CATEGORY: SUPERCOMPUTING - SC10
POSTER P5302
CONTACT NAME Kazuki Tsuzuku: tsuzuku.k@gmail.com

GPU TECHNOLOGY CONFERENCE

Power Capping of CPU-GPU Heterogeneous Systems using Power and Performance Models

Kazuki Tsuzuku, Toshio Endo
Tokyo Institute of Technology

Introduction

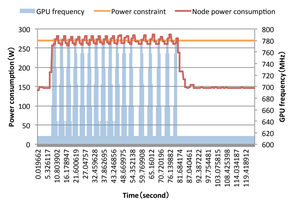
The issue of power consumption of HPC systems and supercomputers has been and will be an important research topic.
A realistic power budget for an exascale system is considered as 20 MW, which re- quires an energy efficiency of 50 GFLOPS/W.
Power capping has attracted attention to design “overprovisioned systems”, which improves power efficiency. Since modern processors are equipped with dynamic voltage frequency scaling(DVFS), we can satisfy the power limitation by lowing processor frequencies.
However, too low frequencies make application runtimes longer and sometimes harmful for energy optimization.

Objective and Approach

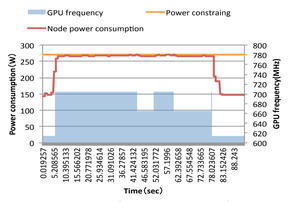
- Objective
  - Power control of CPU-GPU heterogeneous systems
  - Minimization of effect on performance and energy consumption by power control
- Approach
  - Monitoring power consumption
  - Change of GPU frequency dynamically

Power capping methods

Dynamic power capping method
- if power consumption exceeds the power limit, decrease GPU clock
- if power consumption is much lower than the power limit, decrease GPU clock
- GPU clock change interval is 5 seconds
  - if the interval is shorter, fluctuation of the clock speed, which leads a frequent excess of the power consumption



Interval is 5 seconds

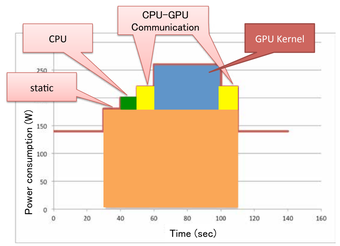


Interval is 0.2 seconds

Static power capping method
- determine appropriate CPUs and GPUs clock speeds based on a power and performance model

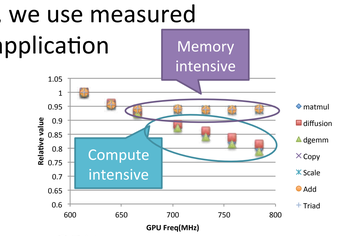
Power and performance models

Node energy consumption model
- $E = P_{static}*T_{all} + P_{Comm}*T_{Comm}$ $P_{CPU}*T_{CPU} + P_{GPU}*T_{GPU}$
- We construct a model equation for each variable



GPU power consumption model
- $P_{GPU}(f) = \alpha_{kernel}*A_{GPU}*V(f)^2*f+\beta_{kernel}*B_{GPU}+C_{GPU}*V(f)$
  - α,β represent characteristics of application kernels
  - A,B,C are architecture parameters
  - We construct the model by measuring test application in advance, it can estimate arbitrary application kernels power consumption

GPU performance model
- $T_{GPU}(f) = \max(T_{compute}(f), T_{memory})$
  - Computation cost is in inverse proportion to GPU clock speed
  - We assume that computation and memory operations are fully overlapped on GPU kernels
  - In order to obtain $T_{compute}(f), T_{memory}$, we use measured value preliminary execution of the application
- $T_{GPU}(f) = \max(T(f_{min})*f_{min}/f, T(f_{max}))$



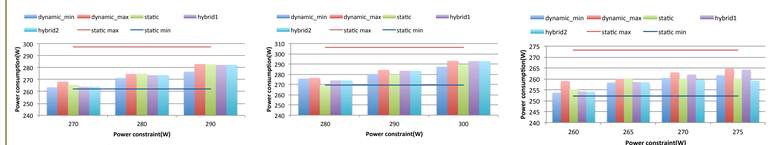
Evaluation

Experiment environment
- Computing node with two CPUs and two GPUs
- Applications use single CPU core and a GPU
- CPU : Intel® Xeon®CPU E5-2660
- GPU : NVIDIA K20Xm

GPU application benchmarks
- diffusion : thermal diffusion simulation
- matmul : matrix multiplication
- gpustream : stream benchmark, measuring memory bandwidth for GPU

Average power consumption

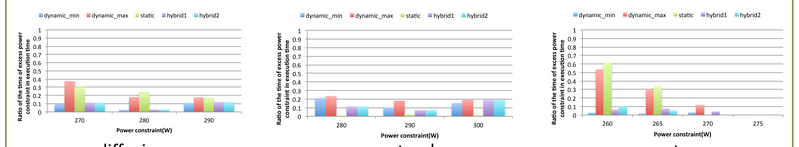
- Power capping techniques can control power consumption in all cases



diffusion matmul gpustream

Time of excess power constraint

- Excess durations tend to be larger with the static method and dynamic max
  - static : due to errors introduced by models
  - dynamic max : initial clock is maximum
- Excess durations tend to be smaller with dynamic min, hybrid1 and hybrid2
  - initial clock is optimal or minimum and they can decrease GPU clock when the power consumption exceeds power constraint

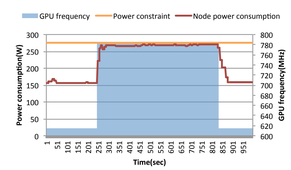


diffusion matmul gpustream

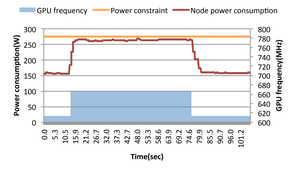
Energy consumption

diffusion matmul gpustream

- matmul and diffusion
  - as the power constraint is relaxed, energy consumption tends to be smaller
- gpustream
  - Energy consumption with higher power constraint does not reduce energy consumption, except with the static and hybrid2
  - Hybrid2 does not increase GPU frequency more than optimal one → reduce energy consumption



275W dynamic min



275W hybrid2

Future work

- Extension of our method for more CPU intensive applications
- Supporting applications that consist of several phases, each of which shows different characteristics in power and performance

Conclusion

- We have proposed an efficient power capping method for compute nodes equipped with accelerators, based on DVFS
- Proposed technique successfully reduces the excess of power consumption by errors of models