Technologies beyond the K computer

September 5th, 2012

Takashi Aoki
Next Generation Technical Computing Unit
Fujitsu Limited
Agenda

- Corporate profile
- Fujitsu supercomputer past and present
- Second generation Petascale supercomputer PRIMEHPC FX10
  - Hardware
  - Software
- Challenge to the future
Japan’s largest IT services provider and No. 3 in the world. *

We do everything in ICT. We use our experience and the power of ICT to shape the future of society with our customers.

Over 170,000 Fujitsu people support customers in more than 100 countries.

Our products and services

Technology Solutions

**Services**

Our datacenters in the world

**Systems platform**

- PRIMERGY TX120
- ETERNUS DX8000
- Supercomputer PRIMEHPC FX10

Ubiquitous Product Solutions

**Device solutions**

- LIFEBOOK E751C
- Smart phone F07D
- Tablet PC ARROWS

- High-end multi-core processor SPARC64 VII+
- FM3 family (32-bit RISC MCU)
- FRAM (Ferroelectric Random Access Memory)
‘shaping tomorrow with you’ wherever you are.

Over 170,000 Fujitsu colleagues working with customers in over 100 countries  

As of March 2012
Full range coverage with choice of HPC hardware platform

- **High Performance scaling over several PFlops**
  - Fujitsu propriety CPU and interconnect technologies for high performance, high reliability and high operability

- **High Performance de facto HPC cluster**
  - Following Intel CPU and MIC roadmap and adopt Fujitsu latest packaging technologies for high performance and high operability

**HPC Platform Solutions - Hardware -**

- High-end
  - Petascale Supercomputer
    - PRIMEHPC FX10

- Divisional
  - x86 HPC Cluster
    - CX400 Skinless server

- Departmental
  - BX Series
    - BX900
    - BX400

- Work Group
  - RX Series
    - RX200
  - PRIMERGY series

- Large-Scale SMP System
  - RX900

**HPC Platform Solutions - Hardware -**

- High-end
  - Petascale Supercomputer
    - PRIMEHPC FX10

- Divisional
  - x86 HPC Cluster
    - CX400 Skinless server

- Departmental
  - BX Series
    - BX900
    - BX400

- Work Group
  - RX Series
    - RX200
  - PRIMERGY series

- Large-Scale SMP System
  - RX900

**HPC Platform Solutions - Hardware -**

- High-end
  - Petascale Supercomputer
    - PRIMEHPC FX10

- Divisional
  - x86 HPC Cluster
    - CX400 Skinless server

- Departmental
  - BX Series
    - BX900
    - BX400

- Work Group
  - RX Series
    - RX200
  - PRIMERGY series

- Large-Scale SMP System
  - RX900
Design targets and features of FX10

- **High Performance**
  - High peak performance and high application performance

- **High parallel application productivity**
  - Easy to achieve high performance running highly paralleled programs without inordinate effort of programming

- **Customer’s requirement and FX10 design targets**

- **High operability**
  - Low power consumption
  - High reliability and ease of operation

- **K computer compatibility**
  - Binary compatibility
  - Same programming environment
Design targets and features of FX10

- High Performance
  - High-performance CPU
    - "SPARC64 IXfx" with SPARC V9 + HPC-ACE architecture
  - High performance, highly reliable and fault tolerant 6D mesh/torus interconnect "Tofu"*

- High operability
  - Low power consumption
  - High reliability and ease of operation

- Water cooling system

- High reliability components & functions based on mainframe development experience

- High parallel application productivity
  - Easy to achieve high performance

- "VISIMPACT"* supports efficient hybrid parallel execution

- Parallel Language, programing tools and Petascale HPC middleware for high reliability and operability

- K computer compatibility
  - Binary compatibility
  - Same programing environment

---

*1) Tofu: Torus Fusion
*2) VISIMPACT: Virtual Single Processor by Integrated Multicore Parallel Architecture
PRIMEHPC FX10 System Configuration

