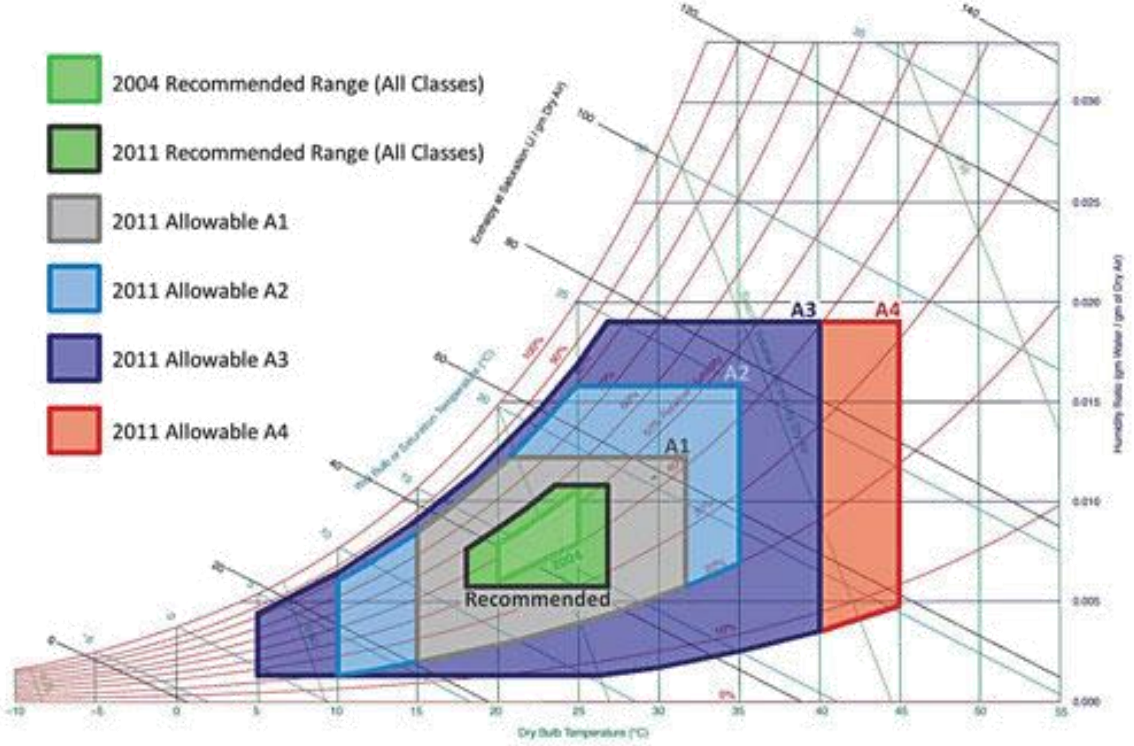
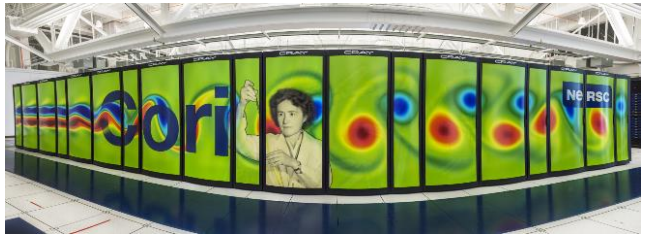
Energy Efficiency Gains From Software: Retrospectives and Perspectives

Dan Reed Reed, Iowa

Sadaf Alam, CSCS/ETH
Bill Gropp, NCSA/Illinois
Satoshi Matsuoka, GSIC/Tokyo Tech
John Shalf, NERSC/LBL





The iceberg view

Gray's cost/performance axes

- Networking
- Computation
- Storage
- Access

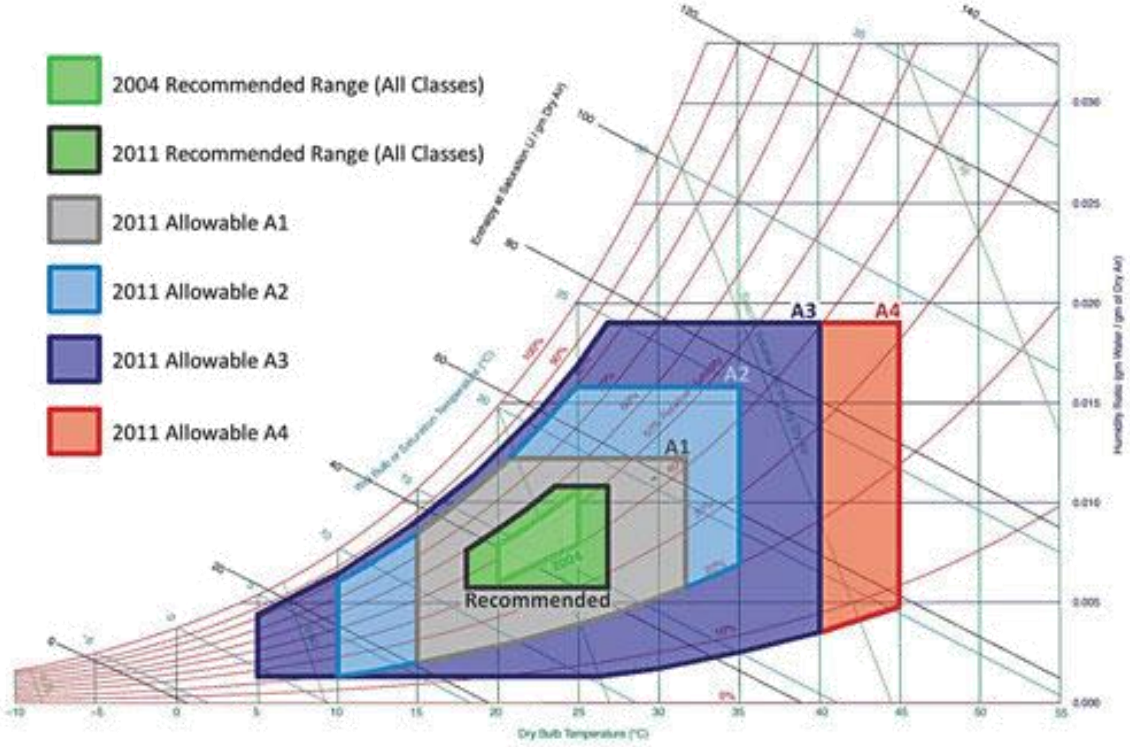
… still driving change

Facility, systems, operations

Software …

Inspiration: Christian Belady

Money, energy, and resilience



Christian Belady, Microsoft

THE UNIVERSITY OF IOWA

Some panel framing questions

What system software has been most helpful in improving system energy efficiency?

Which software stack layers should support energy awareness?

- System software, libraries, MPI runtime/job schedulers, applications, …

Is there user willingness to exploit libraries for energy efficiency?

What might encourage adoption of APIs for measurement/management?

- Sandia Power API, Intel GEOPM, Redfish, …

Where are the biggest opportunities for energy efficiency improvements?

A few more provocative questions

At a PUE of <1.1, how much more is there to be gained?

$$\text{PUE} = \frac{\textbf{Total Facility Energy}}{\textbf{IT Equipment Energy}}$$

Can software-based runtime introspection mechanisms react fast enough?

- Or is dedicated hardware required for the control loop?

What is the user reward for developing energy efficient codes?

- Versus investing time in making the software more scientifically capable

What is the total cost of ownership (TCO) tradeoff in tuning codes for *(a) improved performance* or *(b) reduced energy consumption*

What are the fundamental properties of energy efficient applications?

