Overview of HPC and Energy Saving on KNL for Some Computations

Jack Dongarra

University of Tennessee
Oak Ridge National Laboratory
University of Manchester

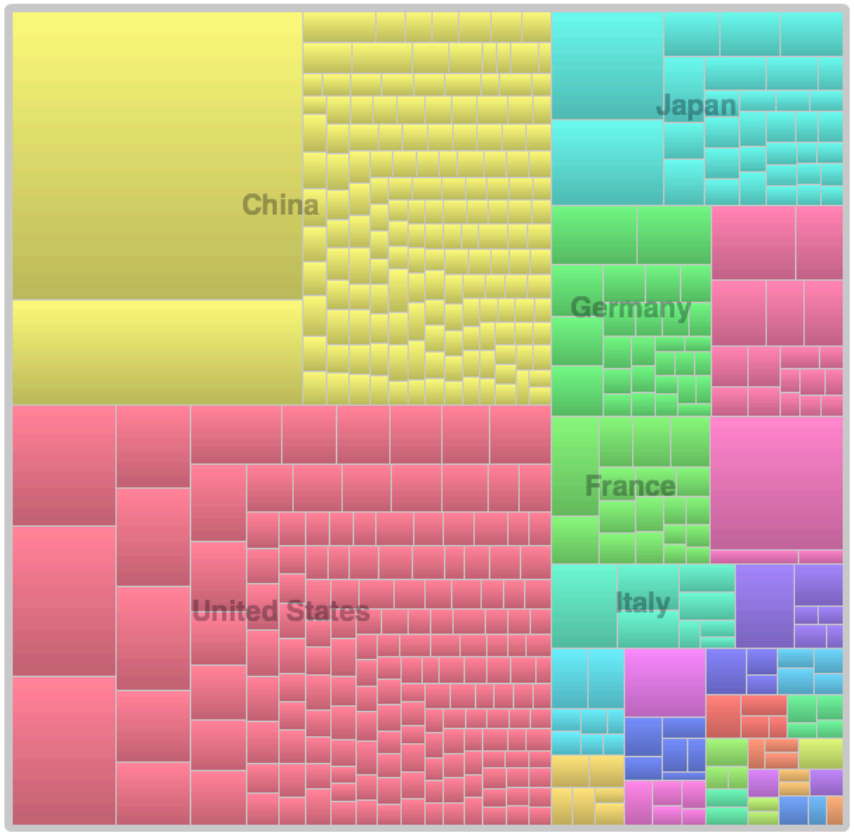
Outline

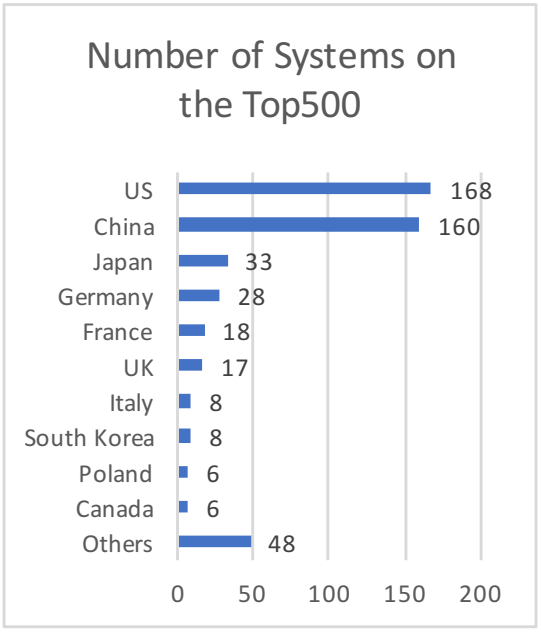
- Overview of High Performance Computing
- With Extreme Computing the “rules” for computing have changed

State of Supercomputing in 2017

- Pflops (> 10^{15} Flop/s) computing fully established with 138 computer systems.
- Three technology architecture or "swim lanes" are thriving.
  - Commodity (e.g. Intel)
  - Commodity + accelerator (e.g. GPUs) (91 systems)
  - Lightweight cores (e.g. IBM BG, Intel's Knights Landing, ShenWei 26010, ARM)
- Interest in supercomputing is now worldwide, and growing in many new markets (~50% of Top500 computers are in industry).
- Exascale (10^{18} Flop/s) projects exist in many countries and regions.
- Intel processors largest share, 92%, followed by AMD, 1%.

Countries Share





China has 1/3 of the systems, while the number of systems in the US has fallen to the lowest point since the TOP500 list was created.

Each rectangle represents one of the Top500 computers, area of rectangle reflects its performance.

June 2017: The TOP 10 Systems

Rank	Site	Computer	Country	Cores	Rmax [Pflops]	% of Peak	Power [MW]	GFlops/ Watt
1	National Super Computer Center in Wuxi	Sunway TaihuLight, SW26010 (260C) + Custom	China	10,649,000	93.0	74	15.4	6.04
2	National Super Computer Center in Guangzhou	Tianhe-2 NUDT, Xeon (12C) + IntelXeon Phi (57C) + Custom	China	3,120,000	33.9	62	17.8	1.91
3	Swiss CSCS	Piz Daint, Cray XC50, Xeon (12C) + Nvidia P100(56C) + Custom	Swiss	361,760	19.6	77	2.27	8.6
4	DOE / OS Oak Ridge Nat Lab	Titan, Cray XK7, AMD (16C) + Nvidia Kepler GPU (14C) + Custom	USA	560,640	17.6	65	8.21	2.14
5	DOE / NNSA L Livermore Nat Lab	Sequoia, BlueGene/Q (16C) + custom	USA	1,572,864	17.2	85	7.89	2.18
	DOE / OS	Cori, Cray XC40, Xeon Phi (68C)					3.94	3.55
							2.72	4.98
9	DOE / OS Argonne Nat Lab	Mira, BlueGene/Q (16C) + Custom	USA	786,432	8.59	85	3.95	2.07
10	DOE / NNSA / Los Alamos & Sandia	Trinity, Cray XC40,Xeon (16C) + Custom	USA	301,056	8.10	80	4.23	1.92
500	Internet company	Sugon Intel (8C) + Nnvidia	China	110,000	.432	71	1.2	0.36

TaihuLight is 60% Sum of All EU Systems

TaihuLight is 35% X Sum of All Systems in US

Piz Daint Power Optimized

- HPL benchmark was run twice, once "normal" and a second time on same size problem, on same number of nodes but with de-tune frequency:
- HPL-Normal : 19.6 PF/s, 2.27 MW, and 8.6 GF/W
  - CPU: 2.6 GHz + Turboboost & GPU: 1265 MHz
    - 10 cores used, 2 cores available to OS, Cuda and MPI.
- HPL Power-Opt: 16.96 PF/s, 1.63 MW, and 10.4 GF/w
  - CPU: 2.3 GHz & GPU: 1088 MHz
    - 8 cores used, 4 available to OS, Cuda and MPI.
- Effect: Piz Daint saved 28.2% of energy for a performance penalty of 13.5%

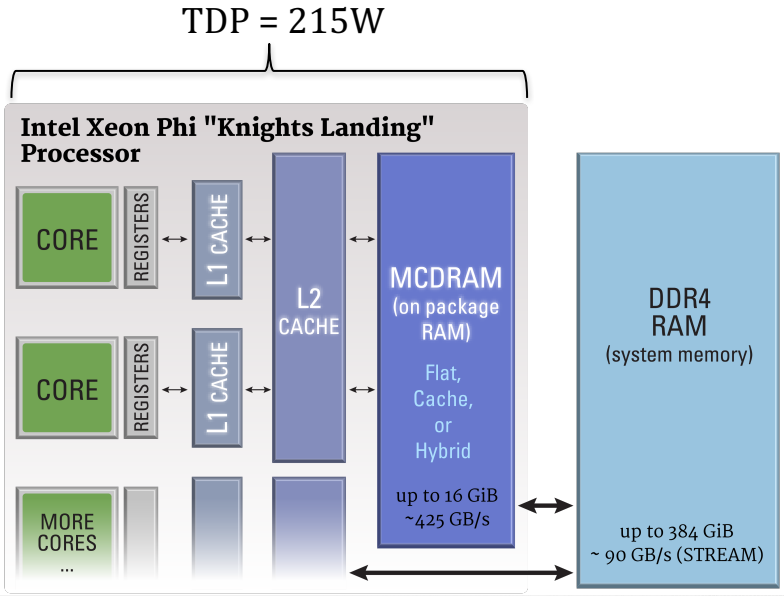
Issues and Problems Facing Extreme Scale

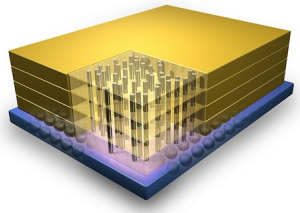
- Moore's Law and Dennard Scaling
- Data Movement Expensive Compared to Floating Point Operations
  - Hardware over-provisioned for floating point operations
  - Data movement can't keep up with floating point rates
- More Parallelism
  - Manycore
  - Hybrid Architectures
    - Not enough work for the number of cores
  - Memory storage not increasing to match flop rate
- Clock Variation
  - Turbo Boost
  - Processors heating up, TDP reached & reduce frequency
  - OS Jitter
- Performance Portability
  - Holy Grail of HPC Software
- Fault detection and recovery
  - Mean time between hardware/software errors shortening
- Energy Consumption and energy efficiency

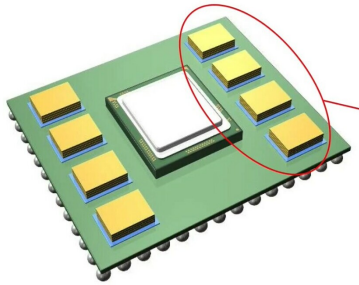
Intel Knights Landing: FLAT mode

68 cores, 1.3 GHz, Peak DP = 2662 Gflop/s

- KNL is in FLAT mode (where entire MCDRAM is used as addressable memory)
  → memory allocations are treated similarly to DDR4 memory allocations
- KNL Thermal Design Power (TDP) is 215W (for main SKUs):