**PRIMEHPC FX10**

- **Compute Nodes**
- **I/O nodes**
- **Local disks**
- **Local file system**
- **Tofu interconnect for I/O**

**Compute node configuration**

- **SPARC64™ IXfx CPU**
- **DDR3 memory**
- **ICC (Interconnect Control Chip)**

**Network (IB or GB)**

- **Management servers**
- **Portal servers**
- **Login server**
- **File servers**
- **Global disk**
- **Global file system**

**Local file system**

- **IB**: InfiniBand
- **GB**: GigaBit Ethernet

Copyright 2012 FUJITSU LIMITED
## FX10 System H/W Specifications

### PRIMEHPC FX10 H/W Specifications

<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>SPARC64™ IXfx</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Performance</td>
<td><a href="mailto:236.5GFlops@1.848GHz">236.5GFlops@1.848GHz</a></td>
</tr>
<tr>
<td>Node</td>
<td>Configuration</td>
<td>1 CPU / Node</td>
</tr>
<tr>
<td></td>
<td>Memory capacity</td>
<td>32, 64 GB</td>
</tr>
<tr>
<td>Rack</td>
<td>Performance/rack</td>
<td>22.7 TFlops</td>
</tr>
<tr>
<td>System (4 ~1024 racks)</td>
<td>No. of compute node</td>
<td>384 to 98,304</td>
</tr>
<tr>
<td></td>
<td>Performance</td>
<td>90.8TFlops to 23.2PFlops</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
<td>12 TB to 6 PB</td>
</tr>
</tbody>
</table>

- **SPARC64™ IXfx CPU**
  - 16 cores/socket
  - 236.5 GFlops

- **System rack**
  - 96 compute nodes
  - 6 I/O nodes
  - With optional water cooling exhaust unit

- **System board**
  - 4 nodes (4 CPUs)

- **System**
  - Max. 23.2 PFlops
  - Max. 1,024 racks
  - Max. 98,304 CPUs
### The K computer and FX10 Comparison of System H/W Specifications

<table>
<thead>
<tr>
<th></th>
<th><strong>K computer</strong></th>
<th><strong>FX10</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td><em>SPARC64™ VIIIfx</em></td>
<td><em>SPARC64™ IXfx</em></td>
</tr>
<tr>
<td>Performance</td>
<td>128 GFlops@2GHz</td>
<td>236.5 <a href="mailto:GFlops@1.848GHz">GFlops@1.848GHz</a></td>
</tr>
<tr>
<td>Architecture</td>
<td>SPARC V9 + HPC-ACE extension</td>
<td>←</td>
</tr>
<tr>
<td>Cache configuration</td>
<td>L1(I) Cache: 32KB/core, L1(D) Cache: 32KB/core</td>
<td>←</td>
</tr>
<tr>
<td></td>
<td>L2 Cache: 6MB(shared)</td>
<td>L2 Cache: 12MB(shared)</td>
</tr>
<tr>
<td>No. of cores/socket</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Memory band width</td>
<td>64 GB/s.</td>
<td>85 GB/s.</td>
</tr>
<tr>
<td><strong>Node</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>1 CPU / Node</td>
<td>←</td>
</tr>
<tr>
<td>Memory capacity</td>
<td>16 GB</td>
<td>32, 64 GB</td>
</tr>
<tr>
<td><strong>System board</strong></td>
<td>Node/system board</td>
<td>←</td>
</tr>
<tr>
<td></td>
<td>4 Nodes</td>
<td></td>
</tr>
<tr>
<td><strong>Rack</strong></td>
<td>System board/rack</td>
<td>←</td>
</tr>
<tr>
<td></td>
<td>24 System boards</td>
<td></td>
</tr>
<tr>
<td>Performance/rack</td>
<td>12.3 TFlops</td>
<td>22.7 TFlops</td>
</tr>
</tbody>
</table>
### The K computer and FX10 Comparison of System H/W Specifications (cont.)