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

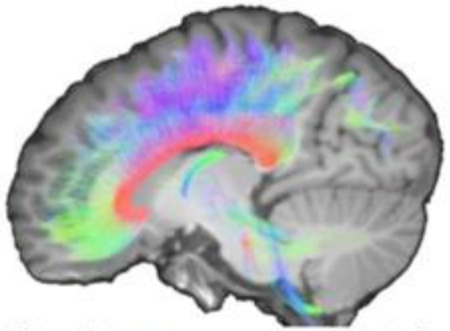
ETH zürich

SC17 Panel: Energy Efficiency Gains From Software: Retrospectives and Perspectives

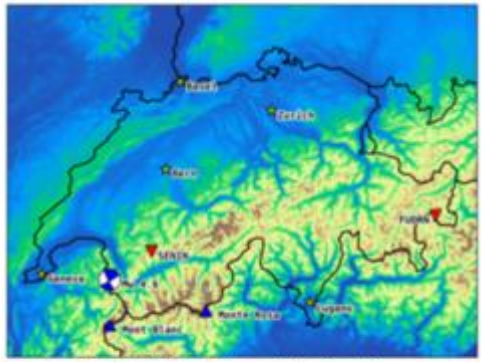
Sadaf Alam
Chief Technology Officer
Swiss National Supercomputing Centre
November 17, 2017

Piz Daint and the User Lab

http://www.cscs.ch/uploads/tx_factsheet/FSPizDaint_2017_EN.pdf
http://www.cscs.ch/publications/highlights/
http://www.cscs.ch/uploads/tx_factsheet/AR2016_Online.pdf

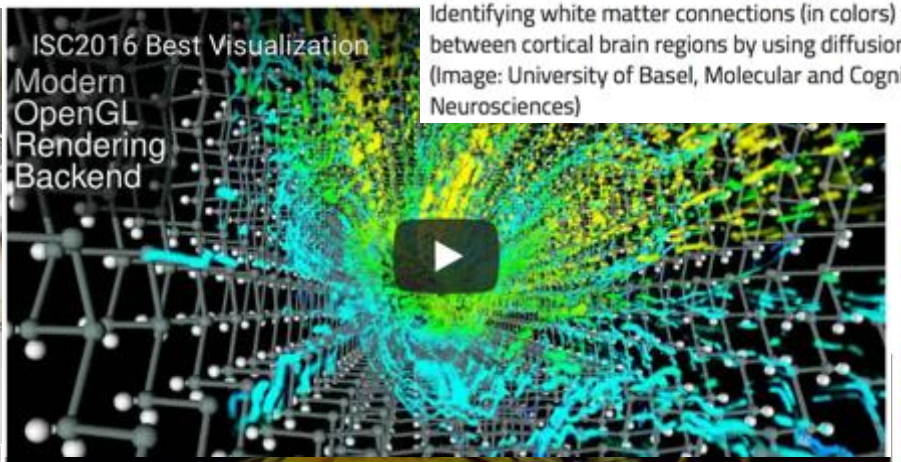


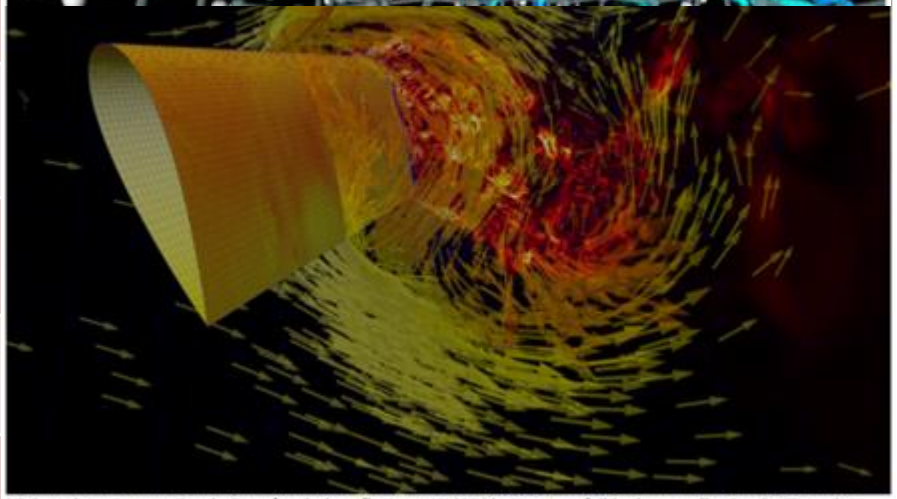
Identifying white matter connections (in colors) between cortical brain regions by using diffusion MRI. (Image: University of Basel, Molecular and Cognitive Neurosciences)



Earthquake simulation setup, with a Mw 4.6 earthquake in the Swiss-French Alps and two recording stations FUORN and SENIN. (Image: Rietmann et al.)

Model	Cray XC40/XC50
XC50 Compute Nodes	Intel® Xeon® E5-2690 v3 @ 2.60GHz (12 cores, 64GB RAM) and NVIDIA® Tesla® P100 16GB
XC40 Compute Nodes	Intel® Xeon® E5-2695 v4 @ 2.10GHz (18 cores, 64/128 GB RAM)
Interconnect Configuration	Aries routing and communications ASIC, and Dragonfly network topology
Scratch capacity	~9 + 2.7 PB



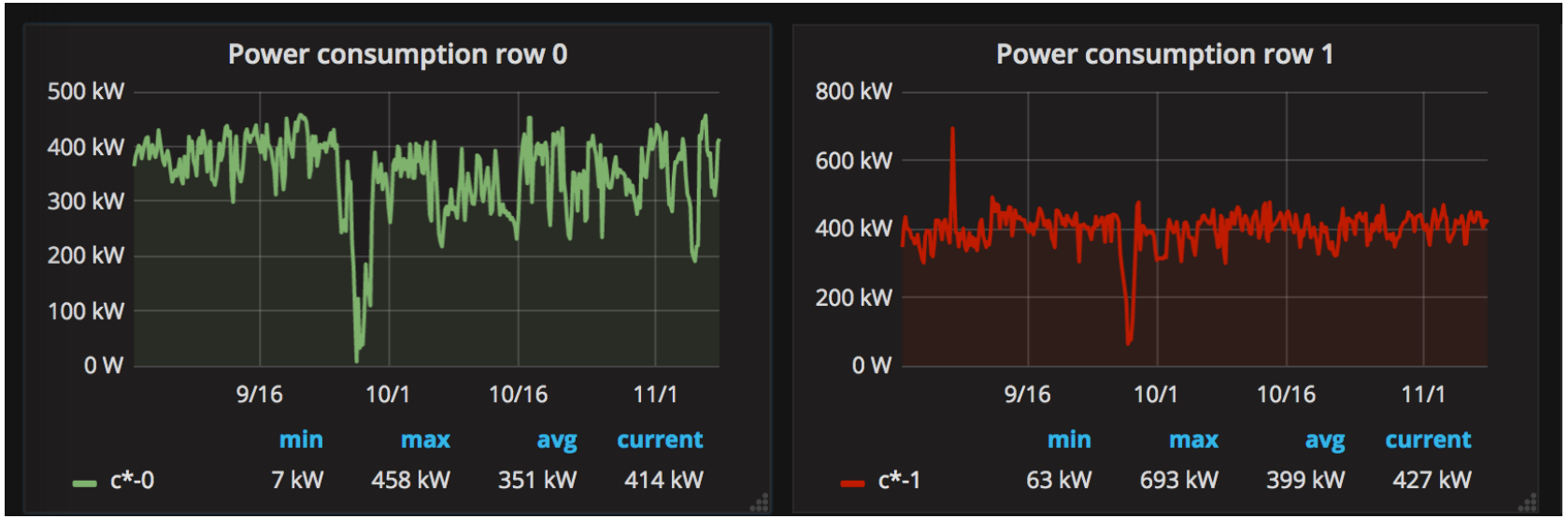


High-order accurate simulation of turbulent flow over a NACA0021 aerofoil in deep stall using PyFR on Piz Daint. (Image: Peter Vincent)