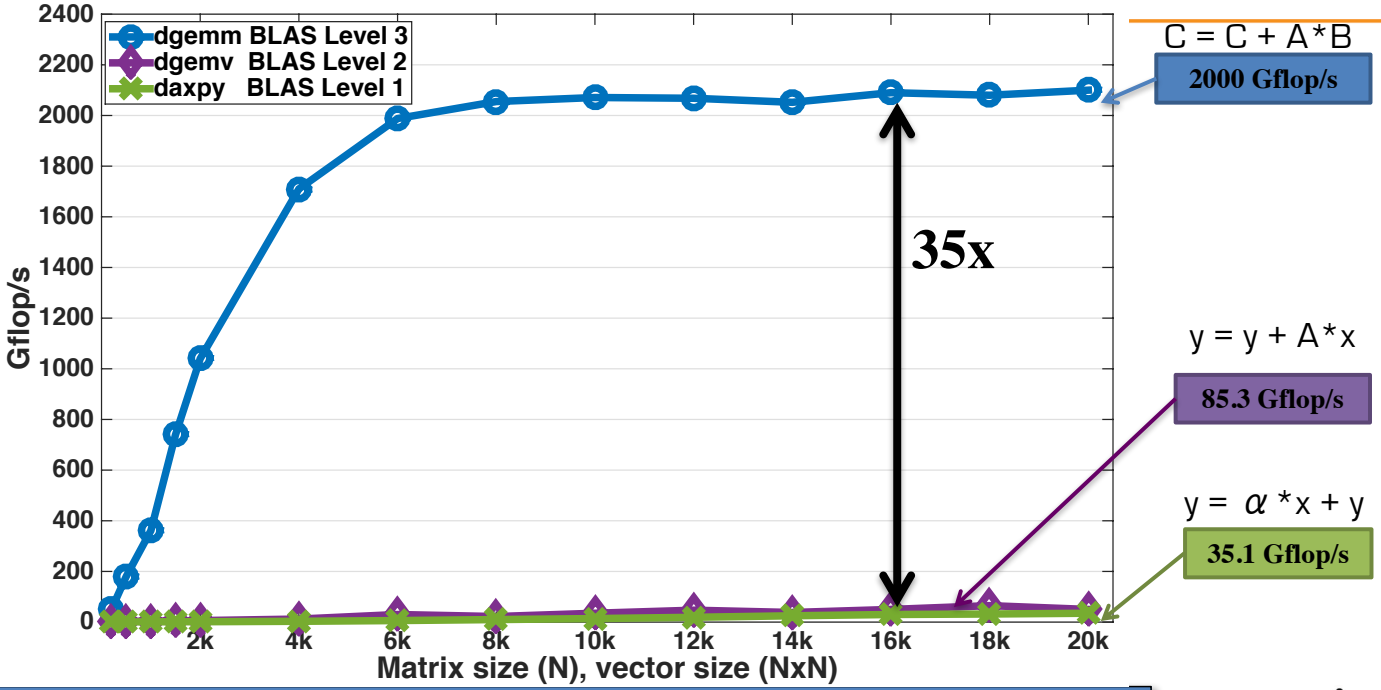




- 5X the bandwidth v. GDDR5
- Up to 16GB
- One-third the footprint
- Half the energy per bit
- Managed memory stack for opt levels of reliability, availability a serviceability

Level 1, 2 and 3 BLAS

68 cores **Intel Xeon Phi KNL**, 1.3 GHz, Peak DP = 2662 Gflop/s



68 cores Intel Xeon Phi KNL, 1.3 GHz
The theoretical peak double precision is 2662 Gflop/s
Compiled with icc and using Intel MKL 2017b1 20160506

PAPI for power-aware computing

- We use PAPI's latest **powercap component** for measurement, control, and performance analysis
  - PAPI power components in the past supported **only reading** power information
  - New component exposes RAPL functionality to allow users to **read and write** power limit

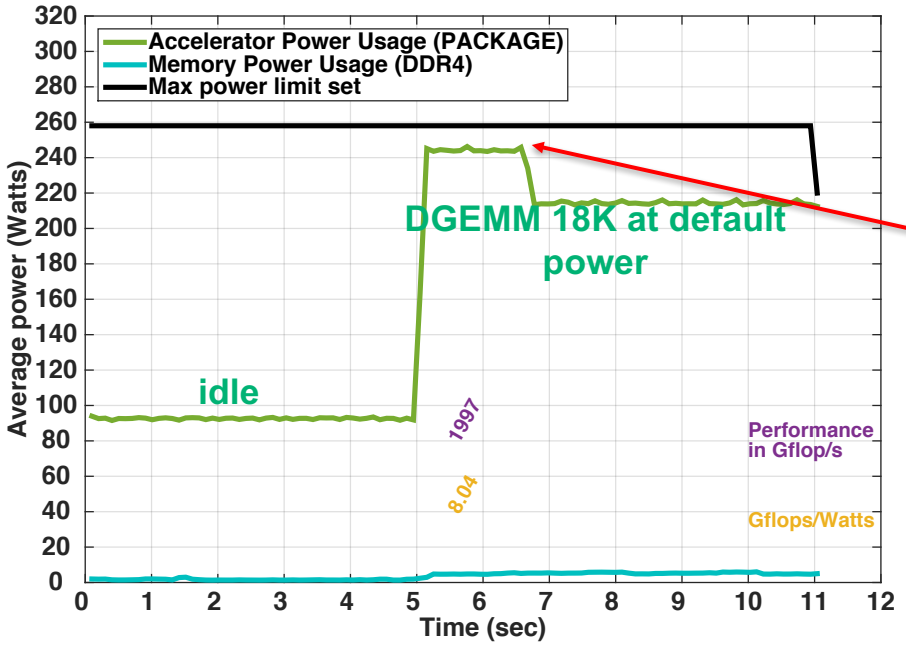
PAPI for power-aware computing

- We use PAPI's latest **powercap component** for measurement, control, and performance analysis
  - PAPI power components in the past supported **only reading** power information
  - New component exposes RAPL functionality to allow users to **read and write** power limit
- Study numerical building blocks of varying computational intensity
- Use PAPI powercap component to detect power optimization opportunities

→ **Objective:** Cap the power on the architecture to reduce power usage while keeping the execution time constant → **Energy Savings** !!!

Level 3 BLAS DGEMM on KNL using MCDRAM

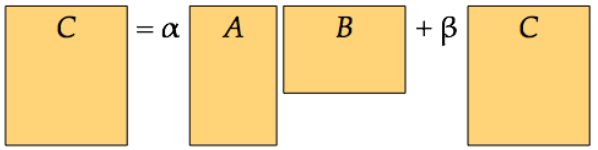
68 cores **KNL**, Peak DP = 2662 Gflop/s Bandwidth **MCDRAM ~425 GB/s** DDR4 ~90 GB/s



set pwr limit default

DGEMM size 18K

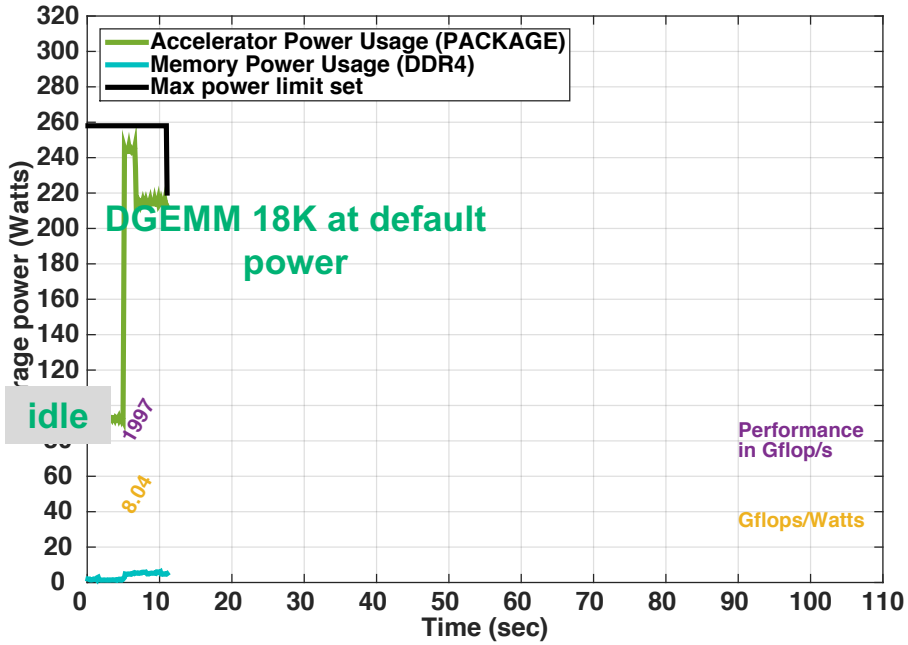
Thermal Design Power is reached



DGEMM is run repeatedly for a fixed matrix size of 18K per step and at each step a new power limit is set in decreasing fashion starting from default, 220 Watts, 200 Watts down till 120 Watts by steps of 10/20.

