High-Performance MPI Application Energy Consumption Profiling



<u>Victor Getmanskiy</u> Oleg Shapovalov Efim Sergeev Dmitry Kryzhanovsky

Singularis Lab, Ltd.

Volgograd State Technical University

Outline

- 1. Power management in Processors
- 2. Energy consumption profiling
- 3. Packages and benchmarks
- 4. Difficulties and reefs
- 5. How to profile: methodology
- 6. Analysis results
- 7. Conclusions
- 8. Common results

Energy-efficient supercomputers



20-th place Dell C8220X Cluster, Intel Xeon E5-2680v2 10C 2.8GHz, InfiniBand 4x FDR, Intel Xeon Phi 7120P **1-st place** ASUS ESC4000 FDR/G2S, Intel Xeon E5-2690v2 10C 3GHz, Infiniband FDR, AMD FirePro S9150

2.39 GFlops/W

Intel® Xeon Phi™

5.27 GFlops/W

Scope of power analysis

- Hardware analysis
 - analysis of computational system consumption
 - no analysis of single application consumption
- Software analysis at application level (Power Top)
 - analysis of single application consumption
 - detecting the consumption sources (IO, memory, HDD, CPU etc.)
- Software analysis at function level (Intel[®] VTune Amplifier)
 - analysis of consumption of single functions and units
 - detecting the functions (methods) requiring energy consumption optimization by the developer

C-states of Intel[®] Xeon Phi[™]

CPU **C-states** are core power states requested by the Operating System Directed Power Management (OSPM) infrastructure. C1-Cn states describe states where the processor clock is inactive and different parts of the processor are powered down.

| C0 | Full on |
|--------------|-----------------|
| C1/C1E | Auto-halt |
| C3 Auto/Deep | Sleep |
| C6 | Deep Power Down |

C-states of Intel[®] Xeon Phi[™]

Processor C States

- C-States ensure lower processor power during idle light workloads
- C-State limits can be set by BIOS
- A processor can go into sleep states several thousand times per second
- OS controls the C states in its idle process

| | Active state | Sle | ep states | | |
|-----------------------|-----------------|-------------------------|------------------------|------------------------|------------------------|
| | <u>C0</u> | <u>C1/C1E</u> <u>C3</u> | | <u>C6</u> | <u>C7</u> |
| | Operating | Halt | Sleep | Deep S | ileep |
| Core clock | N | off | off | off | off |
| PLL | JU | JU | off | off | off |
| Core caches | | | fl <mark>ushe</mark> d | fl <mark>ushe</mark> d | fl <mark>ushe</mark> d |
| Shared cache | | | | | fl <mark>ushe</mark> d |
| Wakeup time* | active | | | \bigcirc | \bigcirc |
| Core Idle power* | | | | ~ 0 | < C6 |
| * Rough approximation | | | | | |

P-states and C-states

C-states – Processor Power States P-states – Processor Performance Power States P-states apply only to C0 state of processor and control voltage and clock frequency

Processor Performance Power States (P-States)

| PO | Processor consumes max power and is at max performance. Additional performance increase with Intel® Turbo Boost Technology |
|----|--|
| P1 | Processor consumes less power and performance capabilities are limited below max. |
| Pn | Performance is at minimal level and lowest power consumption. n must not exceed 16. |

ACPI Spec Rev. 4.0a Defined

Multiple voltage and frequency operating points

- Software controlled by writing to MSRs
- The voltage is optimized based on the selected frequency and number of active processor cores
- All active processor cores share the same frequency and voltage.

Number of supported states is processor dependent

Power states summary

| | P-state | C-state |
|-----------|---------|---------|
| Processor | + (P) | + (PC) |
| Core | - | + (C) |
| Logical | - | + (LC) |

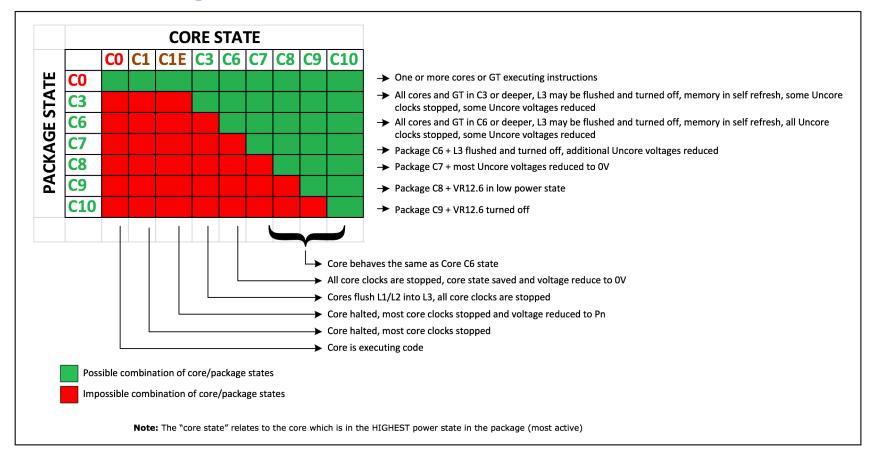
Processor C-state = Min(core C-states)

Core C-state = Minimum barrier(set of all logical C-states)