The image shows the clathrate structures that the researchers discovered using "Piz Daint"; these structures could be suitable for manufacturing more efficient solar cells. However, laboratory trials must first be carried out to show whether they can be synthesised. (Image: http://dx.doi.org/10.1103/PhysRevB.92.014101)

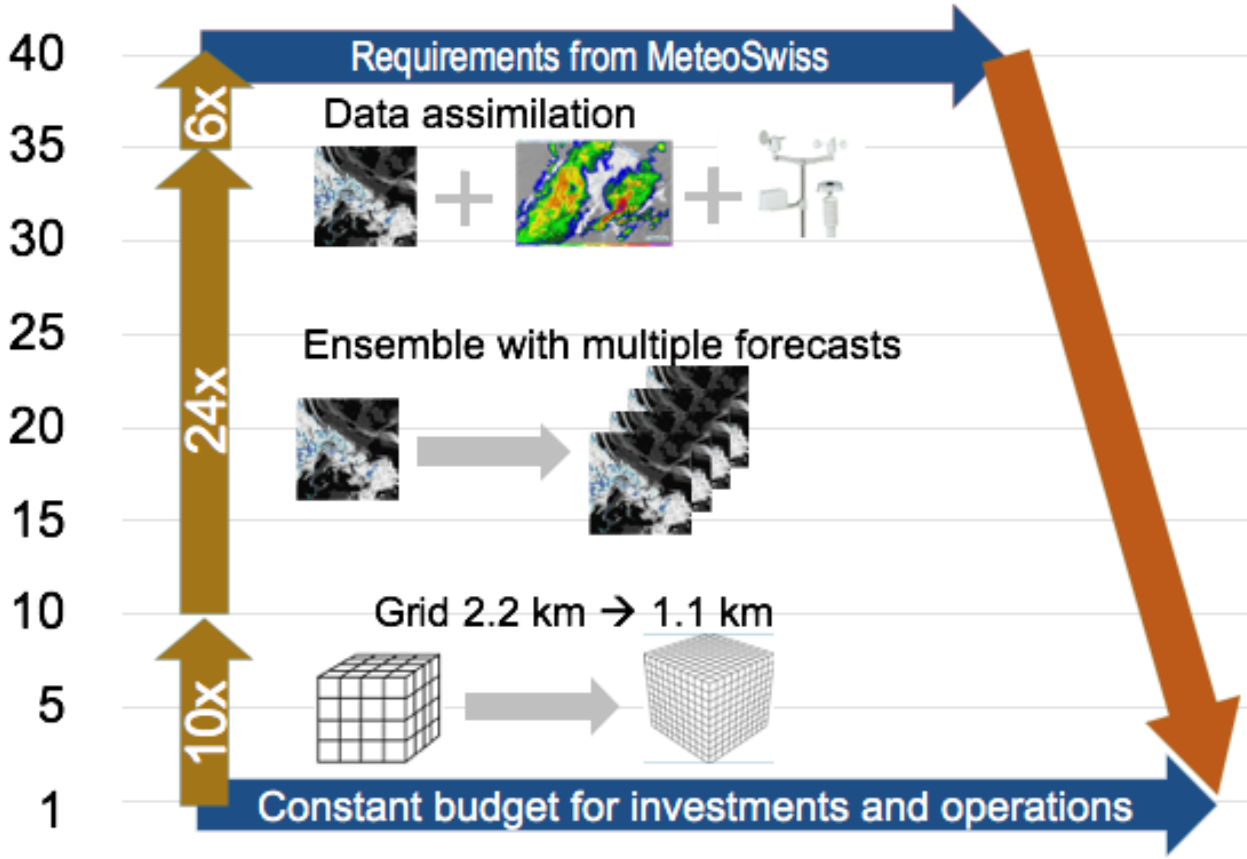
Measurement Tools on Piz Daint

- Level 3 measurements for official submissions
  - Accuracy for Top500, Green500, …
- Grafana dashboard
  - Overall system and cabinet row
- Cray performance measurement database (PMDB)
  - DB with node, blade, cabinet, job, …
- SLURM output
  - Using data available on compute nodes

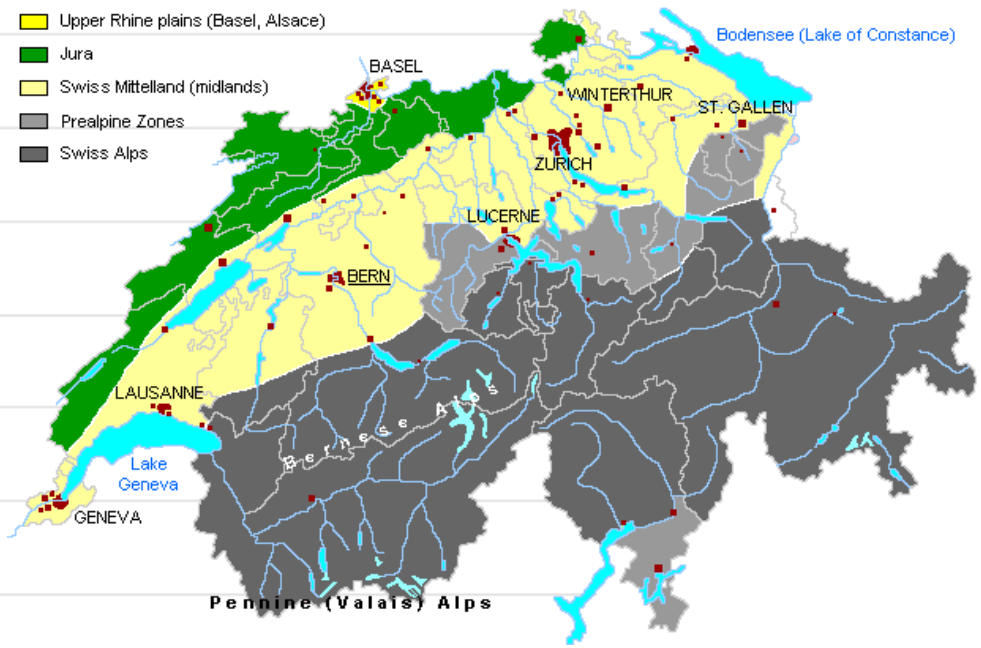



```
> sacct -j 3****** -o JobID%20,ConsumedEnergyRaw
               JobID ConsumedEnergyRaw
-------------------- ----------------
             3******
       3******.batch     379848.000000
      3******.extern 2005970432.000000
          3******.0 2001926407.000000
```

