Performance of Elbrus processors for fast Fourier transform and computational materials science code

Alexey Timofeev
Denis Dergunov, Vladimir Stegailov
Contents

• Introduction of Elbrus processor
• Two libraries for fast Fourier transform
• Computational materials science code VASP
• Conclusions
Elbrus processor

Moscow Center of SPARC Technologies

Institute of Electronic Control Machines named after I.S. Brook

www.mcst.ru

www.ineum.ru

www.elbrus.ru

Alexey Timofeev - Performance of Elbrus processors for FFT and material science
Elbrus processor

Moscow Center of SPARC Technologies

Institute of Electronic Control Machines named after I.S. Brook

Applications server

Cluster 16 Tflops

Desktop computer

Alexey Timofeev - Performance of Elbrus processors for FFT and material science
Elbrus processor

Own development of
• BIOS
• Operating system
• Optimizing compilers from C, C++, Fortran, Java, C#, Javascript and other instruments for programming
• Mathematical Libraries
• Tools for servers and clusters

Parallel Energy Efficient Architecture
• 125 GFlops on 8 cores
• VLIW architecture
• High single-threaded performance
• General Purpose Processor

Effective binary compatibility with Intel x86, x86-64
• Execution of Windows XP, Windows 7 and above, Linux
• Compatibility layer for applications in x86 / x86-64 code in Linux
Elbrus processor

Own development of
- BIOS
- Operating system
- Optimizing compilers from C, C++, Fortran, Java, C#, Javascript and other instruments for programming
- Mathematical Libraries
- Tools for servers and clusters

Parallel Energy Efficient Architecture
- 125 GFlops on 8 cores
- VLIW architecture (as Intel Itanium IA-64)
- High single-threaded performance
- General Purpose Processor

Effective binary compatibility with Intel x86, x86-64
- Execution of Windows XP, Windows 7 and above, Linux
- Compatibility layer for applications in x86 / x86-64 code in Linux

Alexey Timofeev - Performance of Elbrus processors for FFT and material science
Two libraries for fast Fourier transform
FFT Libraries

FFT W

- The most popular library
- It is considered the fastest
- Release 3.3.5 of July 31, 2016.

Contents and features:
- Multidimensional Fourier Transforms;
- O (N logN) for any sizes of the incoming array;
- Parallelism (Posix, OpenMP, MPI)

EML
(Elbrus Mult. Library)

- Own development of the MCTS

Content:
- vector arithmetic;
- linear algebra;
- signal processing;
- image and video processing;
- 2-D and 3-D graphics.
Two stages of the FFT algorithm

Initialization

- once for a given array size

FFTW via `fftw_plan_dft(..)`,
EML via `eml_Signal_FFTInit(...).`

Execution

- one or many times

FFTW via `fftw_execute_dft(p,in,out),`
EML via `eml_Signal_FFTFwd(...).`
Intel Xeon E5-2660v4 (2.0 GHz) vs Elbrus-8S (1.3 GHz) for FFTW

Initialization

Execution (1000 launches)

Intel is three times faster

Processors are nearly equal

Alexey Timofeev - Performance of Elbrus processors for FFT and material science
EML (Elbrus Math library) vs FFTW on Elbrus-8S

- **Initialization**
  - EML initialization is ten times faster.

- **Execution**
  - EML is nearly equal to FFTW.
Computational materials science code VASP on Elbrus-8S
Computation materials science as supercomputer workload

- Example of statistics for the Edinburgh Parallel Computing Centre

### Code Usage on ARCHER (2014-15) by CPU Time:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Code</th>
<th>Node hours</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VASP</td>
<td>5,443,924</td>
<td>DFT</td>
</tr>
<tr>
<td>3</td>
<td>CP2K</td>
<td>2,121,237</td>
<td>DFT</td>
</tr>
<tr>
<td>4</td>
<td>CASTEP</td>
<td>1,564,080</td>
<td>DFT</td>
</tr>
<tr>
<td>9</td>
<td>LAMMPS</td>
<td>887,031</td>
<td>Classical</td>
</tr>
<tr>
<td>10</td>
<td>ONETEP</td>
<td>805,014</td>
<td>DFT</td>
</tr>
<tr>
<td>12</td>
<td>NAMD</td>
<td>516,851</td>
<td>Classical</td>
</tr>
<tr>
<td>20</td>
<td>DL_POLY</td>
<td>245,322</td>
<td>Classical</td>
</tr>
</tbody>
</table>

52% of all CPU time used by Chemistry / Materials Science / Biomolecular Simulation
The dependence of the time for the 1st iteration of the liquid Si model test in VASP on the number of cores per socket.

- Elbrus-8S
- E5-2697v3 (MVS1P5)
- E5-1650v3
Our way to compare “apples” to “oranges” - i.e. different architectures

• We use reduced parameter to compare different architectures – number of FLOP that could be executed during computational time

\[
\text{Theoretical peak performance (} R_{\text{peak}} \text{)} \times \text{iteration time (VASP)} \times (T_{\text{iter}}) = \text{Number of operations, that could possibly be done during iteration time (FLOP)}
\]

• We use balance parameter to characterize memory subsystem

\[
\text{Theoretical peak performance (} R_{\text{peak}} \text{)} \div \text{Memory bandwidth (Megabytes/sec)} = \text{Balance (FLOP/byte)}
\]
The dependence of the time for the first iteration of the liquid Si test on the number of cores per socket in the reduced parameters $R_{\text{peak}} \tau$ and balance $B$. 

![Graph showing the dependence of time for the first iteration of the liquid Si test on the number of cores per socket for different processors.](image)

$R_{\text{peak}}$, Time for iteration (GFlops)

Balance (Flops/B)
Conclusions

Hardware and software ecosystem of Russian Elbrus processors are mature enough for material science calculations that has been checked on the example of VASP.

Elbrus-8S shows larger time-to-solution values, however there is no large gap between performance of Elbrus-8S and Xeon Haswell CPUs.

A new metric that allows us to compare the processors of different architectures is proposed.

FFTW performance on Elbrus-8S is competitive with Intel Xeon Broadwell CPUs. EML on Elbrus-8S (1.3GHz) appears to be close or even more effective than FFTW on Intel Xeon E5-2660v4 (2.2 GHz).