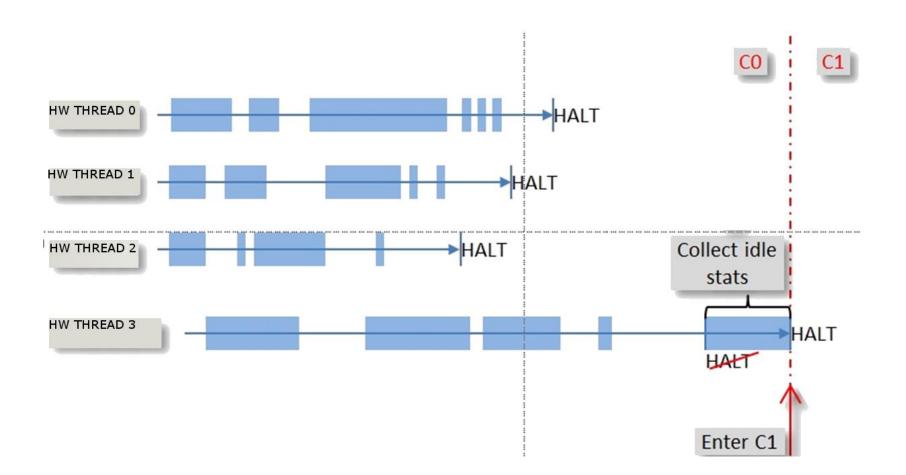
Logical C-state = anything the OS wants (hardware threads "states" on Intel Xeon Phi)

C-states (Haswell Mobile Processor)

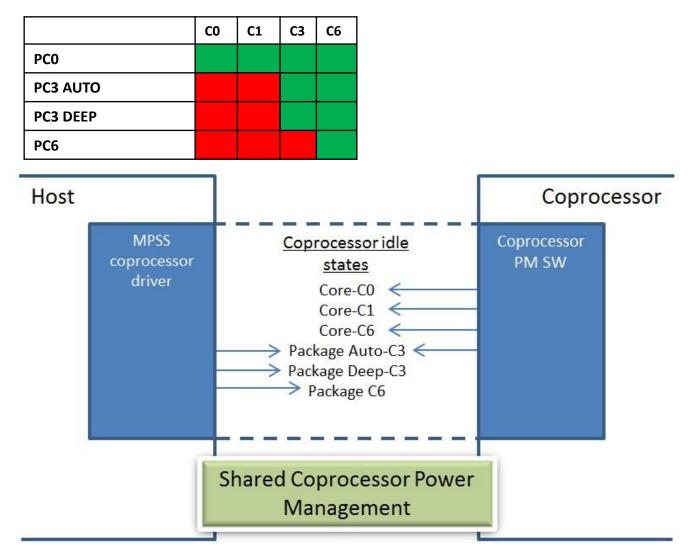
Processor Package and Core C-States



Switching to Package IDLE state



C-states (Intel[®] Xeon Phi[™])



Intel[®] Xeon Phi[™] C-states In-Depth

| Package Idle State | Core State | Uncore State | TSC/LAPIC | C3WakeupTi mer | PCI Express* Traffic |
|---|------------|--------------|-----------|---|-------------------------|
| Auto C3 (initiated by PM SW can be overriden by MPSS) | Preserved | Preserved | Frozen | On expiration, package exits PC3 | Package exits PC3 |
| Deep C3 | Preserved | Preserved | Frozen | No effect | Times out |
| PC6 | Lost | Lost | Reset | No effect | Times out |

Package Auto-C3: Ring and Uncore clock gated Package Deep-C3: VccP reduced Package C6: VccP is off (I.e. Cores, Ring and Uncore are powered down)

Software tools

| | OS | CPU models | CPU consumption | Consumption of the other devices | Linkage with the code |
|---------------------------------------|-------------------------|------------------------------------|-----------------------|--|-------------------------------------|
| Power Top | Linux, Solaris | Core2Duo | Modelled | Modelled | No |
| Joulemeter | Windows | Core2Duo | Modelled | Modelled | No |
| Intel Power Gadget | Windows, Linux, OS X | Core-i, XEON (Sandy Bridge) | Direct measurement | _ | No |
| XCode 5 Power Profiler | OS X | Core-i, XEON (Sandy Bridge) | Combined indices | _ | Statistics for single threads |
| Intel [®] VTune Amplifier | Windows, Linux | Core-i, XEON (Sandy Bridge) | Direct measurement | _ | Yes |

Intel[®] VTune Amplifier is the most comfortable tool for developers

Packages and benchmarks

Open source and popular in industry HPC applications:

- **GAMESS** computational quantum chemistry.
- **GROMACS** molecular dynamics for modelling physicochemical processes.
- LAMMPS simulation of the classical molecular dynamics of particle system.
- NAMD an object oriented code for modelling molecular dynamics of big biomolecular systems.
- **OpenFOAM** computational hydrodynamics and work with fields (scalar, vector, tensor).

Difficulties and reefs

- 1. Building the packages with various implementations of MPI.
- 2. Need to balance between the size of task (sample), the size of application profile and its level of detail.
- Problems with detecting units under dynamic linking and detecting functions (methods) inside units.