<table>
<thead>
<tr>
<th>Interconnect</th>
<th><strong>K computer</strong></th>
<th><strong>FX10</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
<td>6D Mesh/Torus</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>5GB/s x2 (bi-directional)</td>
<td>←</td>
</tr>
<tr>
<td>No. of link per node</td>
<td>10</td>
<td>←</td>
</tr>
<tr>
<td>Additional features</td>
<td>H/W barrier, reduction</td>
<td>←</td>
</tr>
<tr>
<td></td>
<td>no external switch box</td>
<td>←</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cooling</th>
<th><strong>K computer</strong></th>
<th><strong>FX10</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU, ICC(interconnect chip), DDCON</td>
<td>Direct water cooling</td>
<td>←</td>
</tr>
<tr>
<td>Other parts</td>
<td>Air cooling</td>
<td>Air cooling + Exhaust air water cooling unit (Optional)</td>
</tr>
</tbody>
</table>

Cooling:
- CPU, ICC(interconnect chip), DDCON
- Direct water cooling
- Air cooling
- Air cooling + Exhaust air water cooling unit (Optional)
Node configuration

- Single CPU as a node
  - SPARC64™ IXfx based
  - 32/64GB memory capacity
  - Single CPU per node to maximize memory BW
  - High memory bandwidth of 85 GB/s

- On board InterConnect Controller (ICC)
  - Direct RDMA and global synchronization operations
  - No external switch

- Node type
  - Compute node
    - Consist of CPU, ICC and memory
    - No I/O capability except interconnect
    - Four nodes are mounted on a system board
  - I/O node
    - Same CPU as compute node
    - Includes four PCI Express Gen2 x8 slots
    - 8 GB/s I/O bandwidth per I/O node
    - One node is mounted on an I/O system board
High-performance and low-power multi-core CPU

- High performance core by HPC-ACE
  - Multiply number of register, SIMD operation, software controllable cache, etc.
- VISIMPACT: Support highly efficient hybrid execution model (thread + process)
  - Shared second cache, hardware barrier among cores and compiler support

### SPARC64™ IXfx specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>SPARC V9 + HPC-ACE</td>
</tr>
<tr>
<td># of FP operations /clock/core</td>
<td>8 (= 4 Multiply and Add)</td>
</tr>
<tr>
<td>No. of cores</td>
<td>16</td>
</tr>
<tr>
<td>Peak performance and clock</td>
<td>236.5 <a href="mailto:Gflops@1.848GHz">Gflops@1.848GHz</a></td>
</tr>
<tr>
<td>Memory bandwidth</td>
<td>85 GB/s</td>
</tr>
<tr>
<td>Power consumption</td>
<td>110 W (typical)</td>
</tr>
</tbody>
</table>

- High performance-per-power ratio and High reliability
  - Water cooling system has lowered the CPU temperature and leak current
  - Wide-ranging error detection/self-recovery functions, instruction retry function
Overview of HPC-ACE

“High Performance Computing - Arithmetic Computational Extensions”

- Extended number of integer registers and floating point registers
- Software-controllable “Sector Cache”
- Flexible Single Instruction Multiple Data (SIMD) operation
- Hardware barrier synchronization for VISIMPACT
  - VISIMPACT: automatic thread-parallelization compiler technology
- Other special features
  - XFILL instruction
  - Reciprocal approximation instruction
  - Reciprocal square root approximation instruction
  - Trigonometric function acceleration instructions
HPC-ACE: Extended Number of Registers

- Enables larger loop unrolling and eliminates register spills

**Integer registers**
- SPARC-V9: 160 / 32
- HPC-ACE: 192 / 64

**Double precision floating-point registers**
- SPARC-V9: 32
- HPC-ACE: 256 (Scalar) / 128 (SIMD)
**NPB3.3-LU** high cost loop