MeteoSwiss' Performance Ambitions in 2013

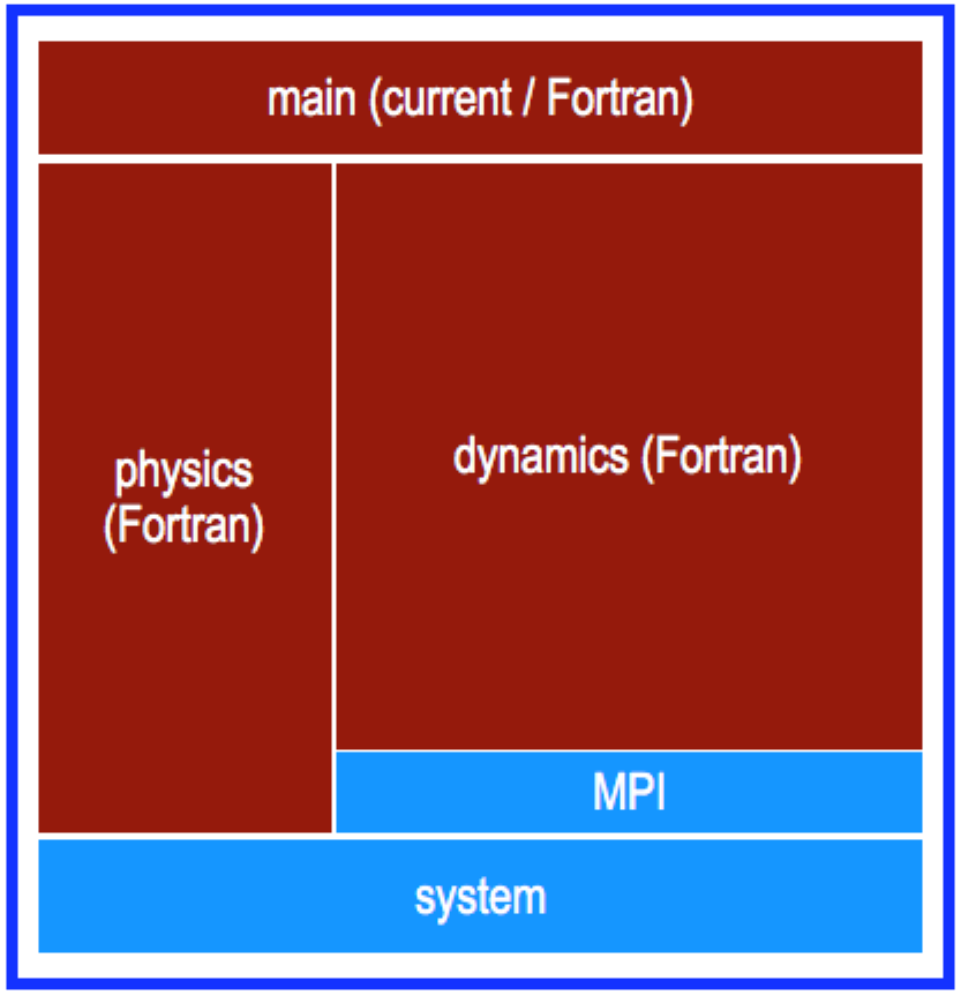


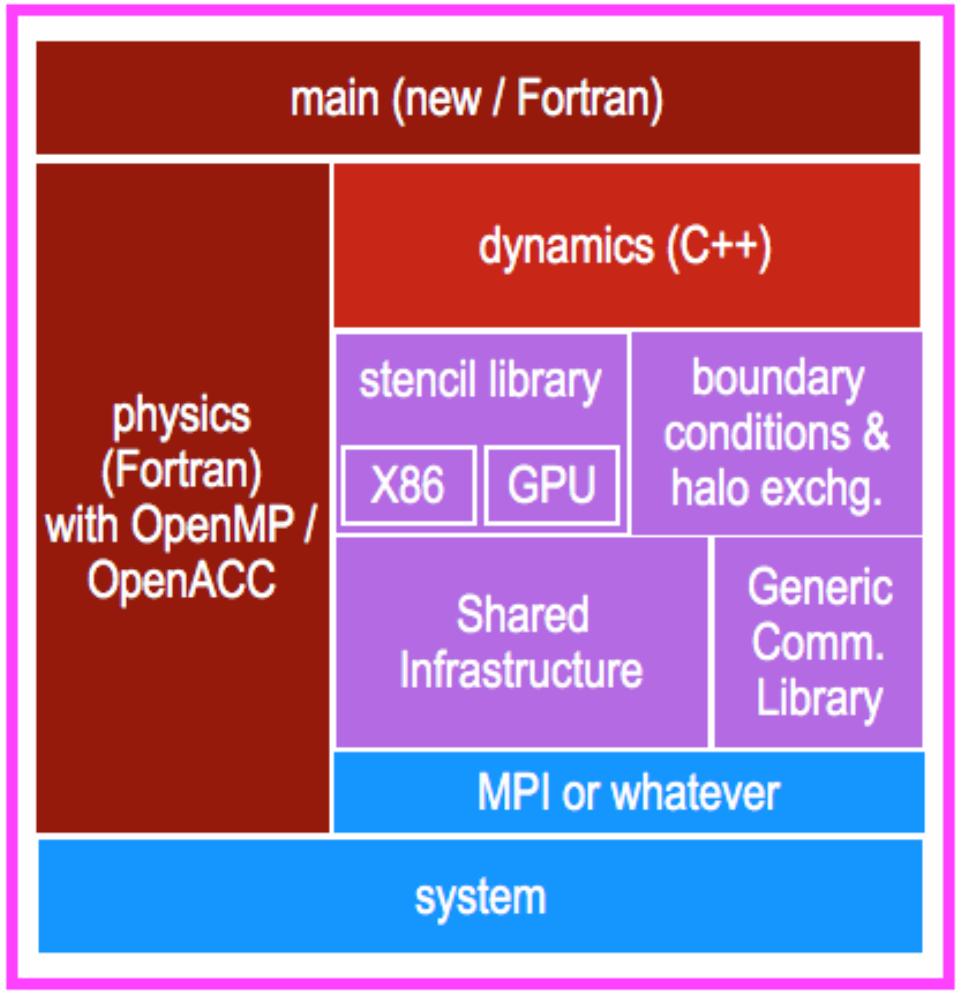
?



