TECHNISCHE UNIVERSITÄT DRESDEN

Center for Information Services and High Performance Computing (ZIH)

eeClust

VampirTrace Extensions and Energy-Efficiency-Benchmarks

11.09.2012

Daniel Molka (daniel.molka@tu-dresden.de)

ZIH
Center for Information Services & High Performance Computing

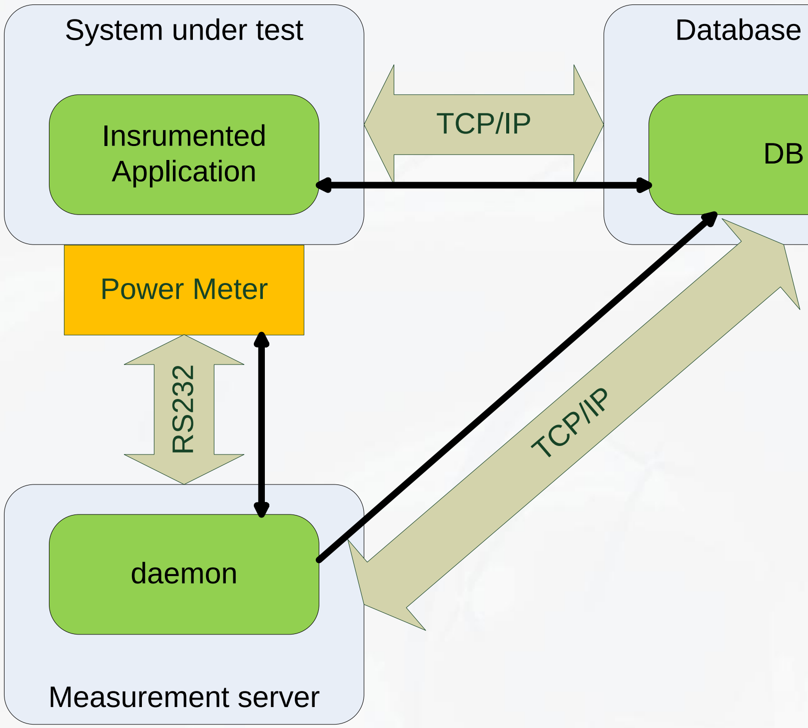
Outline

- VampirTrace Extensions
  - Uncore Performance Counter
  - Power Consumption Measurement
- Energy Efficiency benchmarks
  - eeMark
    - Kernels
    - Kernel Sequences
    - Results
  - Spec OMP2012 Energy Metric