- By using extended number of registers, compiler can generate more efficient scheduling and also eliminate unnecessary memory operations.

```
39 1  
do j = jst, jend 
<<< Loop-information Start >>>
<<< [OPTIMIZATION]
<<< PREFETCH : 8
<<< c : 8
<<< Loop-information End >>>
do l = ist, iend

40 2 2
41 2 2
c form the block diagonal
42 2 2
43 2 2
44 2 2
45 2 2
tmp1 = 1.0d+00 / u(1,i,j,k)
tmp2 = tmp1 * tmp1
tmp3 = tmp1 * tmp2
46 2 2
47 2 2
48 2 2
49 2 2
50 2 2
d(1,i,j) = 1.0d+00
+ dt * 2.0d+00 * ( tx1 * dx1
51 2 2
52 2 2
53 2 2
d(1,i,j) = 0.0d+00
54 2 2
d(1,i,j) = 0.0d+00
55 2 2
d(1,i,j) = 0.0d+00
56 2 2
d(1,i,j) = 0.0d+00
57 2 2
58 2 2
59 2 2
60 2 2
61 2 2
62 2 2
63 2 2
64 2 2
65 2 2
66 2 2
67 2 2
68 2 2
c(5,i,j) = - dt * tx2
69 2 2
70 2 2
71 2 2
72 2 2
73 2 2
74 2 2
c(5,i,j) = d t * tx2
75 2 2
76 2 2
77 2 2
78 2 2
79 2 2
80 2 2
81 1 2
end do
end do
```

![Graph showing 32 registers vs 256 registers with a 1.42 improvement]
Performance boost by 256 FP registers w/ 138 application program kernels

<table>
<thead>
<tr>
<th>Program No.</th>
<th>Improved ratio</th>
<th>Performance improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>120%</td>
<td>Average</td>
</tr>
<tr>
<td></td>
<td>252%</td>
<td>Max.</td>
</tr>
</tbody>
</table>

Performance improvement by # of FP registers extension (from 32 to 256)
Increasing the cache hit rate by selectively leave a reused data in the cache

- The cache is divided into two sectors (Sectors 0 and 1).
- Sector 1 is used for data that will be reused.
- Sector 0 is used for other data.

Data in Sector 1, which will be used again soon, is no longer removed from cache, by the access of data that uses Sector 0.

- The user can specify the data to be retained in Sector 1 by specifying it on the compiler directive line.

```
!ocl CACHE_SECTOR_SIZE(N1,N2)
!ocl CACHE_SUBSECTOR_ASSIGN(a)
do j=1,m
   do i=1,n
      a(i) = a(i) + b(i,j) * c(i,j)
   enddo
endo
do
```

Array `a` is no longer removed from the cache by references to array `b` or `c`. Array `a` is held in Sector 1.
- All others are held in Sector 0.
**NPB3.3-CG case**

- By putting array P on sector 1, floating point data cache access wait is reduced

---

**Optimized code**

```c
int i, j, k;
float a, b, c, d, e, f, g, h, x, y, z;

!ocl CACHE_SECTOR_SIZE(4,8)
!ocl CACHE_SUBSECTOR_ASSIGN(p)

120 1

!----- npb_cg kernel loop -----
<<<< Loop-information Start >>>
<<<< [PARALLELIZATION]
<<<< Standard iteration count: 4
<<<< Loop-information End >>>

121 2 pp
do j=1,n
122 2 p
sum = 0.d0
<<<< Loop-information Start >>>
<<<< [OPTIMIZATION]
<<<< SIMD
<<<< SOFTWARE PIPELINING
<<<< Loop-information End >>>

123 3 p 4v
    do k=rowstr(j), rowend(j) ! 64LOOP
124 3 p 4v
    sum = sum + a(k) * p(colidx(k))
125 3 p 4v
    enddo
126 2 p
    q(j) = sum
127 2 p
    enddo
128 1

!ocl END_CACHE_SUBSECTOR
!ocl END_CACHE_SECTOR_SIZE
```

---

**Graph**

- **Before (w/o sector)**: 2.5E-01 seconds
- **After (with sector)**: 1.5E-01 seconds

**x 1.23 improvement**
Eight floating-point ops can be executed simultaneously per core
- Two SIMD instructions can be executed simultaneously per core
- SIMD instruction executes two floating-point ops (single or double precision)
- FMA is supported