Difficulties and reefs

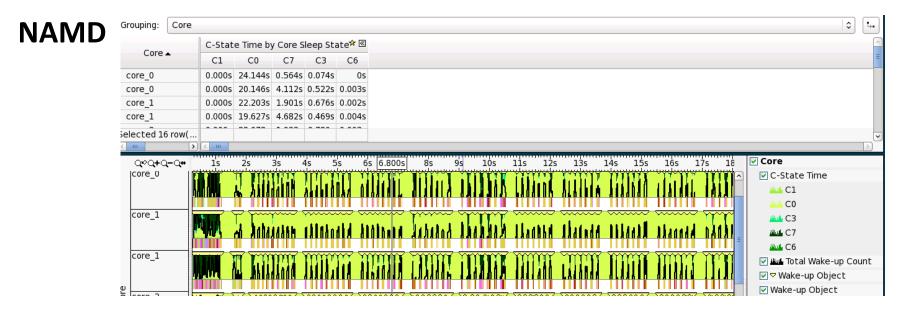
- 5. Collecting PMU events is limited with only one profiling process.
- Detecting the mode of profiling for one and all processes and the necessity in analysis of MPI scheduler energy consumption.
- 7. Adjusting hardware for maximum scalability of the applications (NUMA, hyper-threading).

Difficulties: dynamic MPI linking

Advanced Hotspots Hardware Event Counts viewpoint (change) ③

| d 😁 Analysis Target 🙏 Analysis Type 📓 Summary 🔗 PMU Events | 🗳 Caller/Callee 🏼 🍕 Top-down Tree | 🗄 Tasks and Frames | | | |
|--|-----------------------------------|---|-------------|--|--|
| Grouping: Module / Function / Call Stack | | | | | |
| Madda (Duration (Call Charle | Hardware Ev | Hardware Event Count by Hardware Event Type | | | |
| Module / Function / Call Stack | Energy Core- | Energy Pack | Energy DRAM | | |
| ▶namd2 | 13,262,674,064 | 16,642,949,488 | 469,946,38 | | |
| Þ[Unknown] | 417,066,448 | 522,906,672 | 14,126,76 | | |
| Þvmlinux (_) | 300,799,840 | 386,657,728 | 11,820,30 | | |
| ▽libmpi.so.4.1 | 270,844,064 | 339,839,936 | 9,358,56 | | |
| ▶ func@0xd0150 | 175,018,896 | 219,609,904 | 6,047,60 | | |
| ▶MPIDI_CH3I_Progress | 80,318,144 | 100,763,760 | 2,787,32 | | |
| I_MPIintel_ssse3_rep_memcpy | 4,101,904 | 5,150,928 | 133,92 | | |
| ♦MPIDI_CH3_iSendv | 2,053,728 | 2,580,160 | 69,95 | | |
| ♦ MPID_nem_tcp_connpoll | 1,631,216 | 2,047,744 | 52,44 | | |
| ▷ MPI_lprobe | 1,584,016 | 1,985,088 | 53,63 | | |
| ♦ MPID_nem_tcp_poll | 1,119,296 | 1,402,960 | 38,83 | | |
| ▷MPID_nem_network_poll | 630,288 | 793,632 | 24,67 | | |
| ♦ MPIDI_nem_active_vc | 580,944 | 730,560 | 22,27 | | |
| PMPI_Wtime | 523,632 | 657,680 | 16,06 | | |
| ♦ MPID_nem_tcp_vc_active | 473,712 | 592,784 | 17,92 | | |
| ♦ MPID_lprobe | 467,712 | 586,832 | 15,90 | | |
| ∮func@0x2b5df0 | 286,464 | 360,208 | 8,51 | | |
| ♦MPIDI_CH3U_Recvq_FU | 232,224 | 290,960 | 6,40 | | |
| ♦ MPID_Isend | 222,320 | 282,032 | 8,00 | | |
| ▷MPID_nem_tcp_iStartContigMsg | 175,136 | 219,952 | 6,08 | | |
| PMPI_Recv | 120,752 | 150,528 | 5,88 | | |
| ♦ MPID_Wtime_todouble | 119,456 | 149,536 | 4,09 | | |
| ▷MPIR_Test_impl | 118,016 | 147,856 | 3,96 | | |
| ▷MPI_lsend | 117,744 | 147,056 | 6,27 | | |
| ▷MPID_Recv | 111,248 | 141,584 | 3,77 | | |
| ∮func@0x205ba0 | 63,024 | 78,784 | 1,60 | | |
| ▷MPID_nem_tcp_iSendContig | 62,960 | 78,720 | 1,53 | | |
| PMPI_Get_count | 62,272 | 77,184 | 2,11 | | |

Difficulties: profiling mode for all / one process

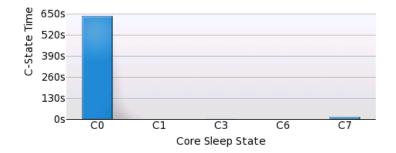


LAMMPS

 \bigcirc

Elapsed Time per Core Sleep State Histogram 🗈

This histogram represents a breakdown of the Elapsed Time per Core Sleep State over all cores.