Level 3 BLAS DGEMM on KNL using MCDRAM

68 cores **KNL**, Peak DP = 2662 Gflop/s Bandwidth **MCDRAM ~425 GB/s** DDR4 ~90 GB/s



set pwr limit default

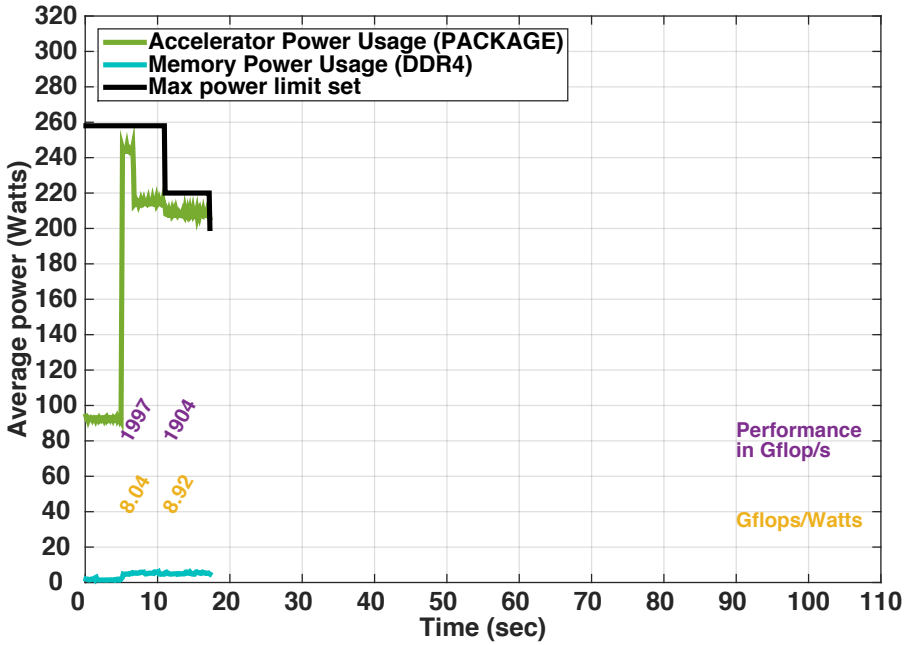
DGEMM size 18K

DGEMM is run repeatedly for a fixed matrix size of 18K per step and at each step a new power limit is set in decreasing fashion starting from default, 220 Watts, 200 Watts down till 120 Watts by steps of 10/20.

THE UNIVERSITY of TENNESSEE
Department of Electrical Engineering and Computer Science

Level 3 BLAS DGEMM on KNL using MCDRAM

68 cores **KNL**, Peak DP = 2662 Gflop/s Bandwidth **MCDRAM ~425 GB/s** DDR4 ~90 GB/s

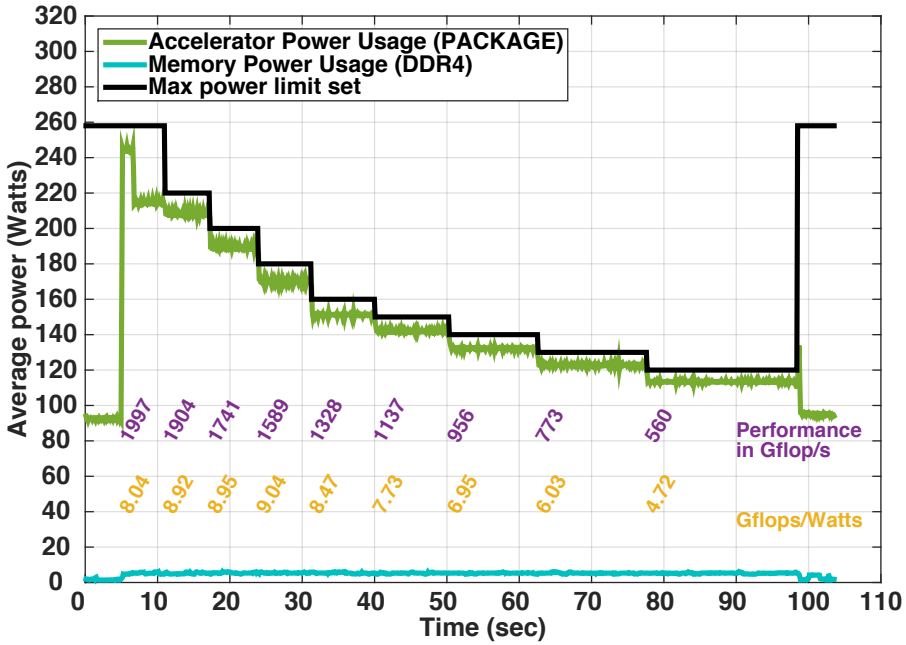


set pwr limit default
DGEMM size 18K
set pwr limit 220W
DGEMM size 18K

DGEMM is run repeatedly for a fixed matrix size of 18K per step and at each step a new power limit is set in decreasing fashion starting from default, 220 Watts, 200 Watts down till 120 Watts by steps of 10/20.

Level 3 BLAS DGEMM on KNL using MCDRAM

68 cores **KNL**, Peak DP = 2662 Gflop/s Bandwidth **MCDRAM ~425 GB/s** DDR4 ~90 GB/s



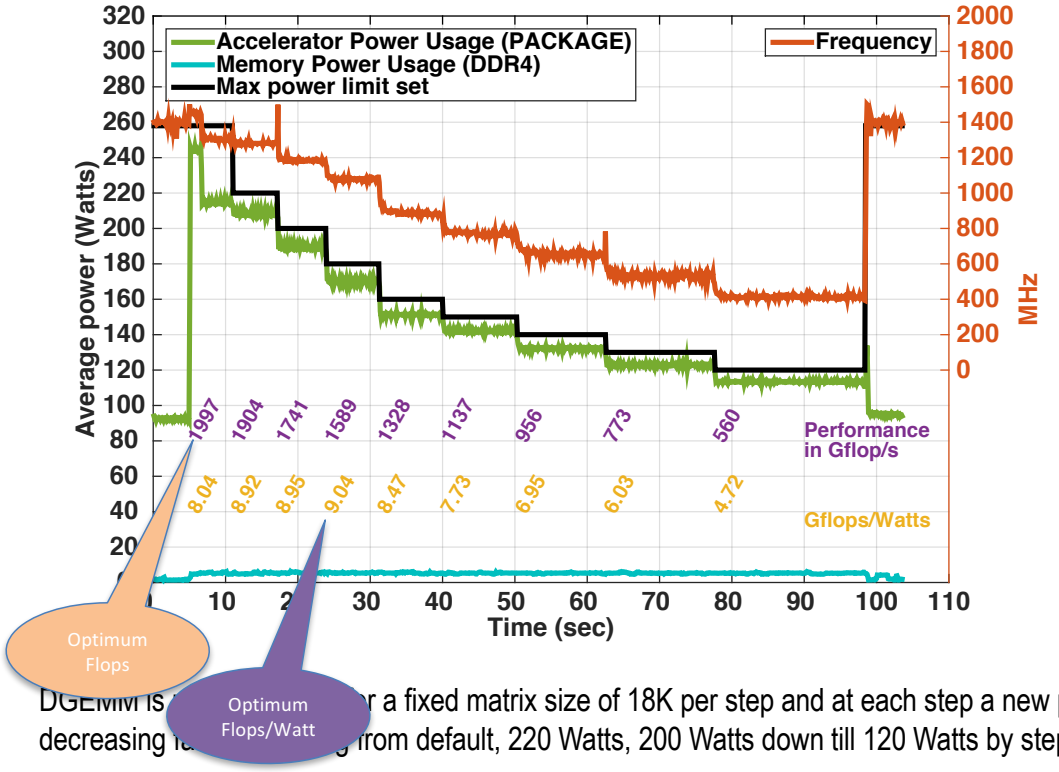
set pwr limit default
DGEMM size 18K
set pwr limit 220W
DGEMM size 18K
set pwr limit 200W
DGEMM size 18K
set pwr limit 180W
DGEMM size 18K
set pwr limit 160W
DGEMM size 18K
set pwr limit 150W
....

DGEMM is run repeatedly for a fixed matrix size of 18K per step and at each step a new power limit is set in decreasing fashion starting from default, 220 Watts, 200 Watts down till 120 Watts by steps of 10/20.

THE UNIVERSITY of TENNESSEE
Department of Electrical Engineering and Computer Science

Level 3 BLAS DGEMM on KNL using MCDRAM

68 cores **KNL**, Peak DP = 2662 Gflop/s Bandwidth **MCDRAM ~425 GB/s** DDR4 ~90 GB/s

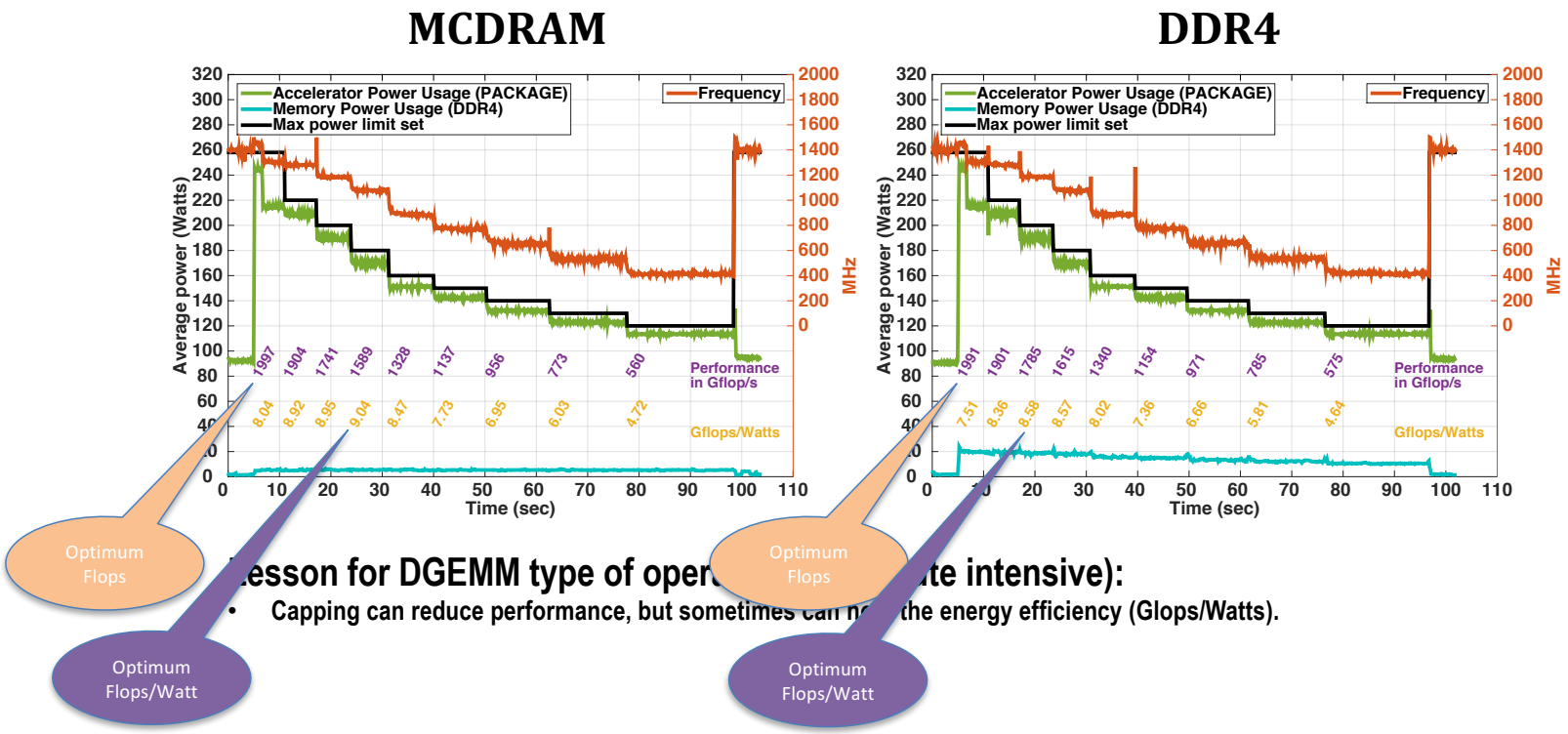


- **When power cap kick-on the frequency is decreased which confirm that capping affect through the DVFS**

DGEMM is … a fixed matrix size of 18K per step and at each step a new power limit is set in decreasing … from default, 220 Watts, 200 Watts down till 120 Watts by steps of 10/20.