Software can flexibly perform SIMD optimization
- It is possible to execute operations in SIMD by obtaining pieces of data one by one from noncontiguous memory spaces
- It is possible to selectively store floating register into memory (mask operation)

|-----------|-----------|-----------|-----------|

Floating-point Pipelines

Floating-point Registers

SIMD basic

SIMD extended
Example of Computational chemistry program

- Due to the branch operation, “if” in the loop, SIMD option shows NO effect
- By using mask operation, compiler can SIMDize the loop and utilize software pipelining. Results 2.5x performance improvement

40 1 do iv=1, natv
41 1 local unroll(4)
42 1 !$omp parallel do default(none)
43 1 !$omp private(earg,work)
44 1 !$omp shared(ngr,iv,uuv,tuv,tuvres)
<<<< Loop-information Start>>>>
<<<< [OPTIMIZATION]
<<<< SIMD
<<<< SOFTWARE PIPELINING
<<<< Loop-information End>>>>

45 2 p 4v do ig=1, ngr
46 2 p 4v earg = -uuv(ig,iv) + tvu(ig,iv,1)
47 3 p 4v if (earg >= 0) then
48 3 p 4v work = 1.0d0 + earg
49 3 p 4v else
50 3 p 4v work = exp(earg)
51 3 p 4v endif
52 2 p 4v tvuves(ig,iv,1) = work
53 2 p 4v enddo
54 1
eendo
55 1 !$omp parallel

HPC-ACE: SIMD extension (mask operation effect)

x 2.5 improvement
**HPC-ACE: XFILL capability**

- XFILL capability works in *Earthquake simulation program*
  - XFILL fills L2 cache line with undetermined data (allocate cache line without data load)
  - So, with XFILL in advance, following FP reg store instructions should hit and would not cause data load from memory
  - XFILL can reduce memory read accesses and improve performance when a memory throughput is the bottleneck

```plaintext
7    integer, parameter :: NXP  = 400  ! X-size per 1 pro
8    integer, parameter :: NYP  = 200  ! Y-size per 1 pro
9    integer, parameter :: NZ   = 4300  ! Z-size per 1 pro
184   1 p do J = 1, NY
185    2 p do I = 1, NX
        <<< Loop-information Start >>>
        <<< [OPTIMIZATION]
        <<<   SIMD
        <<<   SOFTWARE PIPELINING
        <<<   PREFETCH  : 2
        <<<   DZV: 2
        <<<   XFILL : 2
        <<<   DZV: 2
        <<< Loop-information End >>>
186   3 p v do K = 3, NZ-1
187   3 p v   DZV (k,i,J) = (V(k,i,J) - V(k-1,i,J))*R40 &
188       3  - (V(k+1,i,J) - V(k-2,i,J))*R41
189   3 p v end do
```

---

**Figure:**
- x 1.5 improvement

**Graph:**
- [sec.]
- Without XFILL
- With XFILL

---

Copyright 2012 FUJITSU LIMITED
**VISIMPACT technology**

- Fine-grain thread-parallelization
  - Low-overhead barrier synchronization with HPC-ACE ASI registers
  - Coalesced memory access exploits shared L2 cache
  - “Virtual Single Processor by Integrated Multi-core Parallel Architecture”

- Fujitsu compilers support VISIMPACT automatic parallelization

---

**Vectorization**

\[
\text{DO } J=1,N \\
V \quad \text{DO } I=1,M \\
V \quad A(I,J)=... \\
V \quad \text{END}
\]

**Conventional Threading**

\[
P \quad \text{DO } J=1,N \\
P \quad \text{DO } I=1,M \\
P \quad A(I,J)=... \\
P \quad \text{END}
\]

**VISIMPACT**

\[
P \quad \text{DO } I=1,M \\
P \quad A(I,J)=... \\
P \quad \text{END}
\]

---

requires separate or large L2 cache
Fujitsu compiler transforms MPI programs to hybrid parallel executions automatically, by parallelizing a process on a CPU into multi-threads to cores.