Difficulties: profiling mode for all / one process

| | | | Caller/Callee 🗣 To | p-down Tree | |
|--|----------------------------|---------------|--------------------|-------------|-----------------|
| Grouping: Package / H/W Cont | ext / Function / Call Stac | k | | | ≎ ⊑. |
| Package / H/W Context / | Hardware Even | t Count by Ha | rdware Event Type | | 5 |
| Function / Call Stack | Energy Core | Energy Pack | Energy DRAM | Module | Functio |
| ▼package_0 | 21,531,249,952 | 26,797,78 | 595,312,768 | } | |
| Þcpu_0 | 2,525,701,456 | 3,138,866, | 67,433,328 | 3 | |
| ¢cpu_5 | 2,714,464,656 | 3,379,069, | 75,423,920 |) | |
| ¢cpu_23 | 2,713,808,400 | 3,378,586, | 75,347,472 | 2 | |
| ¢cpu_20 | 2,717,485,696 | 3,382,720, | 75,422,784 | ŧ | |
| ¢cpu_19 | 2,713,301,264 | 3,377,333, | 75,386,624 | t I | |
| ¢cpu_22 | 2,713,173,008 | 3,377,259, | 75,405,232 | 2 | |
| ¢cpu_17 | 2,714,255,728 | 3,378,965, | 75,419,728 | 3 | |
| ¢cpu_2 | 2,633,659,280 | 3,275,806, | 71,787,424 | ŧ | |
| ¢cpu_18 | 84,475,184 | 107,835,680 | 3,623,130 | 5 | |
| ¢cpu_16 | 872,768 | 1,185,184 | 45,152 | 2 | |
| ¢cpu_1 | 52,512 | 155,440 | 15,728 | 3 | |
| Рсри_З | 0 | 0 | (|) | |
| Þcpu_4 | 0 | 0 | 2,240 |) | |
| ¢cpu_6 | 0 | 0 | (|) | |
| Þcpu_7 | 0 | 0 | (|) | |
| ¢cpu_21 | 0 | 0 | (|) | |
| Þpackage_1 | 21,071,926,512 | 26,283,35 | 604,383,312 | 2 | |
| Selected 1 row(s): | 21,531,249,952 | 26,797,78 | 595,312,768 | 3 | |
| × | | | III | 1 | |
| ્ય ્+ ્−્⇔ 5 | s 10s | 15s | 20s 25s | 30s | 35s 40s 45s 50s |
| Thread (0x20 Thread (0x20 Thread (0x20 Thread (0x20 Thread (0x20 Thread (0x20 Thread (0x20 | ontext Switches (0%) | | | | |

Difficulties: adjusting hardware (NUMA, hyberthreading)

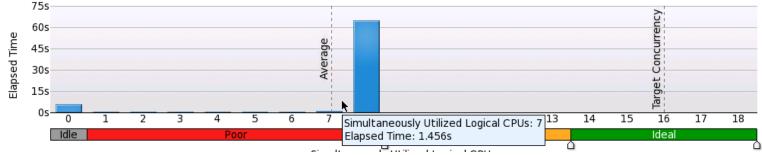
| CPU Power Management Configuration | | Enable the power management features. |
|---|--|---|
| EIST Turbo Mode C1E Support CPU C3 Report CPU C6 Report CPU C7 Report Package C State limit | [Enabled] [Enabled] [Enabled] [Enabled] [Enabled] [D6] | |
| Energy/Performance Bias Factory Long Duration Power Limit Long Duration Power Limit Factory Long Duration Maintained Long Duration Maintained Recommended Short Duration Power Limit Short Duration Power Limit | [Balanced Performance] 95 Watts 0 10 s 0 1.2 * Long Duration 0 | <pre>++: Select Screen f4: Select Item Enter: Select +/-: Change Opt. F1: General Help F2: Previous Values F3: Optimized Defaults F4: Save & Exit ESC: Exit</pre> |

Difficulties: adjusting hardware (NUMA, hyberthreading)

Without NUMA

📀 CPU Usage Histogram 🗈

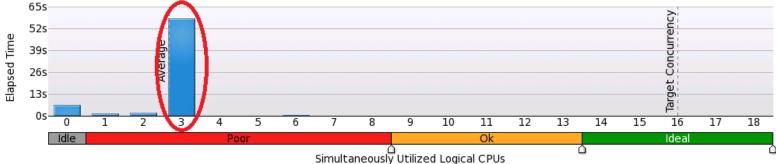
This histogram represents a breakdown of the Elapsed Time. It visualizes what percentage of the wall time the specific number of CPUs were running simultaneously. CPU Usage may be higher than the thread concurrency if a thread is executing code on a CPU while it is logically waiting.



With NUMA

📀 CPU Usage Histogram 🗈

This histogram represents a breakdown of the Elapsed Time. It visualizes what percentage of the wall time the specific number of CPUs were running simultaneously. CPU Usage may be higher than the thread concurrency if a thread is executing code on a CPU while it is logically waiting.



How to profile

Methodology:

- 1. Use static linking.
- 2. Profile all CPUs at node.
- 3. Find a relative part of the scheduler energy consumption.
- 4. Profile one of the processes being run.
- 5. Detect the MPI library functions in the application unit.
- 6. Estimate the energy consumption relative parts of the application functions and MPI library.

How to run analyses

- Profiling must be run in the mode <u>Advanced Hotspot Analysis</u>.
- To run a profiler GUI use amplxe-gui.