THE UNIVERSITY of TENNESSEE
Department of Electrical Engineering and Computer Science

Level 3 BLAS DGEMM on KNL MCDRAM/DDR4

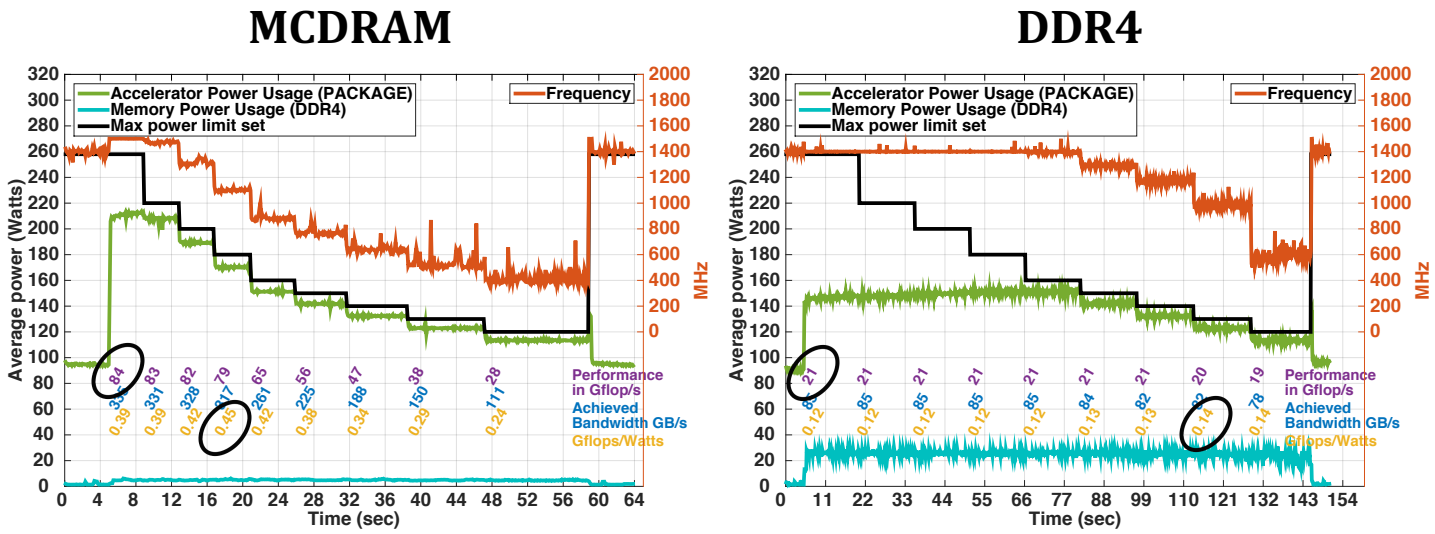


Lesson for DGEMM type of oper… …te intensive):

- **Capping can reduce performance, but sometimes can n… …the energy efficiency (Glops/Watts).**

DGEMM is run repeatedly for a fixed matrix size of 18K per step and at each step a new power limit is set in decreasing fashion starting from default, 220 Watts, 200 Watts down till 120 Watts by steps of 10/20.

Level 2 BLAS DGEMV on KNL MCDRAM/DDR4



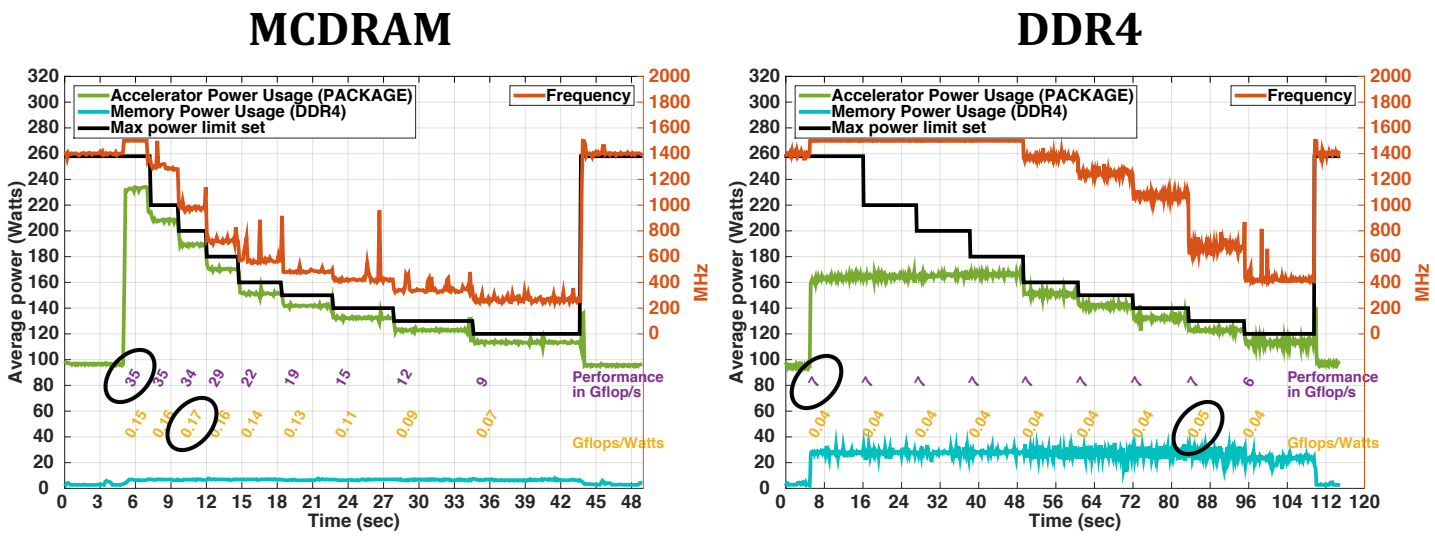
Lesson for DGEMV type of operations (memory bound):

- **For MCDRAM, capping at 190 Watts reduces energy consumption by 20% without any loss of performance!**
- **For DDR4, capping at 130 Watts improves energy efficiency by ~17%.**
- **Overall capping 40 Watts below the observed power at default setting provides about 17-20% energy savings without significant increase in time to solution.**

DGEMV is run repeatedly for a fixed matrix size of 18K per step and at each step a new power limit is set in decreasing fashion starting from default, 220 Watts, 200 Watts down till 120 Watts by steps of 10/20.

THE UNIVERSITY of TENNESSEE
Department of Electrical Engineering and Computer Science

Level 1 BLAS DAXPY on KNL MCDRAM/DDR4



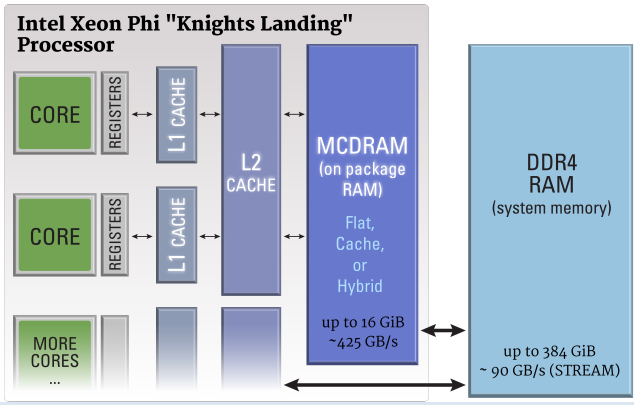
Lesson for DAXPY type of operations (memory bound):

- **For MCDRAM, capping at 180 Watts improves energy efficiency by ~16% without loss in performance.**
- **For DDR4, capping at 130 Watts improves energy efficiency by ~25%.**
- **Overall, capping 40 Watts below the observed power at default setting provides about 16-25% energy saving without significantly reducing the time to solution.**

DAXPY is run repeatedly for a fixed matrix size of 1E8 per step and at each step a new power limit is set in decreasing fashion starting from default, 220 Watts, 200 Watts down till 120 Watts by steps of 10/20.