Uncore Performance Counter

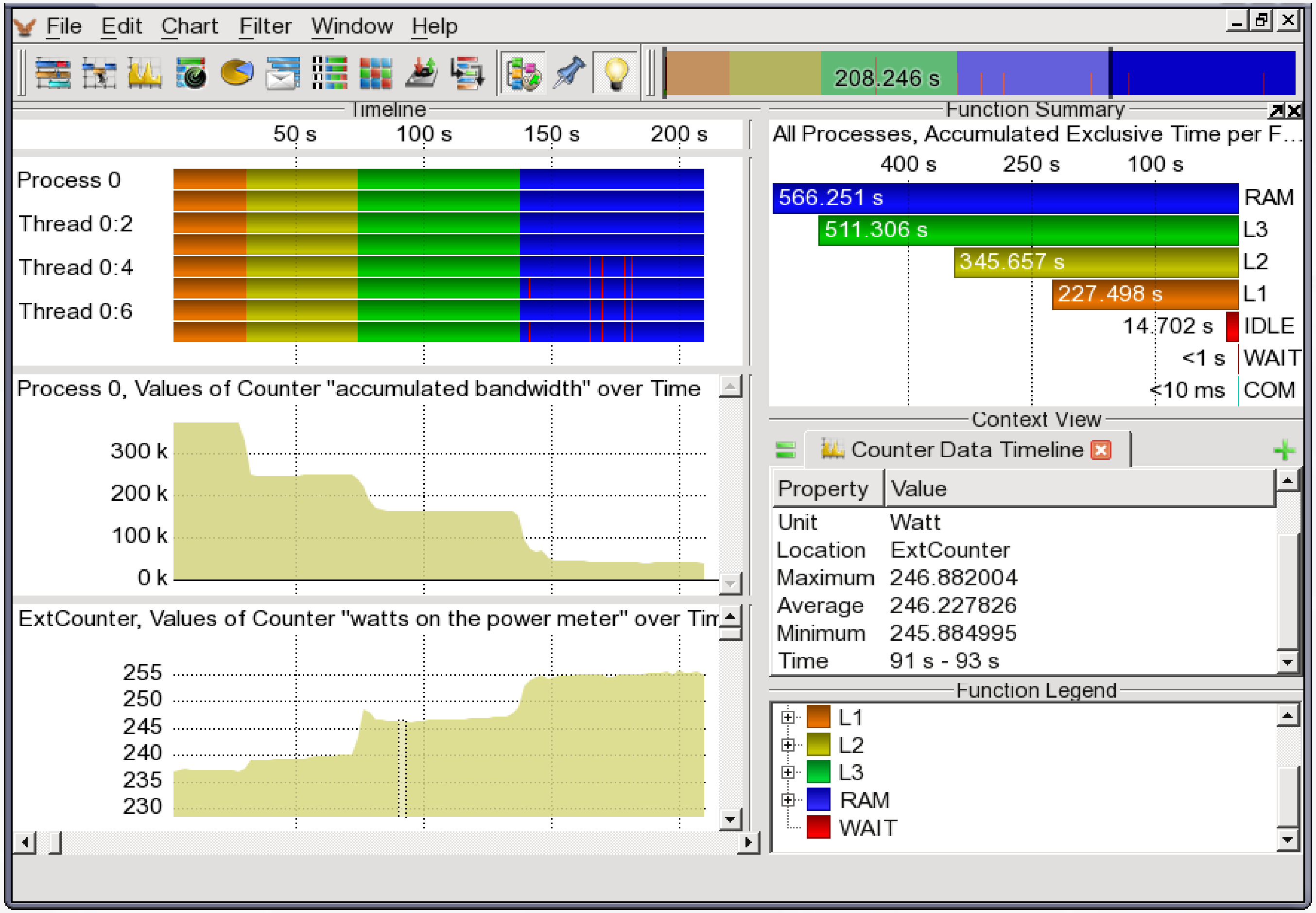
- Uncore Counters
  - Important to understand influence of shared resources
  - accessable via perfmon2
    - systemwide mode required
    - Can not be used concurrently with per-core measurements
- PAPI-C component for Uncore Counters
  - Shared resources not supported by standard PAPI distribution
  - Component records data via perfmon2 low-level API and provides interface for PAPI-C
  - VampirTrace reads data via PAPI interface

Power Measurement



- Additional servers reduce overhaed on system under test
- VampirTrace Plugin Counter to add information to trace files
  - PowerTracer support
  - Dataheap support

Power consumption of data transfers



- Microbenchmarks that stress individual cache levels
- Power consumption increases if more cache levels are used

Energy consumption of data transfers

Table IV
BANDWIDTH AND ENERGY CONSUMPTION OF DATA TRANSFERS (USING THE MOVAPS INSTRUCTION) FROM DIFFERENT MEMORY LOCATIONS (INTEL XEON X5670)

Location	P_{total}	$Bandwidth$	E_{trans}
L1	256.1 W	561.6 GB/s	64 pJ/Byte
L2	265.2 W	372.2 GB/s	121 pJ/Byte
L3	263.6 W	171.6 GB/s	254 pJ/Byte
RAM	269.9 W	39.9 GB/s	1250 pJ/Byte

- Estimation of average energy consumption per transferred Byte based on transfer rate of individual cache levels and associated power consumption
- Memory accesses consume an order of magnitude more power than cache accesses