| | /home/testuser/in | tel/ampixe/projects/LAMMPS - II | ntel VTune Amp | lifier (на node30.cluster) | |
|-------------------|--|---|---|---|-------------------------------|
| Project Navigator | 💹 🕼 🖆 🖻 👦 🕨 i | 🗲 🕕 🛛 Welcome 🔹 New Am | × | | = |
| /home/testuser/ | 💹 Choose Analysis 1 | Гуре | | | Intel VTune Amplifier XE 2013 |
| VIII I LAMMPS | d Å Analysis Type | | | | Þ |
| _ | Analysis Type Analysis Type Analysis Type Algorithm Analysis Basic Hotspots Advanced Hotspots Advanced Hotspots Aconcurrency Aconcurrency Locks and Waits Intel Core 2 Processor. Nehalem / Westmere A Sandy Bridge / Ivy Bridge Access Contention Bandwidth Access Contention Branch Analysis Core Port Saturation Cycles and uOps Memory Access Port Saturation Singhts Corner Platform Power Analysis Custom Analysis Custom Analysis Custom Analysis Custom Analysis CPU Sleep States 0 | Advanced Hotspots Identify time-consuming code in y (formerly, Lightweight Hotspots) analysis by collecting call stacks, a and analyzing the CPI (Cycles Per analysis uses higher frequency said CPU sampling interval, ms: -Select a level of details provided Hotspots Hotspots, stacks and context Hotspots, call counts, stacks a Event mode: All Analyze user tasks So Details Events configured for CPU: Inter NOTE: For analysis purposes, Int the Sample After values in the depends on the value of the Du the Droiect Dreparties diplor Event Name CPU_CLK_UNHALTED.REF_TSC | uses the kemel d context switch a Instruction) met mpling at lower or 1 with event-base switches and context switches el(R) Xeon(R) E5 el VTune Amplifie table below by a ration time estin Sample After 2200000 | Ariver and extends the hotspot Ind statistical call count data tric. At the default level this verhead compared to Bas ad sampling collection. Detailed ches processor er XE 2013 may adjust multiplier. The multiplier nate option specified in LBR Filter Reference cycl | Project Properties |
| | | CPU_CLK_UNHALTED.THREAD | 2200000 | | |
| | | INST_RETIRED.ANY | 2200000 | Instructions re | |
| | < III > | | | | Command Line |

How to see the command line

- If you click the button Command Line..., you will see the command to run a console profiler.
- If you reset the check-box Hide knobs default values, the default keys will be added to the command line.

| Copy Command Line to Clipboard | × |
|---|---------------|
| Command line: | |
| /usr/local/intel/vtune_amplifier_xe_2013/bin64/amplxe-cl -collect advanced-hotspots knob sampling-interval=1 -knob collection-detail=stack-sampling -knob event-mode knob enable-user-tasks=false -follow-child -mrte-mode=auto -target-duration- type=short -no-allow-multiple-runs -no-analyze-system -data-limit=500 -slow-fram threshold=40 -fast-frames-threshold=100 -app-working-dir /var/local/INTEL/src/ lammps-25Nov13/bench /var/local/INTEL/src/lammps-25Nov13/bench/lmp_impiMKL var x 2 -var y 2 -var z 2 <./in.lj | =all · es- |
| Сору | |
| □ Use -collect-with action | |
| □ Hide knobs with default values | |
| This command line can be used to collect data on a remote machine. To do this, pres the Copy button to copy the command line to the clipboard, and then run the command line remotely, copy the result directory back to the local host, and open t result file in the Amplifier XE. | |
| Help Close | |

Adjusting the environment

.bashrc

.bashrc.namd_IntelMPI

| module unload MPI-3/MVAPICH2 | |
|---|----------|
| module unload MPI-3/OpenMPI | |
| module unload MPI-3/MPICH | |
| module load MPI-3/IntelMPI | |
| export PATH=/var/local/INTEL/install/intelmpi-icc/third- | |
| <pre>party/src/NAMD_2.9_Source/Linux-x86_64-ics-2013-SP1:\$PATH</pre> | |
| export PATH=/var/local/INTEL/src/NAMD 2.9 Source/charm-6.4.0/bi | n:\$PATH |

How to run the profiling

```
#!/bin/bash
DIR=$ (pwd)
WORKING DIR=$DIR/NAMD/small
# remove profile output folder
$DIR/RemoveDir.sh $PROFILE DIR
# rewrite bashrc
(cat $DIR/.bashrc) > ~/.bashrc
(echo source $DIR/.bashrc.namd IntelMPI) >> ~/.bashrc
source ~/.bashrc
# run
cd $WORKING DIR
mpirun -genv I MPI FABRICS=shm:tcp -genv I MPI WAIT MODE=1 \
-host node31 -n 1 -wdir=$WORKING DIR amplxe-cl -collect advanced-hotspots -knob
collection-detail=stack-sampling -r $PROFILE DIR - namd2 apoa1 : \
-host node31 -n 15 -wdir=$WORKING DIR namd2 apoa1 : \
-host node30 -n 16 -wdir=$WORKING DIR namd2 apoa1
```

Some notes about running

- The folder for collecting output data specified with the key –r is added with the rank number (for example, .0). So, if we specify
 - -r ~/profiles/LAMMPS

we'll have the folder

~/profile/LAMMPS.0

- The profiling output results should be watched via GUI.
- The profiling output data folder must be cleaned before each launching VTune Amplifier.

Power consumption metrics

- Energy Core, μJ energy dissipated on the core
- Energy Pack, μJ energy dissipated on the processor
- Energy DRAM, μJ energy dissipated on the memory unit
- You need to group the functions to detect which of them consume more energy (it is to be the package being run itself)