THE UNIVERSITY of TENNESSEE
Department of Electrical Engineering and Computer Science

Intel Knights Landing

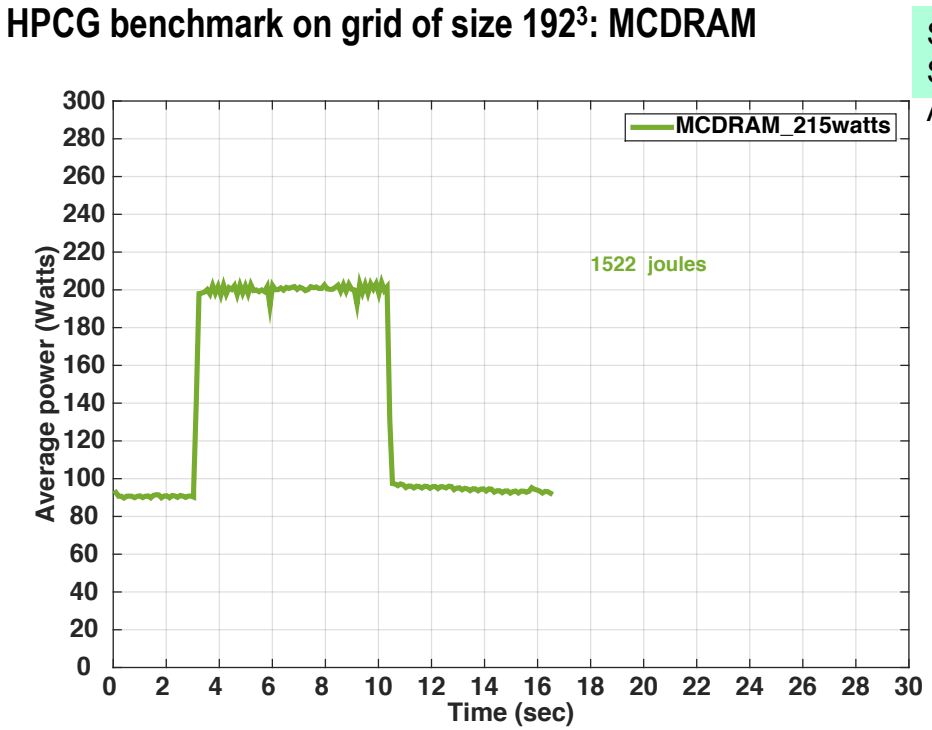


Level 1, 2 and 3 BLAS

68 cores **Intel Xeon Phi KNL**, 1.3 GHz, Peak DP = 2662 Gflop/s

MCDRAM			DRAM		
BLAS	Rate Gflop/s	Power Efficiency Gflop/s per W	BLAS	Rate Gflop/s	Power Efficiency Gflop/s per W
Level 3	1997	9.04	Level 3	1991	8.58
Level 2	84	.45	Level 2	21	.12
Level 1	35	.17	Level 1	7	.05

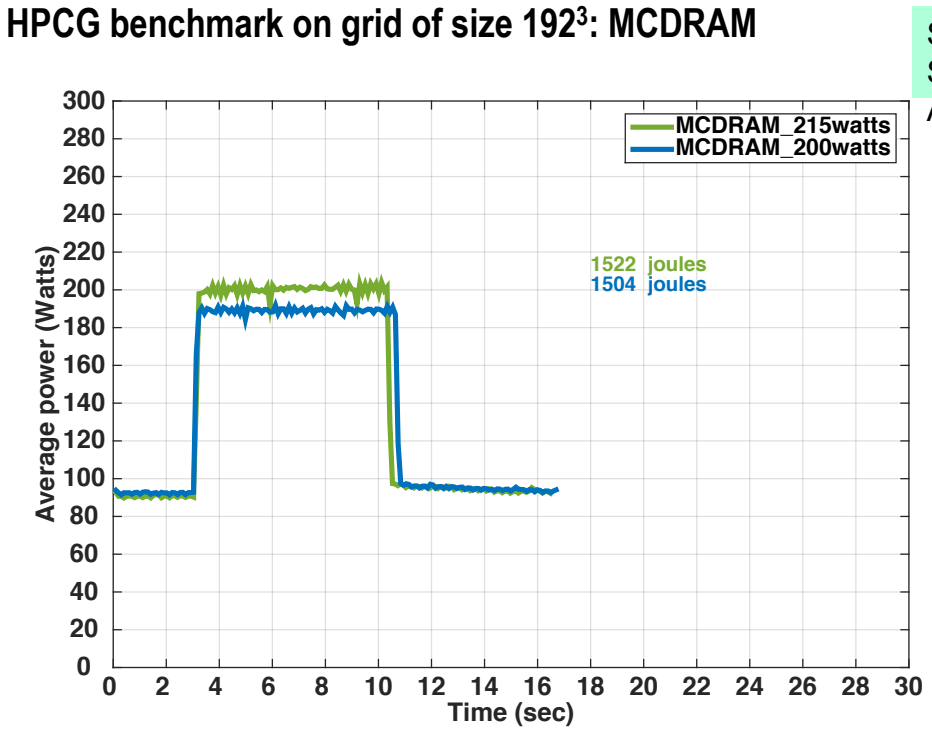
Power-awareness in applications



Sparse Matrix Vector operations
Solving Ax=b with Conjugate Gradient Algorithm

- **At TDP basic power limit 215**

Power-awareness in applications



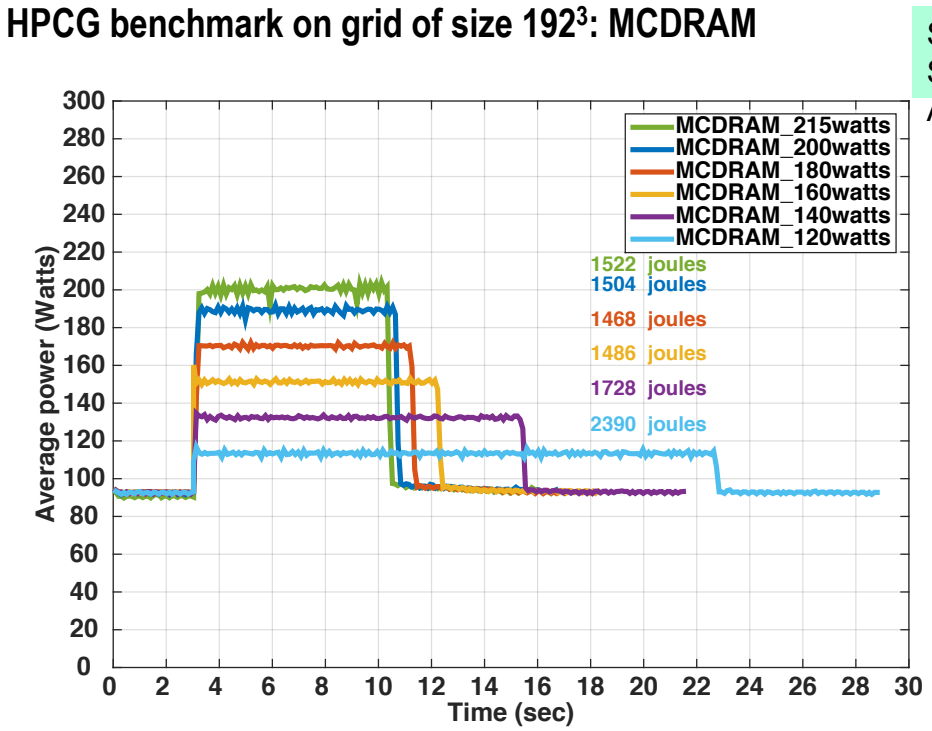
Sparse Matrix Vector operations
Solving Ax=b with Conjugate Gradient Algorithm

- **Decreasing the power limit to 200 W do not results in any loss in performance while we observe a reducing of power rate consumption**

ICL UT INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY of TENNESSEE
Department of Electrical Engineering and Computer Science

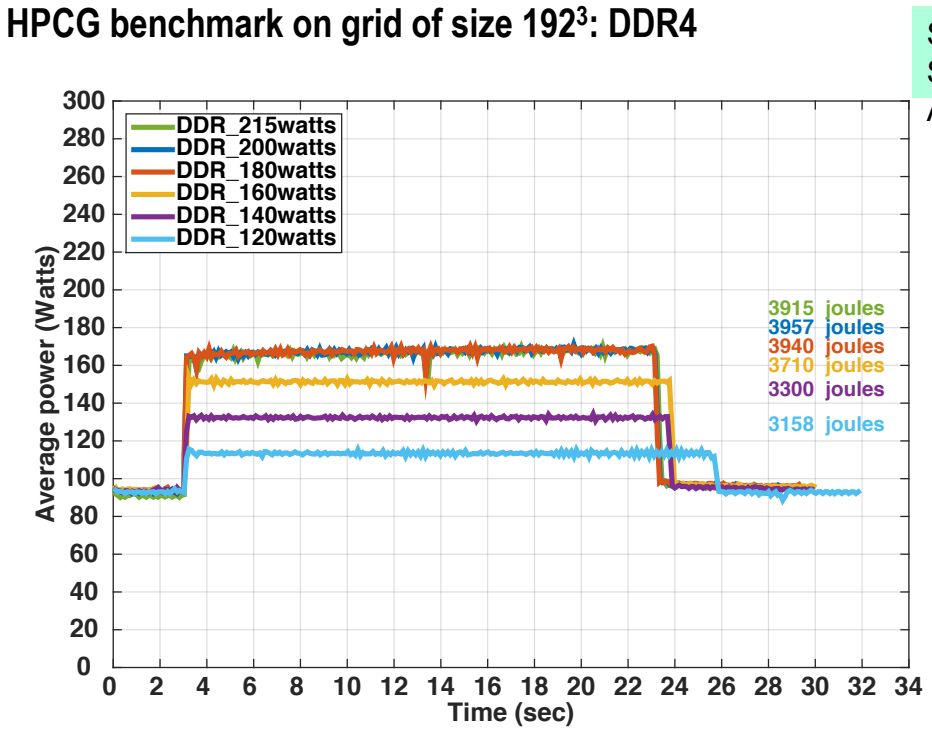
Power-awareness in applications



Sparse Matrix Vector operations
Solving Ax=b with Conjugate Gradient Algorithm

- **Decreasing the power limit to 200 W do not results in any loss in performance while we observe a reducing of power rate consumption**
- **Decreasing the power limit down by 40 Watts (setting pwr limit at 160 W) will provide energy saving, power rate reduction and without any meaningful loss in performance, less than 10% reduction in time to solution.**

Power-awareness in applications



Sparse Matrix Vector operations
Solving Ax=b with Conjugate Gradient Algorithm

- Decreasing the power limit to 160 W do not results in any loss in performance while we observe a reducing of power rate and energy saving.
- Decreasing the power limit down by 40 Watts (Pwr limit at 140 W) will providing large energy saving (15%), power rate reduction, without any loss in performance.
- At 120 Watts, less than 10% reduction in time to solution while a large energy saving (20%) and power rate reduction are observed