We need a 40x improvement between 2012 and 2015 at constant cost

COSMO: old and new (refactored) code





HPC wire

September 15, 2015

Today's Outlook: GPU-accelerated Weather Forecasting

John Russell

MeteoSwiss New Weather Supercomputer

World's First GPU-Accelerated Weather Forecasting System

2x Racks

48 CPUs

192 Tesla K80 GPUs

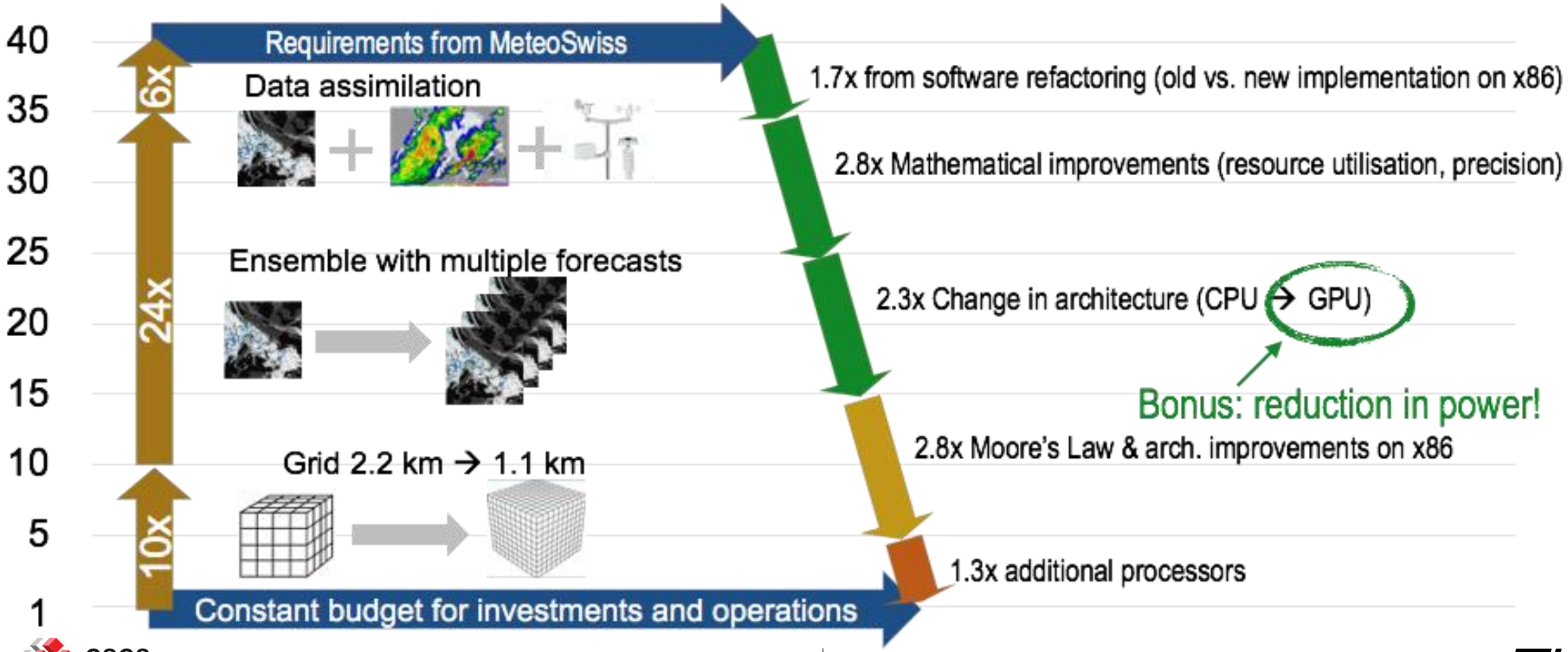
> 90% of FLOPS from GPUs

Operational in 2016

5 NVIDIA

Where the factor 40 improvement came from

Investment in software allowed mathematical improvements and change in architecture



Piz Daint - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100

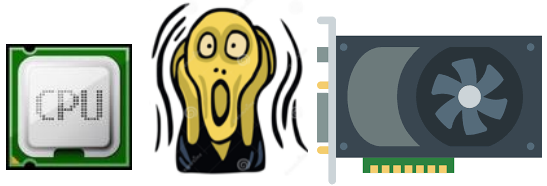
Site:	Swiss National Supercomputing (
System URL:	http://www.cscs.ch/computers/pi
Manufacturer:	Cray Inc.
Cores:	361,760
Memory:	340,480 GB
Processor:	Xeon E5-2690v3 12C 2.6GHz
Interconnect:	Aries interconnect
Performance	
Linpack Performance (Rmax)	19,590 TFlop/s
Theoretical Peak (Rpeak)	25,326.3 TFlop/s
Nmax	3,569,664
HPCG [TFlop/s]	486.398
Power Consumption	
Power:	2,271.99 kW (Optimized: **1631.13** kW)
Power Measurement Level:	3

NOVEMBER 2017

In the latest GREEN500 list, the top three positions are taken by newly installed systems in Japan, all of which are based on the ZettaScaler-2.2 architecture and the PEZY-SC2 accelerator. The SC2 is a second-generation 2048-core chip that provides a peak performance of 8.192 teraflops in single-precision.

The most efficient of these ZettaScaler supercomputers is the Shoubu system B installed at RIKEN's Advanced Center for Computing and Communication. It achieved a power efficiency of 17.0 gigaflops/watt.

- Green500 top systems contain accelerator devices



- Only system in top 10 with level 3 submission
- Subset of cores with lower CPU frequency for Green500 submission
- Engaging users and developers

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Thank you for your attention.

Four Steps to Energy Efficiency Through Software

William Gropp
Wgropp.cs.illinois.edu

What was the biggest contribution to HPC System Energy Efficiency?

- Biggest contribution to HPC system energy efficiency was cooling. System uses air cooling within cabinets, with water removing heat from each cabinet. Heat exchangers on roof provide “free” cooling most of the year.
- No impact on running application. No software changes required.
- Problem: Our PUE is between 1.1 and 1.2; little further gain in efficiency possible here

National Petascale Computing Facility

- Only Facility in the world of this scale on an Academic Campus
- Capable of sustained 24 MW today
- Expandable in space, power and cooling [50,000 ft^2 (4,645+ m^2) machine room gallery and sustained 100 MW]

- Modern Data Center
  - 90,000+ ft^2 (8,360+ m^2) total
  - 30,000 ft^2 (2,790+ m^2) raised floor
    20,000 ft^2 (1,860+ m^2) machine room gallery

- Energy Efficiency
  - LEED certified Gold
  - Power Utilization Efficiency, PUE = 1.1–1.2

How Does Your Center Contribute Most in Energy Efficiency?

- There are still be gaps in the baseline – the efficiency of applications. This is why “race to halt” is (was?) often the best strategy.
- Our focus remains on improving application efficiency – including moving some applications to use GPUs on our mixed XE6/XK7 system
- There are many parts to this
  - One of the most difficult yet most important is having enough of a performance model to understand how close the application is to achievable performance
  - This is different from efficiency of execution based on counting operations (this often over-estimates the number of necessary data moves, leading to a lower estimate for achievable performance)
  - Other steps, such as allocating jobs with topology awareness, are important

Where Should Energy-Awareness Be Added to the Software Stack?

- Easiest if in runtime, scheduler, or closed application
  - That is, easiest if programmers don't see it
- Answer needs to be in context of goal
  - Information gathering (such as we did for I/O performance) including interconnect contribution and message wait (opportunity to reduce power) can be done with limited impact using MPI profiling interface or similar system intercept interfaces
- If runtime can "do no harm", then do it in the runtime
  - If you can't guarantee that, allow users to opt out.
  - See discussion of incentives

Will Users Make Codes Energy Efficient?

- No.
- There is no incentive, and in fact, there is usually a negative incentive: if energy efficiency slows down the application
- Some users may be willing to experiment or to be good “global citizens,” but most are focused on their science.
- Different charging policies must be approved; in our case, by NSF.
  - For example, could charge allocation with a weight that included energy consumed, providing an incentive. But must include full cost of system including depreciation

Does the Community Need to Collect and Share Data?

- Yes. To provide better incentives, there is a need to both measure energy use and provide actionable feedback to users. We need to to collect data on different approaches and reflect them in the context of user incentives
- There are many complexities in this, and it will be hard to provide accurate data about energy use by application
  - This is not the same as having a deterministic way to *charge* for energy use.
  - The problem is that the actions of other applications may impact the energy use of shared resources (e.g., network, file system) and it may not be possible to determine what each application really **used**, as compared with what it was **charged**
  - Yes, this is also true for time charged to applications. That is not the issue; it is whether the users accept the approximations that are used

Four Steps for Energy Efficiency

1. Performance Efficiency
   - Like conservation, first step is to get more from the same (or less)
     - Performance modeling
     - Performance tuning
2. Recognize Hardware Realities
   - Next is to adapt to realities of the hardware
     - Rethink the algorithm
       - Adapt to memory latency, wide vectors, etc.
       - Can’t compare algorithms by comparing floating point counts, or even memory references
     - Avoid BSP-style programs
       - Separate communication and computation phases, and synchronization, waste energy

Four Steps (con't)

3. You Get What You Measure

- You get what you measure (and reward/penalize)
  - Again, focus on performance
  - Expose the cost of waiting

4. Consider The Total Cost

- Finally, need to consider total workflow costs
  - Local and global optimums not necessarily the same
  - Consider data motion for a data set – moving data back and forth can be expensive – even (locally) less efficient in situ analysis may be cheaper overall
- And how that total cost is exposed/charged to the user
  - There must be a *positive* incentive *to the user* for pursuing energy efficiency