| 🛛 🖶 Analysis Target 🖄 Analysis Type 🕅 Summary 🤞 | PMU Events 🛛 🖓 Caller/Calle | e 💊 Top-down Tree 🔣 Ta | asks and Frames | | | | |
|---|---|------------------------|-----------------|-------------|--|--|--|
| Grouping: Module / Function / Call Stack | | | | | | | |
| Medule (Exection (Coll Stock | Hardware Event Count by Hardware Event Type | | | | | | |
| Module / Function / Call Stack | Energy Core- | Energy Pack | Energy DRAM | Module | | | |
| Imp_impiMKL | 12,980,515,552 | 16,394,977,472 | 373,273,584 | | | | |
| ▷LAMMPS_NS::PairLJCut::compute | 7,092,074,544 | 9,005,436,656 | 202,162,704 | lmp_impiMKL | | | |
| LAMMPS_NS::Neighbor::half_bin_newton | 2,226,706,512 | 2,801,350,288 | 70,635,696 | lmp_impiMKL | | | |
| LAMMPS_NS::Comm::reverse_comm | 1,328,104,464 | 1,660,942,272 | 36,841,392 | Imp_impiMKL | | | |
| LAMMPS_NS::Comm::forward_comm | 990,692,832 | 1,231,359,616 | 33,247,184 | lmp_impiMKL | | | |
| ▷LAMMPS_NS::Comm::borders | 445,019,248 | 555,523,424 | 13,378,176 | Imp_impiMKL | | | |
| ▷LAMMPS_NS::Verlet::run | 191,455,376 | 239,864,272 | 5,285,120 | lmp_impiMKL | | | |
| ▷LAMMPS_NS::FixNVE::initial_integrate | 127,063,952 | 161,298,624 | 1,158,240 | lmp_impiMKL | | | |
| LAMMPS_NS::Verlet::force_clear | 88,995,872 | 115,051,120 | 611,088 | lmp_impiMKL | | | |
| ▷LAMMPS_NS::FixNVE::final_integrate | 83,676,672 | 106,114,064 | 1,012,416 | lmp_impiMKL | | | |
| ▷LAMMPS_NS::Neighbor::check_distance | 62,005,232 | 78,737,056 | 834,176 | lmp_impiMKL | | | |
| ▶LAMMPS_NS::Run::command | 60,336,176 | 75,433,648 | 1,531,296 | lmp_impiMKL | | | |
| ▶LAMMPS_NS::Pair::sbmask | 55,547,456 | 70,537,424 | 1,601,600 | lmp_impiMKL | | | |
| ▷LAMMPS_NS::AtomVecAtomic::unpack_reverse | 55,506,720 | 70,200,784 | 1,472,528 | lmp_impiMKL | | | |
| LAMMPS NS::Neighbor::build | 28,384,688 | 37,128,016 | 284,992 | Imp impiMKL | | | |

Analysis of power consumption

- You need to switch on the mode <u>Hardware Events Counts</u> <u>Viewpoint</u>
- The tab Summary displays the general power consumption for the period of time while we were profiling the application

| Ad | Advanced Hotspots Hardware Event Counts viewpoint (<u>change</u>) | | | | | | |
|-------------------------|---|----------------------|--------------------------|------------|----------------|--|--|
| ۲ | Analysis Target 🗚 Analysis Type 🛅 | Summary 🚱 PMU Even | ts 🔹 Caller/Callee 🗳 Top | -down Tree | 🔁 Tasks and Fr | | |
| $\overline{\mathbf{o}}$ | Elapsed Time: [®] 43.736 | S 🗎 | | | | | |
| | CPU Time: [®] 19.479 | 9s | | | | | |
| | Paused Time: 💿 🛛 🔘 |)s | | | | | |
| $\overline{\mathbf{O}}$ | Hardware Events 🗈 | | | | | | |
| | Hardware Event Type | Hardware Event Count | Hardware Event Sample Co | ount Even | ts Per Sample | | |
| | CPU_CLK_UNHALTED.REF_TSC | 42,853,836,454 | 19 | ,453 | 2200000 | | |
| | CPU_CLK_UNHALTED.THREAD | 52,519,525,562 | 19 | ,449 | 2200000 | | |
| | Energy Core | 1,225,176,768 | 19 | ,025 | 0 | | |
| | Energy DRAM | 40,651,776 | 18 | ,905 | 0 | | |
| | Energy Pack | 1,516,926,160 | 19 | ,027 | 0 | | |
| | INST_RETIRED.ANY | 98,222,584,818 | 19 | ,610 | 2200000 | | |
| | Inactive Time | 23,211,774,770 | 6 | ,332 | 0 | | |
| | Preemption Context Switches | 6,333 | 6 | ,332 | 0 | | |
| | Synchronization Context Switches | 69 | | 69 | 0 | | |
| | Wait Time | 36,803,932 | | 69 | 0 | | |

Analysis of power consumption

- To see individual consumption for groups of functions and units you can use the tab PMU Events
- You can use for grouping three parameters (Grouping:Module, Grouping:Function, Grouping:CallStack)
- If you expand the units, the most consuming functions will be displayed at the top

| Advanced Hotspots Hardware Event Counts viewpoint (<u>change</u>) | | | | | | | |
|---|--------------------------|---------------|-------------|---------------|--|--|--|
| 🔄 🖶 Analysis Target 🛝 Analysis Type 🛍 Summary 🔗 PMU Events 🍕 Caller/Callee 🚱 Top-down Tree 🛃 Tasks and Frames | | | | | | | |
| Grouping: Module / Function / Call St | ack | | | | | | |
| Hardware Event Count by Hardware Event Type | | | | | | | |
| Module / Function / Call Stack | Energy Core v | Energy Pack | Energy DRAM | Module | | | |
| Þnamd2 | 1,155,887,840 | 1,430,962,288 | 38,463,696 | | | | |
| Þ vmlinux | 25,460,512 | 31,622,592 | 799,616 | | | | |
| Þigb | 22,929,840 | 28,363,488 | 706,176 | | | | |
| Þbridge | 18,099,840 | 22,398,352 | 577,312 | | | | |
| ⊽libmpi.so.4.1 | 1,237,472 | 1,526,320 | 37,280 | | | | |
| ▶MPIDI_CH3_iSendv | 1,108,000 | 1,367,024 | 33,184 | libmpi.so.4.1 | | | |
| ▷MPID_nem_tcp_iSendContig | 65,040 | 79,792 | 2,112 | libmpi.so.4.1 | | | |
| ▷MPIDI_CH3_EagerContigIsend | 64,432 | 79,504 | 1,984 | libmpi.so.4.1 | | | |
| ▷[Import thunk writev] | 0 | 0 | 0 | libmpi.so.4.1 | | | |