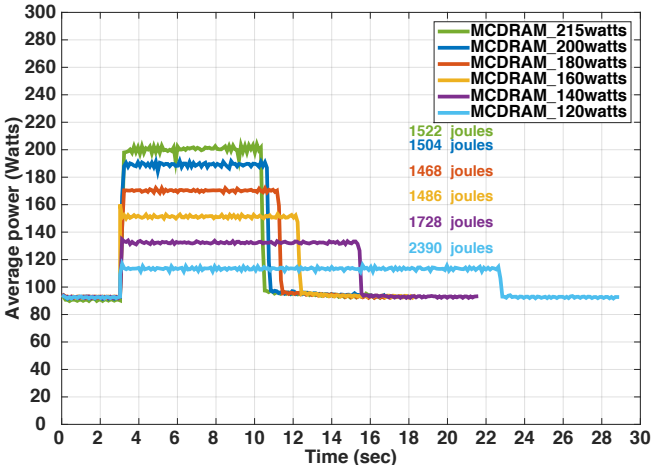
ICL UT INNOVATIVE COMPUTING LABORATORY

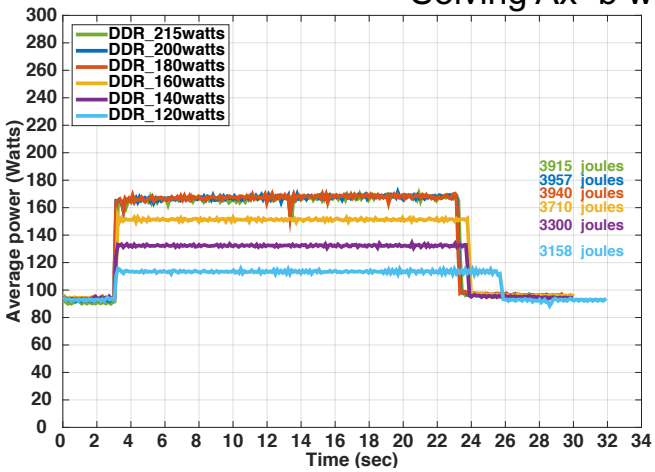
THE UNIVERSITY of TENNESSEE
Department of Electrical Engineering and Computer Science

Power-awareness in applications

HPCG benchmark on grid of size 192^3: MCDRAM & DDR4

Sparse Matrix Vector operations
Solving Ax=b with CG



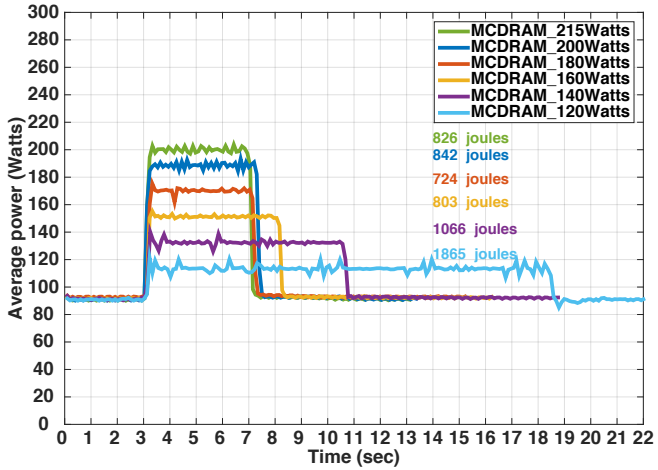


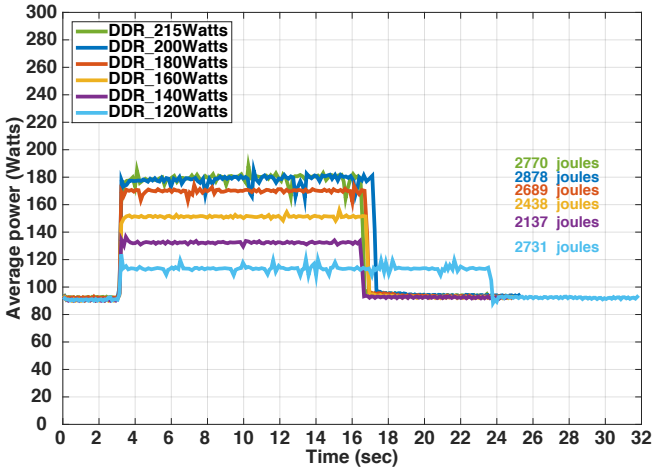
Lesson for HPCG:

- For MCDRAM, decreasing the power limit by 40 Watts from the power observed at default (setting Pwr limit at 160 W) will provide energy saving, power rate reduction without any meaningful loss in performance, less than 10% reduction in time to solution.
- For DDR4, similar behavior observed by decreasing the power limit by 40 Watts from the power observed at default (setting Pwr limit at 120 W) provide a large energy saving (15%-20%), power rate reduction without any significant reduction in time to solution.

Power-awareness in applications

Solving Helmholtz equation with finite difference, Jacobi iteration 12800x12800 grid





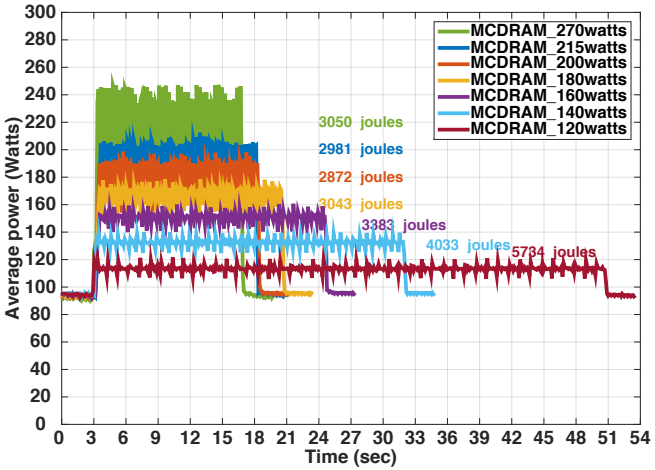
Lesson for Jacobi iteration:

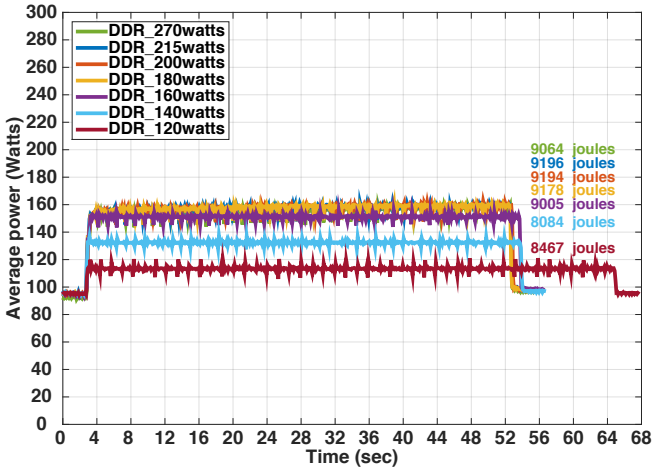
- For MCDRAM, capping at 180 Watts improves power efficiency by ~13% without any loss in time to solution, while also decreasing power rate requirements by about 20%
- For DDR4, capping at 140 Watts improves power efficiency by ~20% without any loss in time to solution.
- Overall, capping 30-40 Watts below the power observed at default limit provides large energy gain while keeping up with the same time to solution, which is similar to the behavior observed for the DGEMV and DAXPY kernels.

THE UNIVERSITY of TENNESSEE
Department of Electrical Engineering and Computer Science

Power-awareness in applications

Lattice-Boltzmann simulation of CFD (from SPEC 2006 benchmark)





Lesson for Lattice Boltzmann:

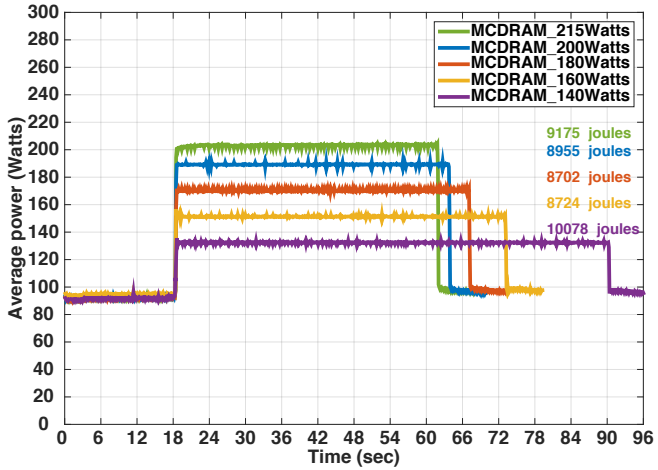
- For MCDRAM, capping at 200 results in 5% energy saving and power rate reduction without increase in time to solution.
- For DDR4, capping at 140 Watts improves energy efficiency by ~11% and reduces power rate without any increase in time to solution.

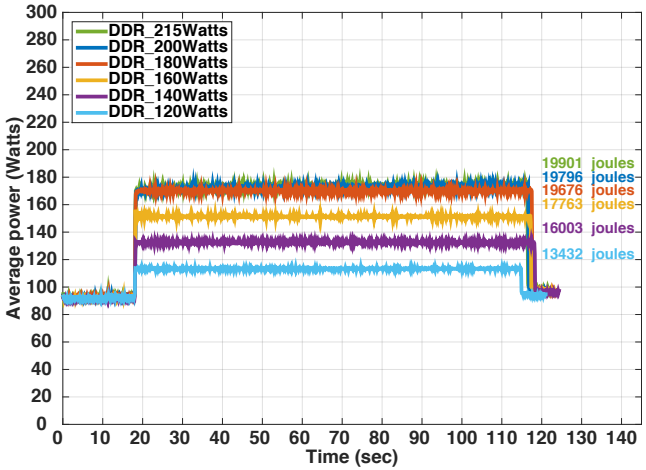
ICL UT INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY of TENNESSEE Department of Electrical Engineering and Computer Science

Power-awareness in applications

Monte Carlo neutron transport application (from XSbench)





Lesson for Monte Carlo:

- For MCDRAM, capping at 180 W results in energy saving (5%) and power rate reduction without any meaningful increase in time to solution.
- For DDR4, capping at 120 Watts improves power efficiency by ~30% without any loss in performance.