Outline

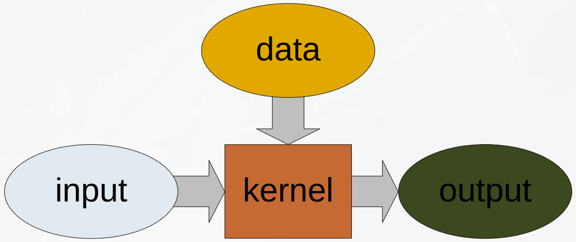
- VampirTrace Extensions
  - Uncore Performance Counter
  - Power Consumption Measurement
- Energy Efficiency benchmarks
  - eeMark
    - Kernels
    - Kernel Sequences
    - Results
  - Spec OMP2012 Energy Metric

eeMark: Motivation

- Varying power consumption of HPC systems
  - Depends on changing utilization of components over time (processors, memory, network, and storage)
  - Applications typically do not use all components to their capacity
  - Potential to conserve energy in underutilized components (DVFS, reduce link speed in network, etc.)
  - But power management can decrease performance
- HPC tailored energy efficiency benchmark needed
  - Evaluate power management effectiveness for different degrees of capacity utilization
  - Compare different systems

Benchmark Design - Kernels

- 3 types of kernels
  - Compute - create load on processors and memory
  - Communication - put pressure on network
  - I/O - stress storage system
- Same basic composition for all types of kernels
  - Three buffers available to each function
  - No guarantees about input other than
    - Data has the correct data type
    - No nan, zero, or infinite values
  - Kernel ensures that output satisfies these requirements as well
    - Buffer data initialized in a way that nan, zero, or infinite do not occur



Kernel Design - Compute Kernels

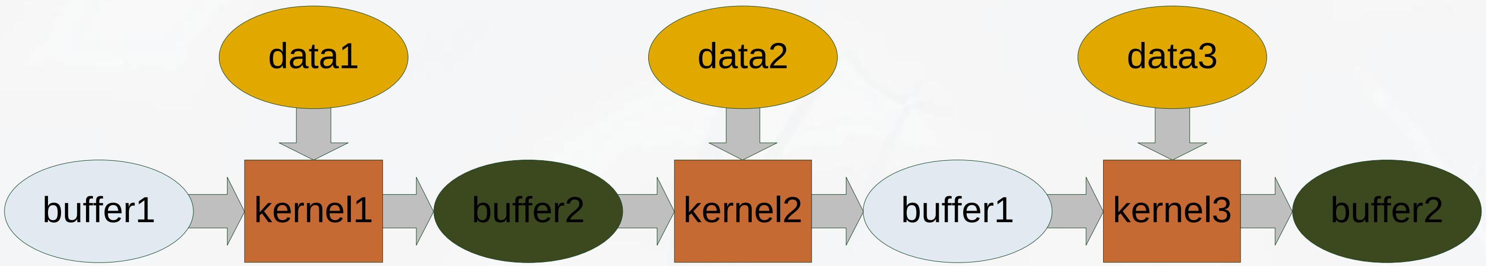
- Perform arithmetic operations on vectors
  - Double and single precision floating point
  - 32 and 64 Bit integer
- Written in C for easy portability
  - No architecture specific code (e.g. SSE or AVX intrinsics)
  - Usage of SIMD units depends on autovectorization by compiler
- Adjustable ratio between arithmetic operations and data transfers
  - Compute bound and memory bound versions of same kernel

Kernel Design - Communication and I/O Kernels

- MPI kernels
  - bcast/reduce involving all ranks
  - bcast/reduce involving one rank per group
  - bcast/reduce within a group
  - send/receive between groups
  - rotate within a group
- I/O kernels
  - POSIX I/O with one file per process
  - MPI I/O in with one file per group of processes

