+ECC

Phenom 9850 2.5 GHz, 4 GB of DRAM, Linux v2.6.23, gcc v4.1.2, GeForce 8800 GTS 512, CUDA v2.0 Beta 2

- Less than 7% of overhead even with ECC!
- Much slower performance of CPU

Number of bodies: 1024, 2048, 4096, 8192, 1638
Elapsed time: 500, 1000, 1500, 2000, 2500
Road to Exascale

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

Earth Simulator
40TF (2002)

KEK 59TF
1.3TF

GPGPU-Node
40TF (2005)

BlueGene/L
3.6TF (2008)

RoadRunner
1.5TF (2008)

TSUBAME 1.0
8.5TF

TSUBAME 2.0
1TF

Japanese NLP
>10PF (2012)

KEK
59TF

Titech
Campus
Grid
1.3TF

BG/L+SR11100


2016 TSUBAME becomes desktop

2019 1ExaFlops (Leadership machines)
TSUBAME 4.0 > 100PF
So the first exascale machine will be a Cloud with massive user base (?)