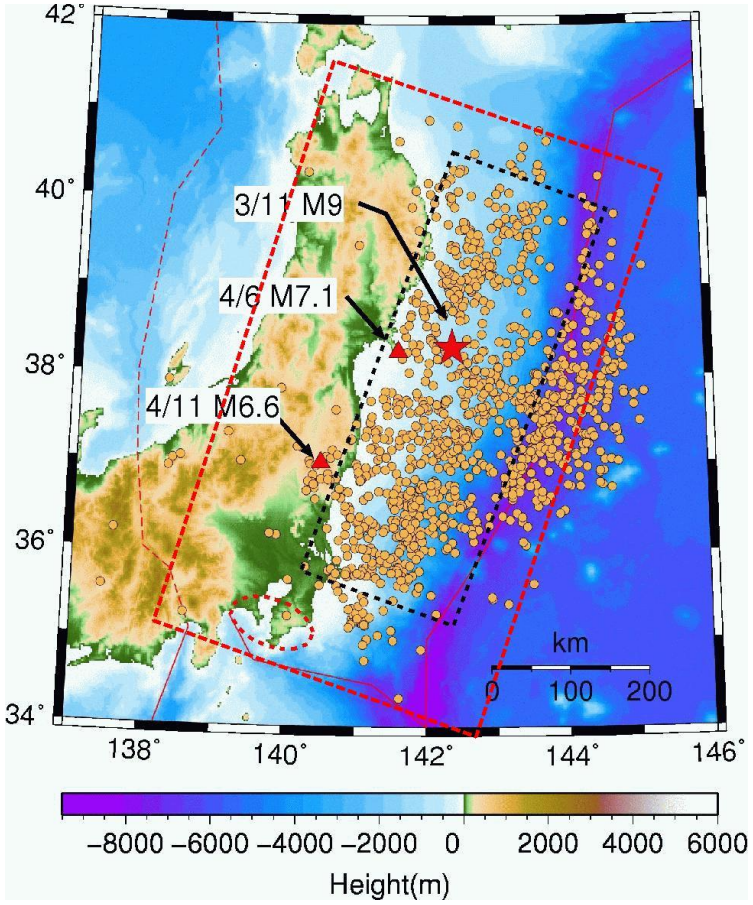
Green Computing in Tsubame2 and 3

Satoshi Matsuoka
Professor, GSIC, Tokyo Institute of Technology /
Director, AIST-Tokyo Tech. Big Data Open Innovation Lab /
Fellow, Artificial Intelligence Research Center, AIST, Japan /
Vis. Researcher, Advanced Institute for Computational Science, Riken

SC17 Green Panel Talk
2017/11/17
Denver, Colorado, USA

Impacts of Mar 11 Earthquake

Operation of TSUBAME2.0 started in Nov 2010, and The Great East Japan Earthquake happened 4 months later



- Earthquake itself did not affect the supercomputer, but…
- Power plant accident caused power crisis in the capital area
  - Rolling outage in March
  - University/Government's direction to save power

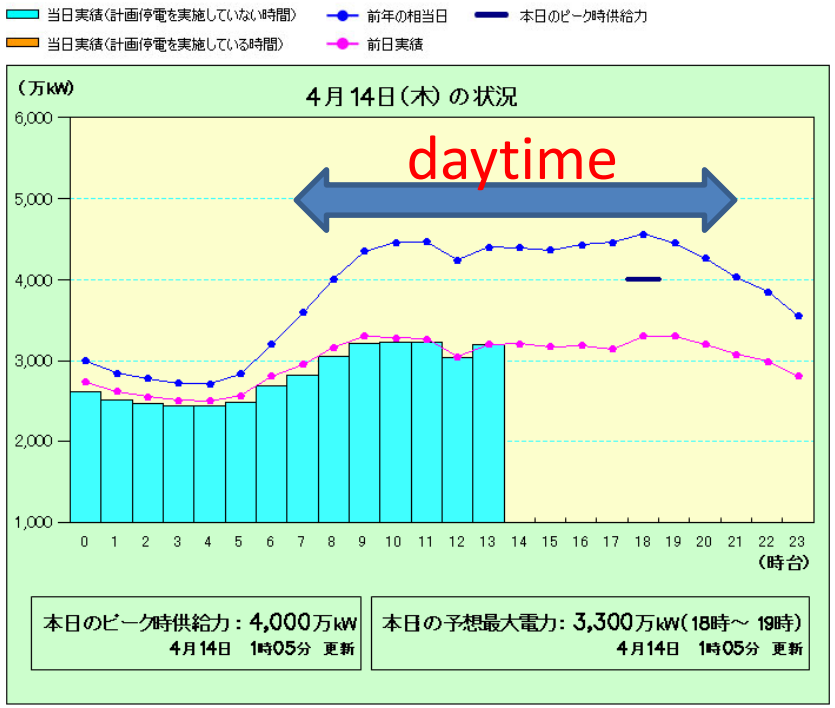
- Although TSUBAME2 is "green", it consumes ~10% (0.9—1.2MW) power of the campus!

TSUBAME Power Operation: 2011 ver. (1)

- University directed us to save power usage by:
  - -50% in early April
  - → # of nodes has to be -70%
  - -25% in late April
  - → # of nodes has to be -35%
- Reducing too much nodes constantly is harmful for supercomputer users
- What is the social demand?

→ The most important constraint is to reduce peak power consumption in daytime

→ "Peak shift" operation



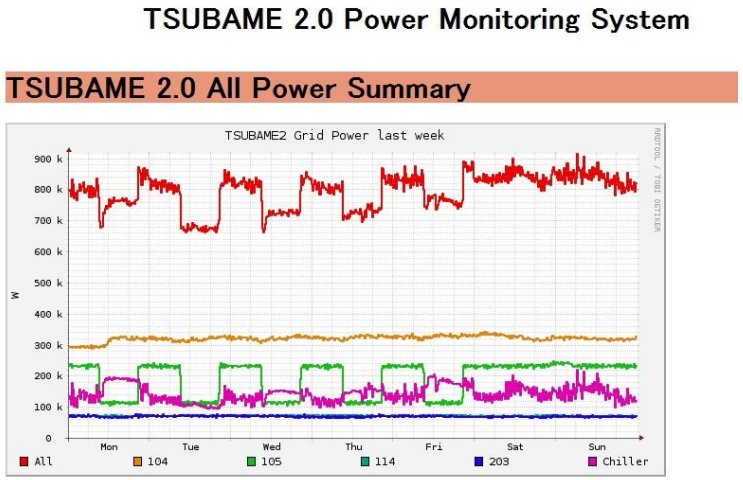
Total power usage of the capital area in a day

TSUBAME Power Operation: 2011 ver. (2)