Benchmark Design - Kernel Sequences

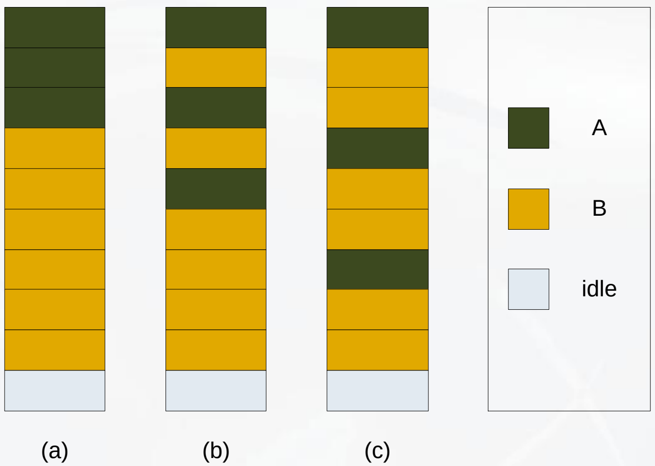
- 2 buffers per MPI process used as input and output
  - Output becomes input of next kernel
- data buffer per kernel



- Input and output used for communication and I/O as well
  - send(input), write(input): - send or store results
  - receive(output), read(output): - get input for next kernel

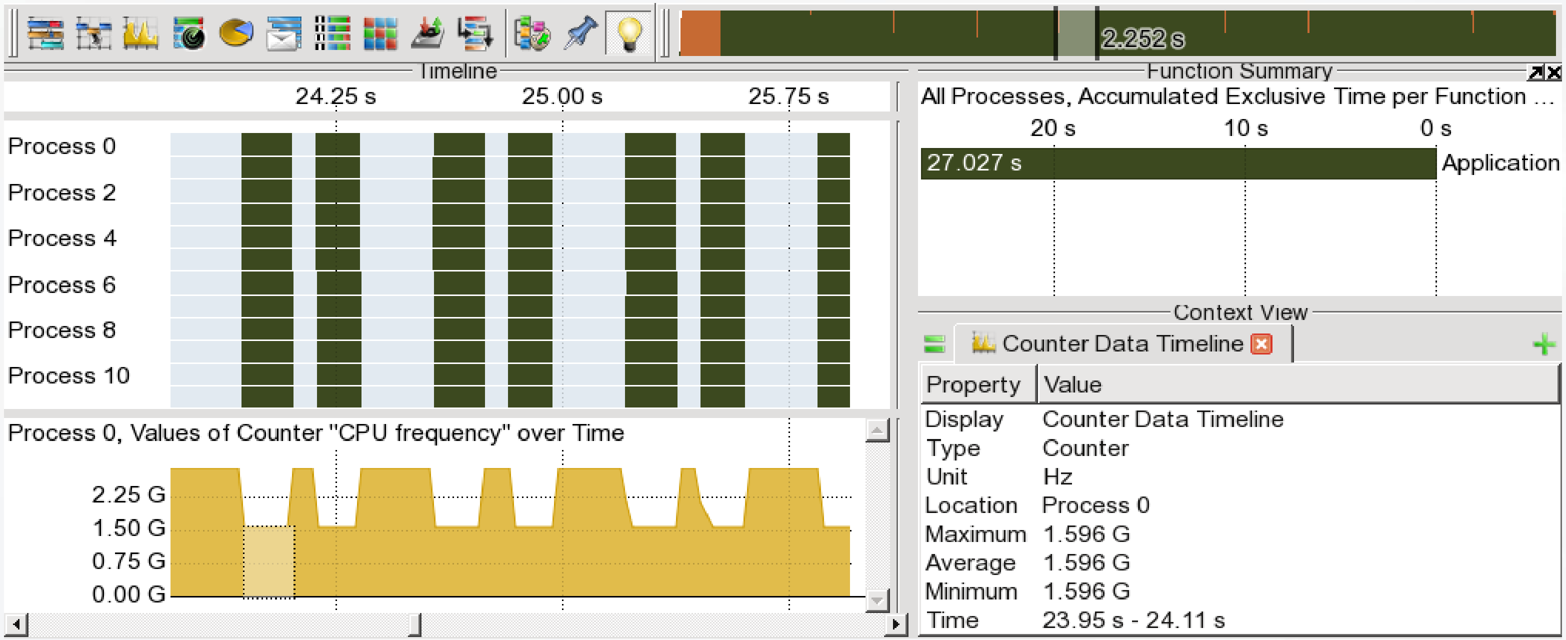
Profiles

- Define kernel sequences for groups of processes
  - Groups with dynamic size adopt to system size
    - E.g. half the available processes act as producers, the other half as consumers
    - Different group sizes possible
    - Multiple distribution patterns