Watching the collected data

| | | <no current="" p<="" th=""><th>roject> - Intel V</th><th>Tune Amplifier (на node31.cluste</th><th>r)</th><th></th><th></th><th></th><th></th></no> | roject> - Intel V | Tune Amplifier (на node31.cluste | r) | | | | |
|---------------------------------|------------------|---|--|--|---------|-----|-----|--|-----|
| 💹 🕼 🖆 🗃 😥 🕨 🖨 | Welcome | NAMD_p | × | | | | | | |
| Advanced Hotspot | s Hardwar | | | nt (<u>change</u>) ⑦ | | | | | |
| | | mary 🚷 PMU E | vents 💊 Caller/ | Calle Select viewpoint: Hardware Event Counts | nd Fram | es | | | |
| Grouping: Module / Function / C | Call Stack | | | Hardware Event Sample Counts | | | | | ↓ ↓ |
| Madula (Evention (Coll Stands | Hardware Even | t Count by Hard | ware Event Type | Hardware Issues | Sou. | Sta | | | |
| Module / Function / Call Stack | Energy Core | Energy Pack | Energy DRAM | Mod Hotspots | File | Add | | | |
| Þ[Unknown] | 1,258,016 | 1,618,496 | 49,984 | Hotspots | | 0 | | | |
| Þbridge | 23,349,824 | 28,955,712 | 754,192 | | | 0 | | | |
| Þext4 | 186,336 | 230,592 | 6,592 | | | 0 | | | |
| Þigb | 28,265,056 | 35,063,488 | 908,176 | | | 0 | | | |
| Þipv6 | 0 | 0 | 0 | | | 0 | | | |
| Þld-2.12.so | 12,848 | 27,552 | 2,240 | | | 0 | | | |
| Þlibc-2.12.so | 119,968 | 149,936 | 4,096 | | | 0 | | | |
| ▶libpthread-2.12.so | 0 | 0 | 0 | | | 0 | | | |
| Þnamd2 | 1,091,474,032 | 1,351,496,528 | 36,065,280 | | | 0 | | | |
| Þnfs | 165,264 | 209,040 | 6,784 | | | 0 | | | |
| Þsunrpc | 0 | 0 | 0 | | | 0 | | | |
| Þvmlinux | 26,536,288 | 32,927,792 | 824,848 | | | 0 | | | |
| Selected 1 row(s): | | 1,618,496 | 49,984 | ш | | | | | |
| Q≈ Q+ Q−Q+ | 5s | 10s | 15s | 20s | 25s | | 30s | 35s | 40s |
| Thread (0x54 Thread (0x54 | | a dar ar indini sakaka di babisi sa yasa da m | - handi si litik da bin da aran da ara | n dan bili dan sul annu kang mili yan shakela na sanuk dasha dinya ka an ana an akan ang | | | | Ů, Ů,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, | |

Sorting by energy consumption

| Advanced Hotspots Hardware Event Counts viewpoint (<u>change</u>) ③ ④ Analysis Target A Analysis Type B Summary & PMU Events Caller/Callee Caller/Callee Counts Type B Summary | | | | | | | |
|---|---|---------------|---------------|-------------|--|--|--|
| Grouping: Module / Function / Call Stack | | | | | | | |
| Mandada / Eventsia | Hardware Event Count by Hardware Event Type | | | | | | |
| Module / Functio | on / Call Stack | Energy Core | Energy Pack | Energy DRAM | | | |
| ▽ namd2 | | 1,091,474,032 | 1,351,496,528 | 36,065,280 | | | |
| ▷ComputeNonbondedUtil::cale | c_pair_energy | 218,274,416 | 269,682,176 | 7,351,360 | | | |
| ▷ComputeNonbondedUtil::cal | _pair_energy_fullelect | 120,518,640 | 149,407,728 | 3,799,456 | | | |
| ♦ MPID_nem_mpich2_test_rec | V | 114,659,056 | 142,278,064 | 3,866,752 | | | |
| ▷pairlist_from_pairlist | | 100,898,512 | 124,588,464 | 3,280,816 | | | |
| ▷ComputeNonbondedUtil::cal | c_self_energy | 97,986,240 | 120,904,912 | 3,174,464 | | | |
| ▷ScriptTcl::run | | 62,542,080 | 77,539,312 | 2,030,240 | | | |
| ♦ MPIDI_CH3I_Progress | | 54,285,056 | 67,308,768 | 1,862,480 | | | |
| ♦ ComputeNonbondedUtil::cal | c_self_energy_fullelect | 49,547,872 | 61,178,608 | 1,617,136 | | | |
| ▷CsdScheduler | | 28,025,920 | 34,720,320 | 907,536 | | | |
| Parameters::assign_dihedral | index | 27,026,336 | 33,420,160 | 996,416 | | | |
| ▷MPID_lprobe | | 25,093,296 | 31,166,688 | 868,112 | | | |

