Development of the Post-K Supercomputer

Toshiyuki Shimizu
Vice president, System development division, Next generation technical computing unit

Post-K is under development, information in these slides is subject to change without notice

November 14th, 2016
Updated on Nov. 18th
FUJITSU HPC and Post-K Development

- Performance of high-end machines preceding the Post-K
- Post-K
  - Goals and approaches
  - Post-K hardware
  - Post-K software, compilers and their performance
- Summary

Copyright 2016 FUJITSU LIMITED
Achievements with the K computer

- **Prestigious Benchmark Awards**
  - TOP500: 10.5Pflops, 93% efficiency
  - HPCG: 602Tflops, 5.3% efficiency
  - Graph500: 38.6TTEPS
  - HPC Challenge Class 1: No.1 at all categories
    - (1) Global HPL, (2) Global RandomAccess, (3) EP STREAM, (4) Global FFT

- **Gordon Bell Prize Awards**
  - “First principles calculation of electronic states of a silico nanowire with 100,000 atoms on the K computer” (2011)
  - “Simulations of Below-Ground Dynamics of Fungi: 1.184 Pflops Attained by Automated Generation and Autotuning of Temporal Blocking Codes” (2016 finalist)

Updated on Nov. 18 at SC16
6 years from the shipment

Nominated as finalist
Post-K Goals and Approaches

Goals
- High application performance and good power efficiency
- Keeping application compatibility while advancing from predecessors
- Good usability and better accessibility for users

Approaches
- Developing high performance and scalable, custom CPU cores
  - Performance: Wider SIMD & high memory BW, mathematical acc. primitives
  - Scalability: SMaC (scalable many core), zero OS jitter (assistant core)
  - Power efficiency: The best device tech, power control functions, optimal resources
- Maintaining performance balance and supporting advanced features
  - High memory BW, “Tofu” interconnect, and RIKEN advanced system software
- Adopting ARM standard architecture
  - Co-operation with ARM/Linux community and utilization of open source software
  - Getting involved in the ARM HPC ecosystem
Post-K Powered by FUJITSU-designed CPU Cores & Tofu

- FUJITSU CPU cores support ARM SVE ISA
  - FUJITSU, as a lead partner in ARM SVE development, contributes to specification of ARM SVE (Scalable Vector Extension), for application performance
  - FUJITSU ARM core incorporates FUJITSU’s proven supercomputer microarchitecture
- ARM SVE, plus optional functions and Tofu, maintain programming models and performance balance
- Post-K complies ARM’s standard frameworks (SBSA, etc.), for compatibility among platforms

<table>
<thead>
<tr>
<th>Functions for Perf.</th>
<th>Post-K</th>
<th>FX100</th>
<th>FX10</th>
<th>K computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVE incorporated</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIMD</td>
<td></td>
<td>512bit</td>
<td>256bit</td>
<td>128bit</td>
</tr>
<tr>
<td>FMA4</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Math. acc. prim.*</td>
<td>✔ Enhanced</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Optional functions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inter-core barrier</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Sector cache</td>
<td>✔ Enhanced</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Prefetch modes</td>
<td>✔ Enhanced</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Interconnect</td>
<td>✔ Enhanced</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

*Mathematical acceleration primitives include trigonometric functions, sine & cosines, and exponential...
System Software for Post-K

Currently in development, based on “co-design” scheme with application developers, including system hardware

### Post-K Applications

<table>
<thead>
<tr>
<th>Management Software</th>
<th>Hierarchical File I/O Software</th>
<th>Programming Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>System management for highly available &amp; power saving operation</td>
<td>Application-oriented file I/O middleware</td>
<td>XcalableMP</td>
</tr>
<tr>
<td>Job management for higher system utilization &amp; power efficiency</td>
<td>Lustre-based distributed file system FEFS</td>
<td>MPI (Open MPI, MPICH)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OpenMP, COARRAY, Math Libs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Compilers (C, C++, Fortran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Debugging and tuning tools</td>
</tr>
</tbody>
</table>

Linux OS / McKernel (Lightweight Kernel)

Post-K System Hardware
FUJITSU Compiler for Post-K

- Maximizes the execution performance of HPC applications
  - Covers a wide range of applications, including integer calculations are dominant
- Targets 512bit-wide vectorization as well as Vector-length-agnostic
  - Fixed-vector-length facilitates optimizations such as constant folding
- Inherits options/features of K computer, PRIMEHPC FX10 and FX100

Language Standard Support
- Fully supported: Fortran 2008, C11, C++14, OpenMP 4.5
- Partially supported: Fortran 2015, C++1z, OpenMP 5.0

Supports ARM C Language Extensions (ACLE) for SVE
- ACLE allow programmers to use SVE instructions as C intrinsic functions

```
// C intrinsics in ACLE for SVE
svfloat64_t z0 = svld1_f64(p0, &x[i]);
svfloat64_t z1 = svld1_f64(p0, &y[i]);
svfloat64_t z2 = svadd_f64_x(p0, z0, z1);
svst1_f64(p0, &z[i], z2);
```

```
// SVE assembler
ld1d  z1.d, p0/z, [x19, x3, lsl #3]
ld1d  z0.d, p0/z, [x20, x3, lsl #3]
fadd  z1.d, p0/m, z1.d, z0.d
st1d  z1.d, p0, [x21, x3, lsl #3]
```
Vectorization by FUJITSU Compiler

- Dynamic instruction counts of representative loops of NPB 3.3-SER

<table>
<thead>
<tr>
<th>TSVC</th>
<th>FX100</th>
<th>Post-K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fortran (135)</td>
<td>96</td>
<td>111</td>
</tr>
<tr>
<td>C (151)</td>
<td>106</td>
<td>121</td>
</tr>
</tbody>
</table>

// Sample of vectorized loop by SVE
// s482
for (int i = 0; i < LEN; i++) {
  a[i] += b[i] * c[i];
  if (c[i] > b[i]) break;
}

Summary of Post-K Development

- Developing high performance, scalable, custom CPU cores
  - SMaC architecture with an assistant core for scalable performance
  - ARM instruction set architecture, SVE, as a standard architecture
  - ARM standard frameworks, SBSA, etc., for compatibility among platforms

- Keeping performance balanced and advancing preceding machines
  - Higher performance and higher data bandwidth

- Advanced system software and applications
  - Co-design scheme with application developers
  - FUJITSU optimizing compilers for Post-K

Post-K will meet requirements & be valuable for science and industries