By reducing the number of ranks, communication efficiency would be improved.

Inter-core hardware barrier and shared L2 cache help efficient execution.
6D-Mesh/Torus Network Topology

- Higher bisection bandwidth and smaller hops than 3D-Torus
- Torus fusion
  - Every XYZ Cartesian grid point has another ABC 3D-Torus
  - X, Z and B are torus (ring) axes
  - A, C and Y are mesh (linear) axes

Conceptual Model
- System software generates virtual 1d-, 2d- or 3d-torus for an arbitrary size of 6d-cuboid

- Virtual topology expands the range of applicable algorithms
ICC : Tofu Interconnect Controller

- Companion chip for SPARC64™ VIIIfx / IXfx processors
- Tofu Interconnect
  - 4 Tofu Network Interfaces
  - Tofu Network Router
- PCI Express Gen2
  - 2 ports for I/O nodes
- Water-cooled

<table>
<thead>
<tr>
<th>Process technology</th>
<th>65 nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die size</td>
<td>18.2 mm x 18.1 mm</td>
</tr>
<tr>
<td>Frequency</td>
<td>312.5 MHz</td>
</tr>
<tr>
<td>No. of Tofu link</td>
<td>10 ports</td>
</tr>
<tr>
<td>Tofu link throughput</td>
<td>in 5 GB/s + out 5 GB/s</td>
</tr>
<tr>
<td>PCI Express Gen2</td>
<td>8 lane x 2 ports</td>
</tr>
<tr>
<td>Host Bus Interface</td>
<td>in 20 GB/s + out 20 GB/s</td>
</tr>
<tr>
<td>Power consumption</td>
<td>28 W (typical)</td>
</tr>
<tr>
<td>No. of transistors</td>
<td>200 million</td>
</tr>
<tr>
<td>Signal Transfer Speed</td>
<td>6.25 Gbps</td>
</tr>
<tr>
<td>Differential signals</td>
<td>128 lanes</td>
</tr>
</tbody>
</table>
Static and Dynamic Failure Avoidance

- Static Failure Avoidance
  - Pre-calculated routing table
  - For intra-job communication

- Dynamic Failure Avoidance
  - Time-out detection by the protocol
  - For I/O communication
Fault Isolation by Virtual Topology

- Jobs using virtual topology can use rectangle region including failed node

- Decreases in executable job size and in system availability are minimized
**All-to-all communication performance**

- Link utilization is important for actual communications
- New optimized algorithm
  - Uses all links uniformly to maximize All-to-All communication performance
  - Four RDMA engines execute 4 sends and 4 receives simultaneously
- Using Tofu features
  - Virtual 3D-Torus
  - Flow-control features
    - for congestion prevention
- Many applications use All-to-All type of communication and enjoy this acceleration

![Graph showing All-to-all communication performance](image)
All-to-all communication trace on Tofu

Trace Result of the K computer

System configuration of Tofu
24 × 18 × 16 × 2 × 3 × 2 = 82,944 nodes
Each node transfers 32KB

Left: new algorithm
Right: standard OpenMPI
(pair-wise exchange)

Colors show link utilization and wait time
Greener – Higher utilization
Redder – Longer wait time

New Algorithm
Elapsed Time: 2.77sec

Standard OpenMPI
(pair-wise exchange)
Elapsed Time: 24.08sec
FX10 Software Stack

Applications

HPC Portal / System Management Portal

Technical Computing Suite

System Management
- System management
- System control
- System monitoring
- System operation support

Job Management
- Job manager
- Job scheduler
- Resource management
- Parallel job execution

High Performance Parallel File System
- Lustre based high performance distributed file system
- High scalability, high reliability and availability

FEFS

Automatic parallelization compiler
- Fortran
- C
- C++

Tools and math. libraries
- Programming support tools
- Mathematical libraries
  (SSL II/BLAS etc.)