- It was allowable to increase running nodes in nighttime and weekend

65% operation in daytime

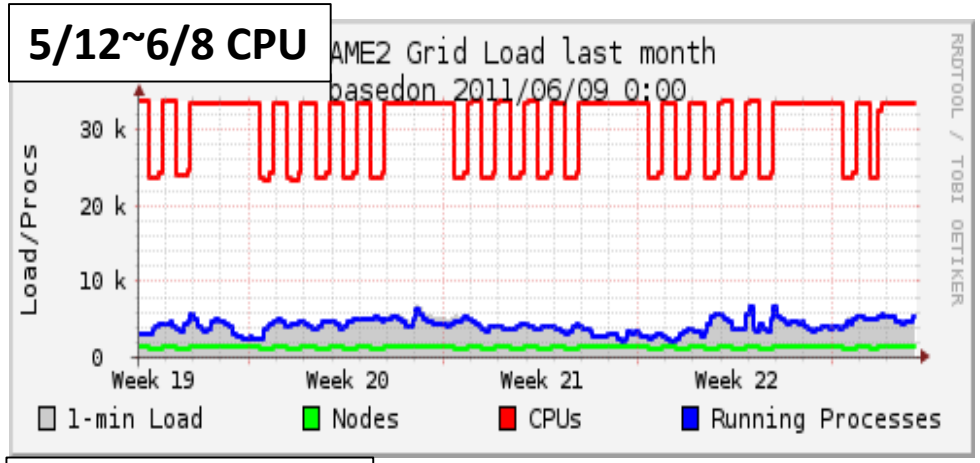
100% operation in nighttime/weekend



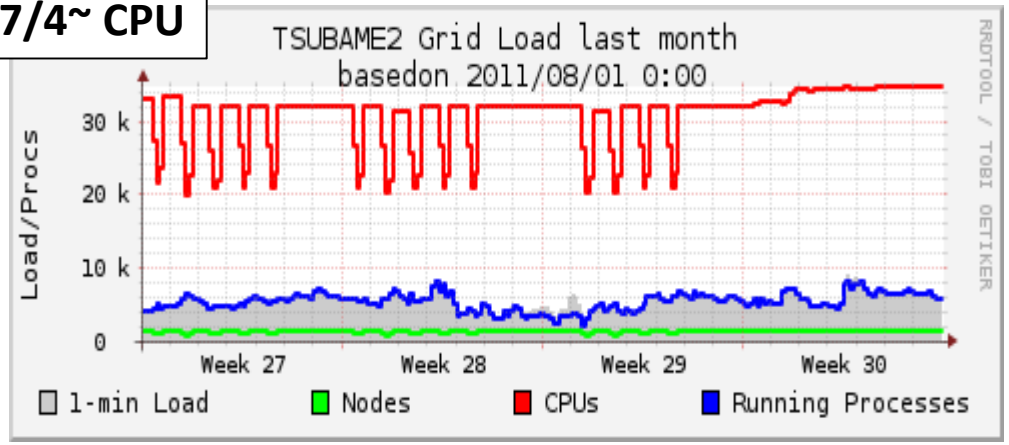
- Tokyo-Tech and NEC developed tools to shutdown/ wake up nodes automatically everyday
- The number “65%” is determined by measurement of power consumption in typical operation

Results of the Operation in 2011

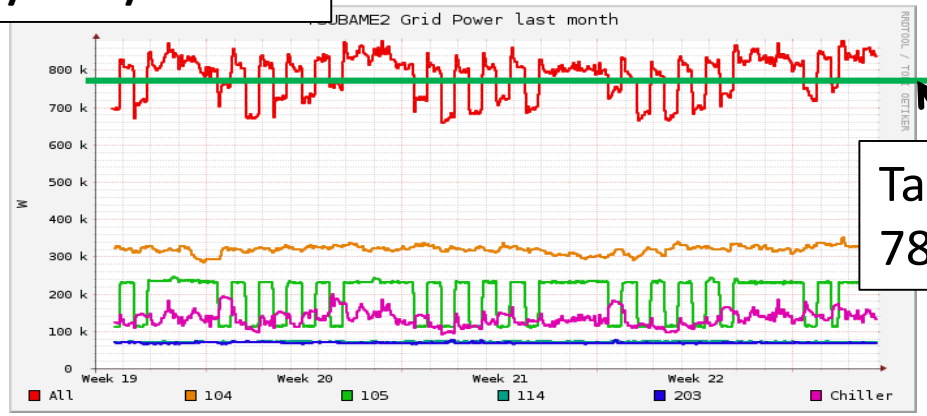
5/12~6/8 CPU



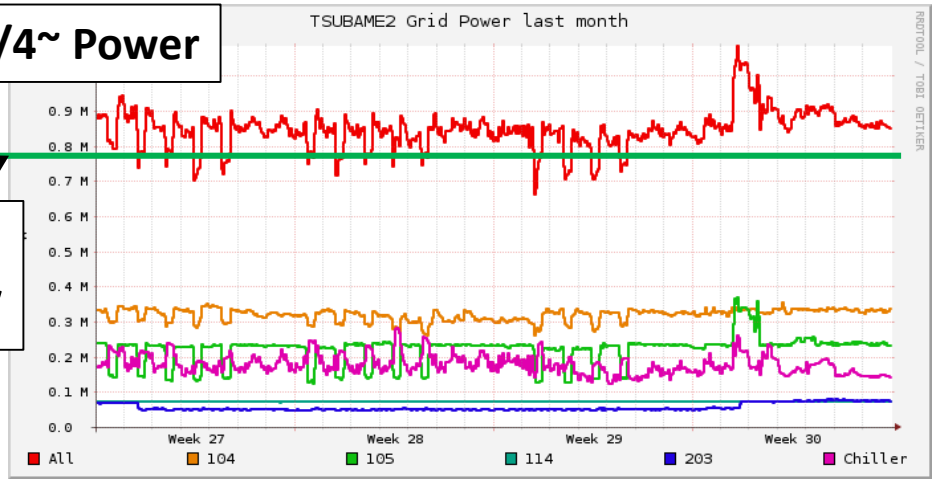
7/4~ CPU



5/12~6/8 Power



7/4~ Power



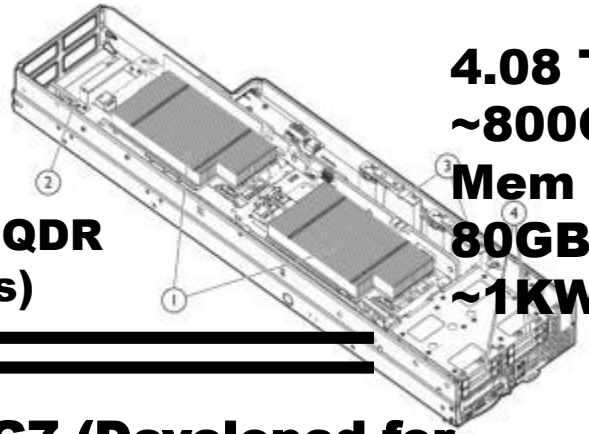
Target
787kW

- The power is successfully limited in daytime
- However, it is still sometimes too pessimistic
- Our goal is not capping # of nodes, but capping power

→ New power-aware system management method is required

TSUBAME2.0⇒2.5 Thin Node Upgrade in 2013

Thin Node



Peak Perf.

4.08 Tflops
~800GB/s Mem BW
80GBps NW
~1KW max

Infiniband QDR x2 (80Gbps)

HP SL390G7 (Developed for TSUBAME 2.0, Modified for 2.5)
GPU: NVIDIA Kepler K20X x 3
1310GFlops, 6GByte Mem(per GPU)
CPU: Intel Westmere-EP 2.93GHz x2

NVIDIA Fermi
M2050
1039/515
GFlops

NVIDIA Kepler
K20X
3950/1310
GFlops

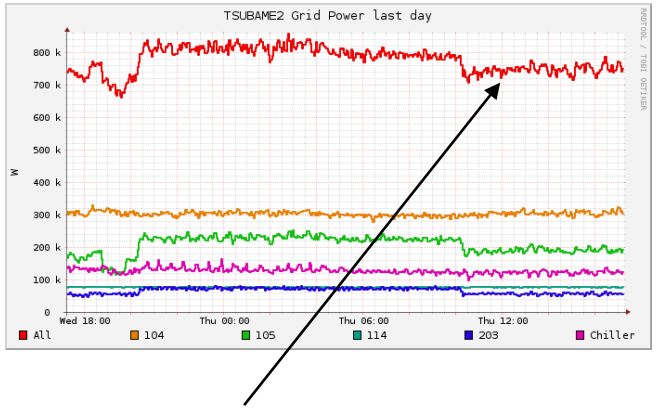
System total: 2.4PFlops → 5.7PFlops

TSUBAME Power Operation: 2014 ver.

Dynamic control is introduced based on real-time system power

(1) Real-time power consumption of the system is measured

- Including cooling

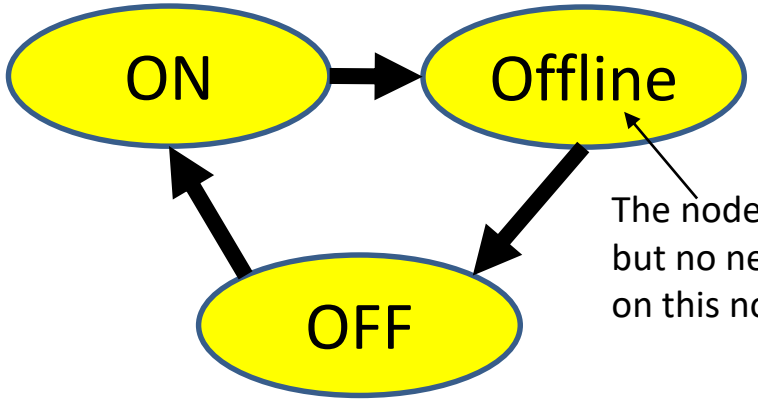


Is current power sufficiently lower than the specified cap?
Or are we in a critical zone?