General energy consumption

Advanced Hotspots Hardware Event Counts viewpoint (<u>change</u>) 🕐 🗌

🔄 \ominus Analysis Target 🛝 Analysis Type 🔋 🕄 Summary 💊 PMU Events 📣 Caller/Callee 🔩 Top-down Tree 🔛 Tasks and I

📀 Elapsed Time: 🛛 42.609s 🗈

| CPU Time: 💿 | 18.581s |
|----------------|---------|
| Paused Time: 💿 | 0s |

📀 Hardware Events 🗈

| Hardware Event Type | Hardware Event Count | Hardware Event Sample Count | Events Per Sample |
|----------------------------------|----------------------|-----------------------------|-------------------|
| CPU_CLK_UNHALTED.REF_TSC | 40,877,324,510 | 18,555 | 2200000 |
| CPU_CLK_UNHALTED.THREAD | 50,122,607,951 | 18,574 | 2200000 |
| Energy Core | 1,171,367,632 | 18,218 | 0 |
| Energy DRAM | 38,622,192 | 18,203 | 0 |
| Energy Pack | 1,450,679,136 | 18,220 | 0 |
| INST_RETIRED.ANY | 90,046,098,213 | 18,989 | 2200000 |
| Inactive Time | 18,768,191,932 | 6,769 | 0 |
| Preemption Context Switches | 6,770 | 6,769 | 0 |
| Synchronization Context Switches | 1,489 | 1,489 | 0 |
| Wait Time | 2,190,496,730 | 1,489 | 0 |

Analysis results

Intel MPI functions with the highest energy consumption

| Module / Function / Call Stack | Energy Core | Energy Pack | Energy DRAM | Energy Core | Energy Pack | Energy DRAM |
|--------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| MPID_nem_mpich2_test_recv | 114659056 | 142278064 | 3866752 | 53% | 53% | 54% |
| MPIDI_CH3I_Progress | 54285056 | 67308768 | 1862480 | 25% | 25% | 26% |
| MPID_Iprobe | 25093296 | 31166688 | 868112 | 12% | 12% | 12% |
| I_MPIintel_ssse3_rep_memcpy | 4829472 | 5981408 | 147552 | 2% | 2% | 2% |
| MPIDI_CH3_iSendv | 3424624 | 4246848 | 107248 | 2% | 2% | 2% |
| MPI_Iprobe | 3172560 | 3928784 | 1068 | 1% | 1% | 0% |
| MPID_nem_tcp_connpoll | 2458384 | 3044944 | 70 | 1% | 1% | 0% |
| MPID_nem_network_poll | 2080144 | 2577088 | 70368 | 1% | 1% | 1% |
| MPI_Recv | 1605376 | 1997280 | 54144 | 1% | 1% | 1% |

Energy consumption relative parts for the application functions and MPI library

| Application | Energy Core | Energy Pack | Energy DRAM | |
|----------------------------|-------------|-------------|-------------|--|
| namd2 | 1091474032 | 1351496528 | 36065280 | |
| Without MPI | 873696888 | 1081513344 | 28925454 | |
| | | | | |
| | Energy Core | Energy Pack | Energy DRAM | |
| Total Power | 1171367632 | 1450679136 | 6 38622192 | |
| | | | | |
| Relative Power Consumption | Energy Core | Energy Pack | Energy DRAM | |
| mpiexec | 0% | 0% | 0% | |
| MPI | 19% | 19% | 18% | |
| namd2 | 75% | 75% | 75% | |

LAMMPS

- Intel MPI the lowest consumption (absolute and relative values)
- MVAPICH the highest consumption

NAMD

- Worse scalability than for LAMMPS
- Parallel computing takes a more intensive data exchange
- <u>Possible conclusion</u>: if the package is a good scalable, then Intel MPI gives a better (lower) consumption; if the packages is a bad scalable, Intel MPI gives worse results.

GROMACS

- Lower MPI energy consumption than for LAMMPS and NAMD
- The function SENDRECV, taking the most energy consumption in GROMACS, requires less overheads and prevents CPU from switching C-state to C-7 (LAMPS and NAMD use SEND and RECV instead of SENDRECV)

GAMESS

- The relative part of MPI energy consumption is the lowest (< 1%)
- It is caused by the high part of computational code in compare with parallel processes synchronization

OpenFOAM

- Has the highest (and extremely high) MPI power consumption.
- Different MPI libraries show values around 50 %.
- In the scope of energy consumption optimization OpenFOAM requires a special attention.

- 1. Static MPI linking with one process profiling provides more correct results.
- 2. Running the package on several cluster nodes provides more correct and representative results.
- Running the package on several cluster nodes under one process profiling is possible for Intel MPI, MPICH, MVAPICH and Open MPI with some differences in the MPI scheduler commands.

- 5. Analyzing energy consumption on cluster, you need to take into account that CPU is in the state C0 the most time. Switching between the states happens only if intensive data exchange takes place (as it is in NAMD).
- Energy consumption of the application functions and MPI library depends on the used MPI implementation and the task you are solving (its sizes).

Common results

- The methodology how to profile applications for power consumption has been described and masterclasses on this topic have been created.
- The benchmark parameters for more representative power profiling have been adjusted.
- The distribution of energy consumption for different MPI implementations and for single MPI functions has been estimated for several HPC packages.

Thank you for your attention

www.singularis-lab.com



SINGULARIS LAB software development

MPI Application Energy Consumption