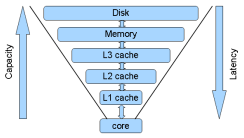
Power Aware Tools

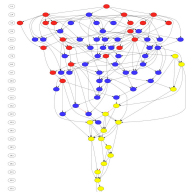
- Designing a toolset that will “automatically” monitor memory traffic.
  - Track the performance and data movement.
- Adjust the power to keep the performance roughly the same while decreasing energy consumption.
- Better power performance without significant loss of performance.

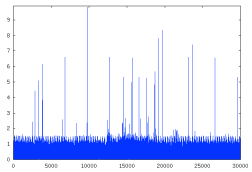
- All in a automatic fashion without user involvement.

Software and Algorithm Must Keep Pace with the Changes in Hardware

- Classical analysis of algorithms is not valid,
  - # of floating point ops ≠ computation time.
- Algorithms and software must take advantage by reducing data movement.
  - Need latency tolerance in our algorithms
- Communication and synchronization reducing algorithms and software are critical.
  - As parallelism grows
- Hardware presents a dynamically changing environment
  - Turbo Boost and OS jitter
- Many existing algorithms can't fully exploit the features of modern architecture







Mixed-Precision Iterative Refinement

- Iterative refinement for dense systems, $Ax = b$, can work this way.

L U = lu(A)	$O(n^3)$
x = L\(U\b)	$O(n^2)$
r = b – Ax	$O(n^2)$
WHILE || r || not small enough	
z = L\(U\r)	$O(n^2)$
x = x + z	$O(n^1)$
r = b – Ax	$O(n^2)$
END	

  - Wilkinson, Moler, Stewart, & Higham provide error bound for SP fl pt results when using DP fl pt.

Mixed-Precision Iterative Refinement

- Iterative refinement for dense systems, $Ax = b$, can work this way.

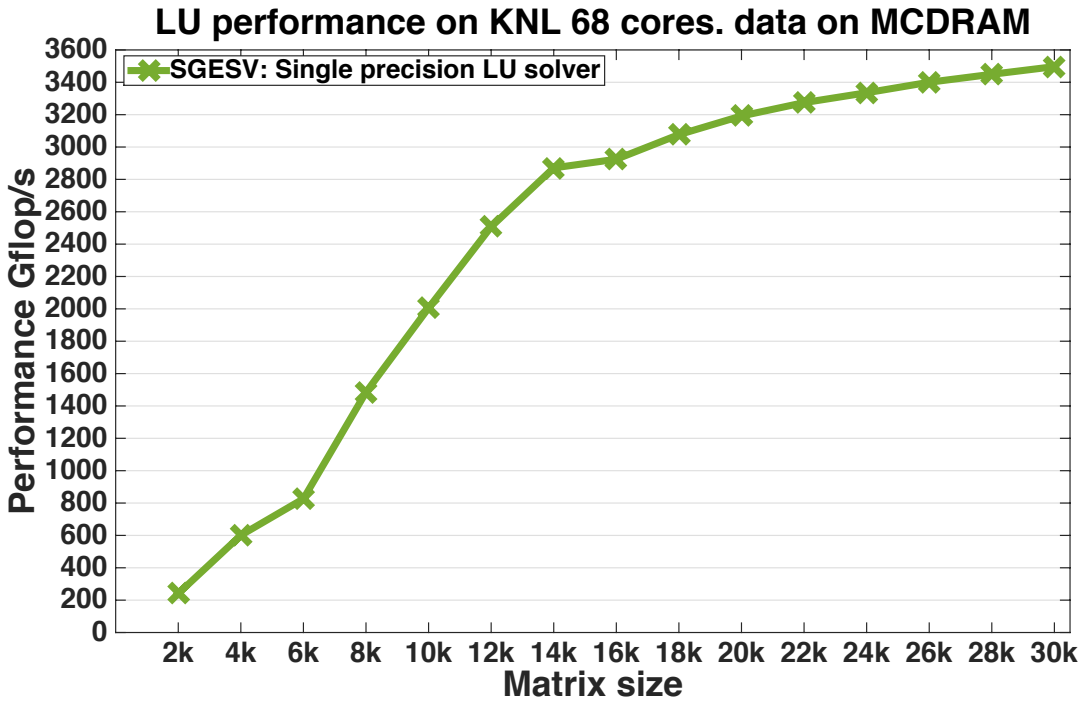
L U = lu(A)	SINGLE	$O(n^3)$
x = L\(U\b)	SINGLE	$O(n^2)$
r = b – Ax	DOUBLE	$O(n^2)$
WHILE || r || not small enough		
z = L\(U\r)	SINGLE	$O(n^2)$
x = x + z	DOUBLE	$O(n^1)$
r = b – Ax	DOUBLE	$O(n^2)$
END		

- Wilkinson, Moler, Stewart, & Higham provide error bound for SP fl pt results when using DP fl pt.
- It can be shown that using this approach we can compute the solution to 64-bit floating point precision.

- Requires extra storage, total is 1.5 times normal;
- $O(n^3)$ work is done in lower precision
- $O(n^2)$ work is done in high precision
- Problems if the matrix is ill-conditioned in sp; $O(10^8)$

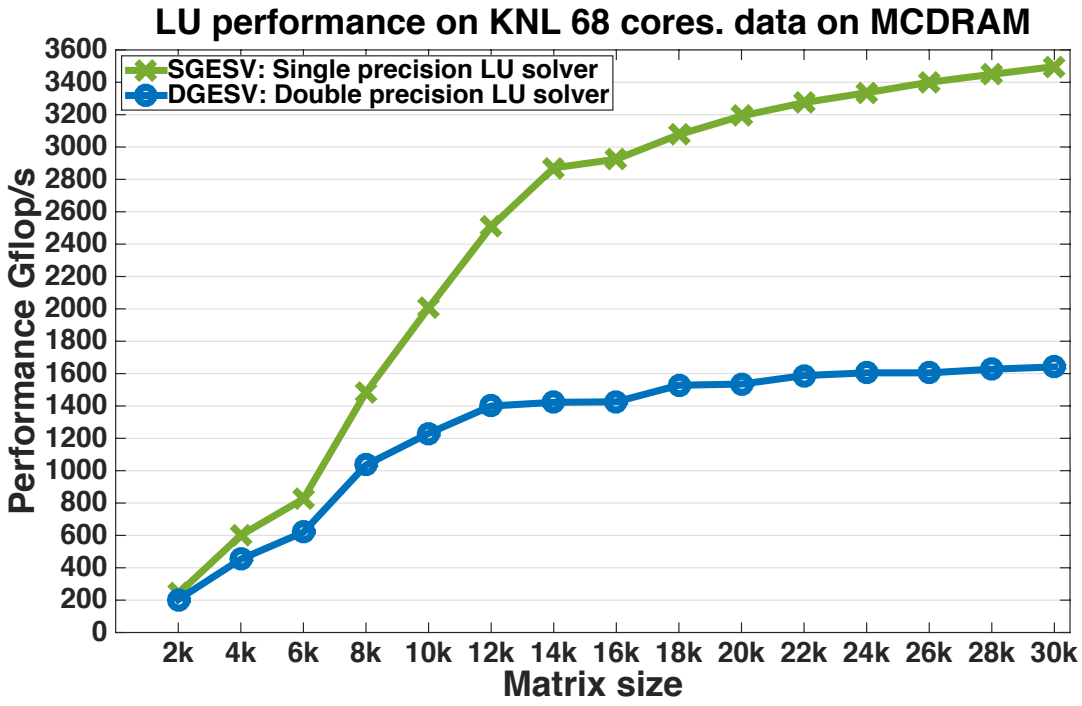
Power-awareness in Algorithms

Iterative refinement to solve Ax=b getting a solution in double precision arithmetic



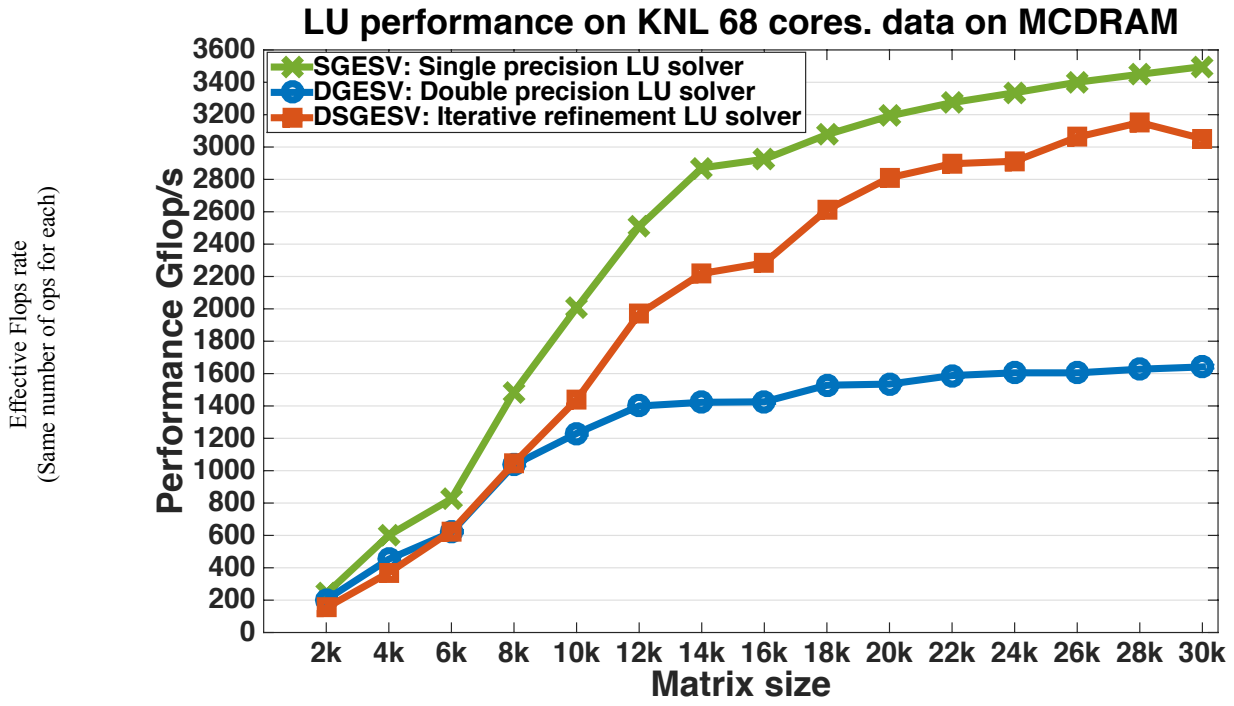
Power-awareness in Algorithms

Iterative refinement to solve Ax=b getting a solution in double precision arithmetic