Parallel languages and libraries
- OpenMP
- MPI
- XPFortran

Linux based OS enhanced for FX10

PRIMEHPC FX10

Copyright 2012 FUJITSU LIMITED
Lustre Extension of FEFS: Features

FEFS Features

Large scale
- Max file size
- Max number of files
- Max client number
- Max stripe count
- 512KB block

High performance
- File striping
- Parallel I/O
- MDS response
- I/O zoning
- Client cache
- Server cache
- OS jitter reduction

Network
- Tofu Interconnect
- IB/Ether
- IB Multi-rail
- LNET Router

Connectivity
- Lustre mount
- NFS export

Lustre Features

Operations Management
- ACL
- Disk Quota
- Directory Quota
- Dynamic configuration change

Reliability
- Failover
- Journal / fsck
- RAS

New
Extended
Reuse
Achieved the world’s top-level throughput*
- Read 334GB/s, Write 249GB/s
  (574 OSSs, 18432 Clients, 192 racks)

Metadata performance of mdtest* (distributed directory)

<table>
<thead>
<tr>
<th>FEFS</th>
<th>Lustre</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IOPS</strong></td>
<td></td>
</tr>
<tr>
<td>K computer**</td>
<td>IA***</td>
</tr>
<tr>
<td>create</td>
<td>34697.6</td>
</tr>
<tr>
<td>unlink</td>
<td>39660.5</td>
</tr>
<tr>
<td>mkdir</td>
<td>87741.6</td>
</tr>
<tr>
<td>rmdir</td>
<td>28153.8</td>
</tr>
</tbody>
</table>

* : Collaborative work with RIKEN on the K computer

** : MDS:RX300S6 (X5680 3.33 GHz 6core x2, 48GB, IB(QDR)x2)
*** : MDS:RX200S5 (E5520 2.27GHz 4core x2, 48GB, IB(QDR)x1)
Language System overview

- Fortran C/C++/Fortran Compiler
- Programming model (OpenMP, MPI, XPFortran)
- Instruction level /Loop level optimization using HPC-ACE
- Debugging and Tuning tools for highly parallel computer

---

### Programming Language, MPI

- **Intra Node**
  - Fortran 2003
  - C
  - C++
  - OpenMP 3.0
  - XPFortran *1
  - MPI 2.1

- **Inter Node**

### Programming tool

- IDE
- Debugger
- Profiler
- RMATT *2

### Math. Lib.

- BLAS
- LAPACK
- SSL II
- ScaLAPACK

---

*1: eXtended Parallel Fortran (Distributed Parallel Fortran)
*2: Rank Map Automatic Tuning Tool
Application Tuning Cycle and Tools

- Job Information
- Profiler
  - Vampir-trace
- MPI Tuning
- Overall Tuning
- CPU Tuning
- FX10 Specific Tools
  - Profiler
    - Vampir-trace
  - Open Source Tools
    - PAPI
- RMATT
- Tofu-PA

Profiler snapshot
On Course to Exascale

- World’s first 1 Exa-Flops computer is expected to appear by 2020
Realization of Exascale system is grand challenge
- At least two-step development is necessary
- The biggest challenge is high density and low power consumption

Fujitsu is developing a Trans-Exa system as a midterm goal
- The Trans-Exa system is expected to be scalable to 100 Petaflops
- Employs
  - Wide SIMD and multicore CPU
  - High performance and lower power consumption interconnect
  - High performance and high density memory technologies

Continues to invest effort in research for the exascale system
- Higher performance and lower power consumption technologies
- Technologies for higher reliability
Key technology developments on Trans-Exa

Goal

Significant improvement of power efficiency, high density

Technology

- Silicon tech. ⇒ Employs the latest tech.
- Innovative memory tech. ⇒ High density & BW memory
- System integration tech. ⇒ Higher integration & density
- The latest optical tech. ⇒ High speed signal transfer

Gains

- Performance / power consumption
- Performance / rack
- Accumulation of key technologies toward exascale systems
shaping tomorrow with you