  - Groups with fixed amount of processes for special purposes
    - E.g. a single master that distributes work
- Define the amount of data processed per kernel
- Define block size processed by every call of kernel

Example: Workload based Frequency Scaling



- Compute bound and memory bound phases in all processes
- Frequency dynamically adjusted based on performance counters

Power Measurement

- Use of existing measurement systems
  - PowerTracer, developed at University of Hamburg
  - Dataheap, developed at TU Dresden
  - SPEC power and temperature demon (ptd)
- Power consumption recorded at runtime
- API to collect data at end of benchmark
- Multiple power meters can be used to evaluate large systems

Benchmark Result

- Kernels return type and amount of performed operations
  - workload = weighted amount of operations
- Performance Score = workload / runtime
  - billion weighted operations per second
- Efficiency Score = workload / energy
  - billion weighted operations per Joule
- Combined Score = sqrt(perf_score*eff_score)

eeMark: Reference Result

Benchmark	Distribution	Iterations	Performance Score	Efficiency Score	Combined Score
compute1_dp	compact	1	2214.67	0.97	46.34
compute2_dp	compact	1	2264.79	0.74	40.95
compute3_dp	fine	1	2384.87	0.94	47.28
compute1_sp	compact	1	1125.11	0.50	23.76
compute2_sp	compact	1	2253.71	0.73	40.60
compute3_sp	fine	1	1334.56	0.56	27.24
compute1_int	compact	1	611.06	0.30	13.46
compute2_int	compact	1	2203.31	0.71	39.48
compute3_int	fine	1	834.24	0.38	17.74
comm1	fine	1	5429.13	1.70	96.13
comm1	compact	1	561.47	0.22	11.20
comm2	compact	1	878.34	0.33	16.96
comm3	fine	1	715.09	0.28	14.18
comm3	compact	1	177.12	0.07	3.58
comm3	roundrobin	1	438.70	0.18	8.81
io1_nompiio	compact	1	403.48	0.27	10.49
io2_nompiio	compact	1	298.44	0.20	7.77
io3_nompiio	compact	1	362.32	0.25	9.49
combined1_dp	fine	1	3996.43	1.33	72.89
combined1_dp	compact	1	1108.31	0.43	21.73
combined2_dp	compact	1	429.24	0.17	8.63
combined1_sp	fine	1	5158.59	1.65	92.24
combined1_sp	compact	1	1108.52	0.42	21.70
combined2_sp	compact	1	343.20	0.14	6.92
combined1_int	fine	1	1293.15	0.47	24.70
combined1_int	compact	1	864.61	0.33	16.95
combined2_int	compact	1	241.75	0.10	4.92
Result:			1445.71	0.53	27.63

Outline

- VampirTrace Extensions
  - Uncore Performance Counter
  - Power Consumption Measurement
- Energy Efficiency benchmarks
  - eeMark
    - Kernels
    - Kernel Sequences
    - Results
  - Spec OMP2012 Energy Metric

SPEC OMP2012

- Successor of SPEC OMP2001
  - Still under development
  - Currently final testing and bugfixing
- Performance and efficiency metric
  - Performance rating based on: runtime / reference runtime
  - New Energy rating based on: energy / reference energy
- ZIH provides reference machine

Summary

- Added functionality to VampirTrace
  - Uncore Performance Counter
  - Power Measurement
- Developed eeMark
  - HPC tailored synthetic energy efficiency benchmark
- Contributed to SPEC OMP2012 development
  - New energy efficiency rating

Thank you

- Further Information at eeClust homepage
  - www.eeClust.de

eeClust

Federal Ministry of Education and Research