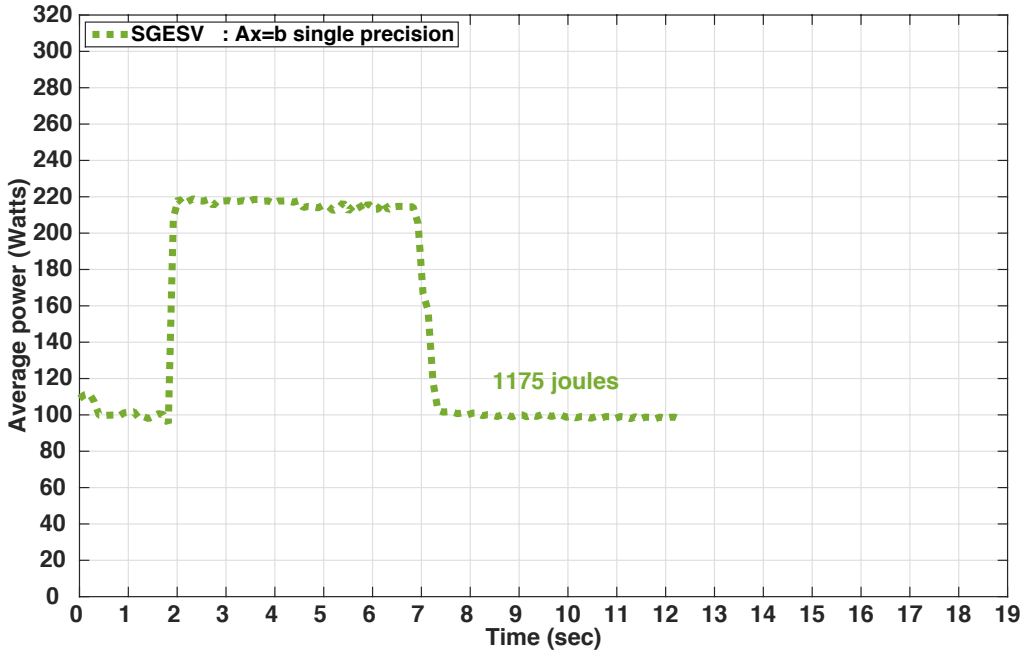
Power-awareness in Algorithms

Iterative refinement to solve Ax=b getting a solution in double precision arithmetic



Power-awareness in Algorithms

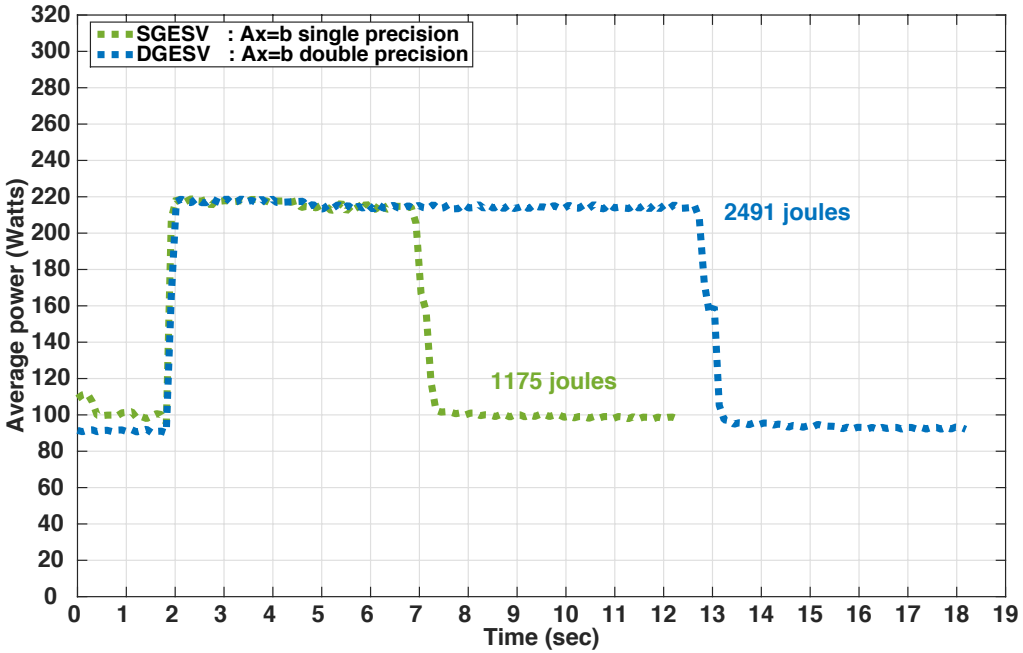
Iterative refinement to solve Ax=b getting a solution in double precision arithmetic



- Power consumption of SP, DP, and mixed precision algorithm to solve Ax=b for a matrix of size 30K on KNL.

Power-awareness in Algorithms

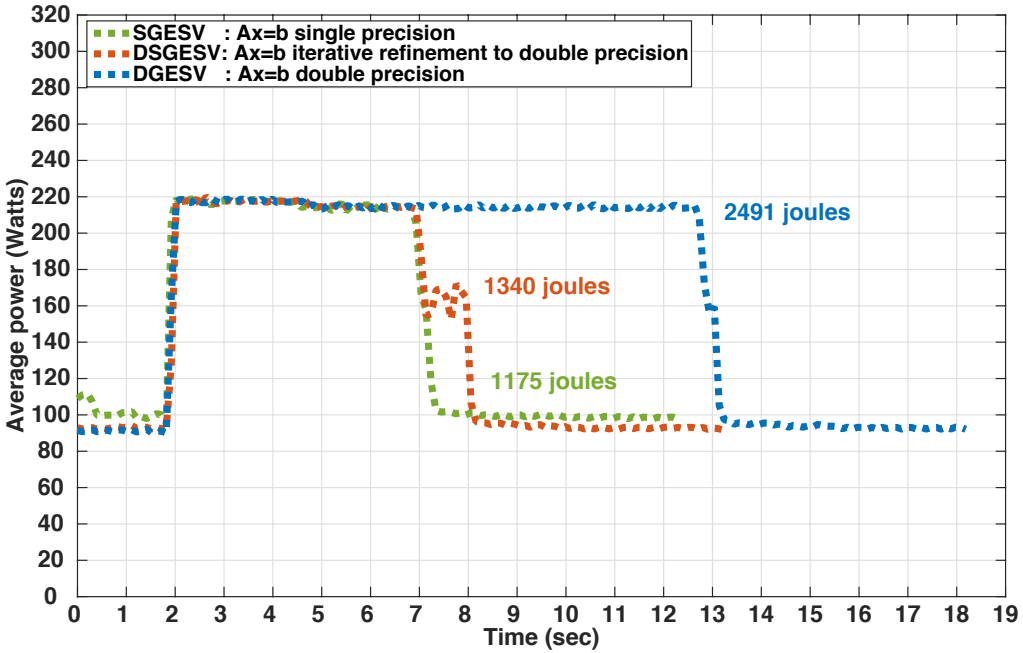
Iterative refinement to solve Ax=b getting a solution in double precision arithmetic



- Power consumption of SP, DP, and mixed precision algorithm to solve Ax=b for a matrix of size 30K on KNL.

Power-awareness in Algorithms

Iterative refinement to solve Ax=b getting a solution in double precision arithmetic



- Power consumption of SP, DP, and mixed precision algorithm to solve Ax=b for a matrix of size 30K on KNL.
- Algorithmic advancements such as mixed precision techniques can also provide a large gain in energy efficiency (45%). We can reduce the energy consumption by about the half.

Mixed-Precision Iterative Refinement

- Iterative refinement for dense systems, $Ax = b$, can work this way.

L U = lu(A)	SINGLE	$O(n^3)$
x = L\(U\b)	SINGLE	$O(n^2)$
r = b – Ax	DOUBLE	$O(n^2)$
WHILE || r || not small enough		
z = L\(U\r)	SINGLE	$O(n^2)$
x = x + z	DOUBLE	$O(n^1)$
r = b – Ax	DOUBLE	$O(n^2)$
END		

- Wilkinson, Moler, Stewart, & Higham provide error bound for SP fl pt results when using DP fl pt.
- It can be shown that using this approach we can compute the solution to 64-bit floating point precision.

- Requires extra storage, total is 1.5 times normal;
- $O(n^3)$ work is done in lower precision
- $O(n^2)$ work is done in high precision
- Problems if the matrix is ill-conditioned in sp; $O(10^8)$

Critical Issues and Challenges at Peta & Exascale for Algorithm and Software Design

- Synchronization-reducing algorithms
  - Break Fork-Join model
- Communication-reducing algorithms
  - Use methods which have lower bound on communication
- Mixed precision methods
  - 2x speed of ops and 2x speed for data movement
- Autotuning
  - Today's machines are too complicated, build "smarts" into software to adapt to the hardware
- Fault resilient algorithms
  - Implement algorithms that can recover from failures/bit flips
- Reproducibility of results
  - Today we can't guarantee this, without a penalty. We understand the issues, but some of our "colleagues" have a hard time with this.

Collaborators and Support

MAGMA team
http://icl.cs.utk.edu/magma

PLASMA team
http://icl.cs.utk.edu/plasma

Collaborating partners

University of Tennessee, Knoxville
University of California, Berkeley
University of Colorado, Denver
University of Manchester

U.S. DEPARTMENT OF
ENERGY

Umeå University

INRIA

Rutherford Appleton Laboratory

University of Manchester