(2) Current power and the target are compared

→ We control the number of working nodes dynamically

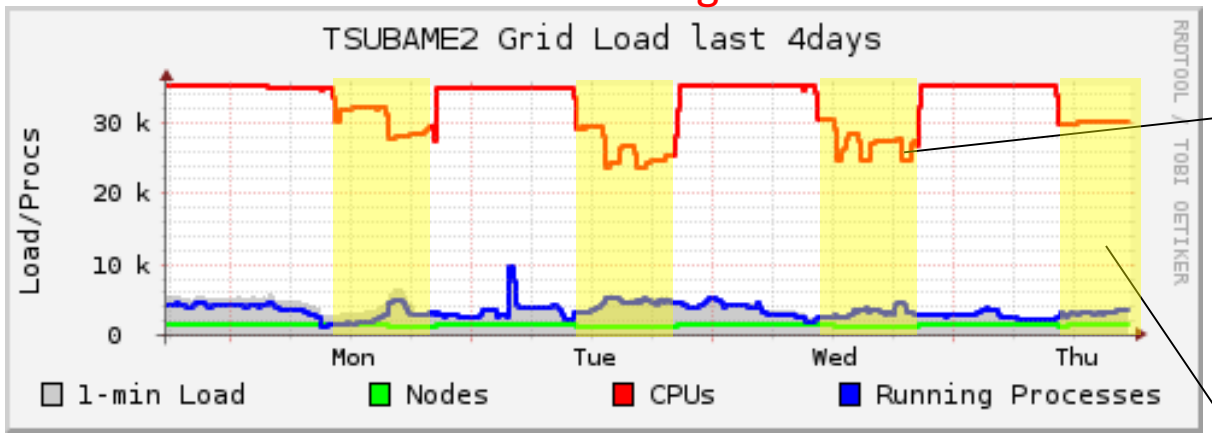
Transition of modes of each node (simplified)



The node is awaken, but no new job starts on this node

Results of the Operation in 2014

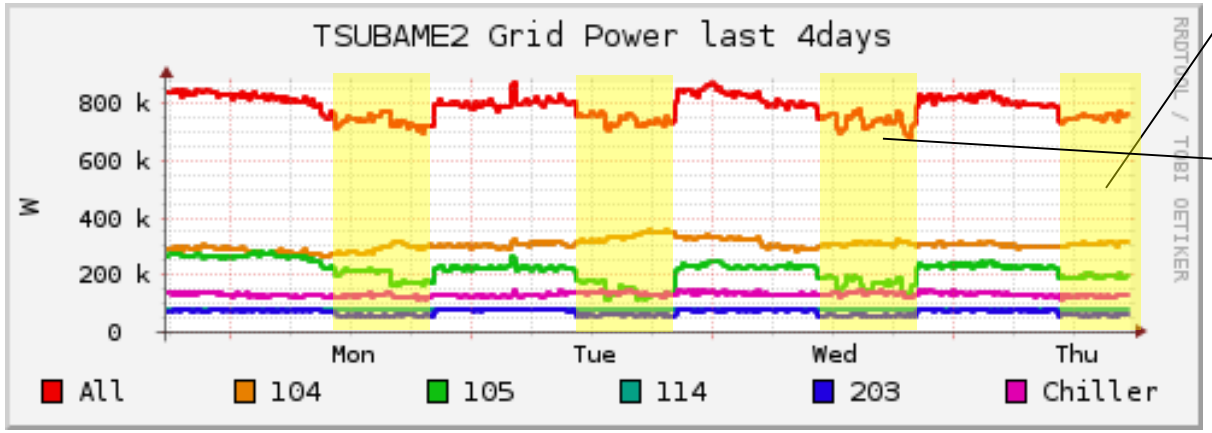
The number of working CPU cores



The number of nodes are controlled dynamically

Daytime

Power Consumption



The power is capped by 800 kW successfully

TSUBAME Power Operation: 2016 ver.

- In addition to the peak-shift (2014 ver) operation, we implemented new mechanism to reduce total energy consumption

→ Idle nodes without running jobs are shut down

  – Disadvantage: Start time of new jobs may be delayed around ten minutes for boot time of nodes

TSUBAME 2.5 Cloud Service Utilization

2017/04/18 12:33

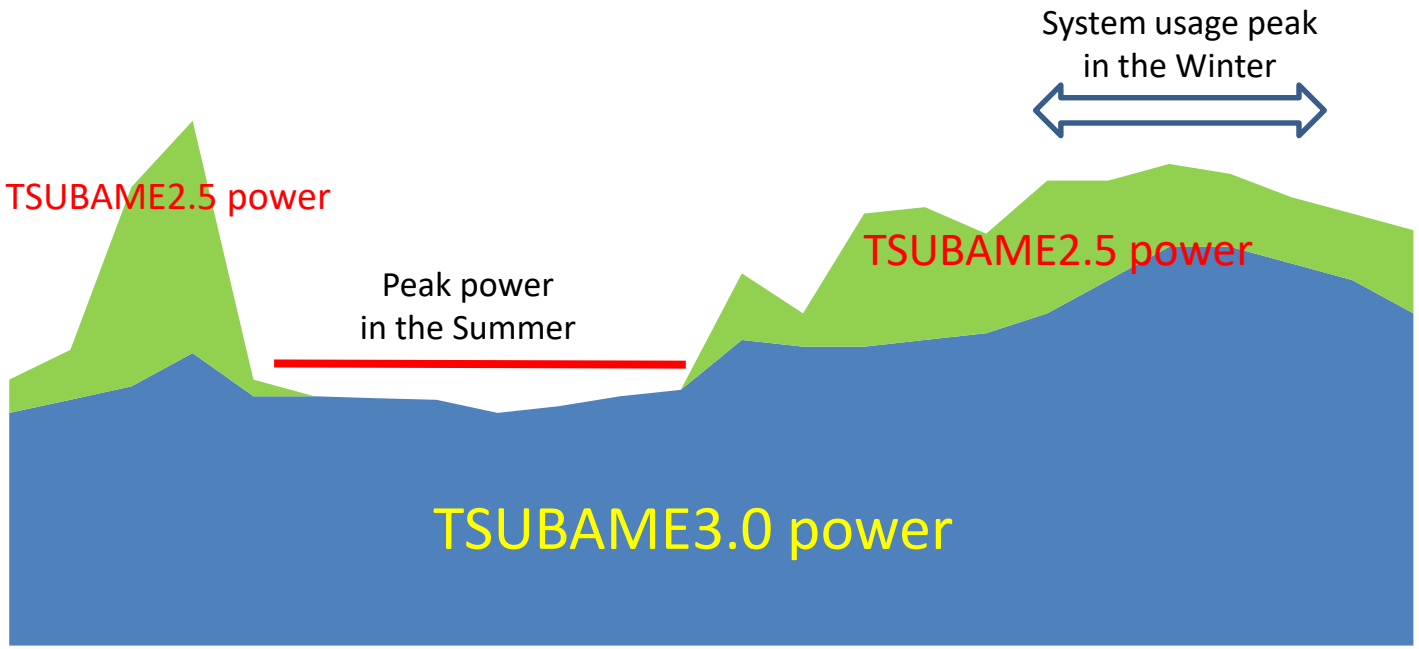
Service List

service	assigned nodes			max nodes	running jobs		users
S	15%	47 / 298 nodes		300 nodes	60%	12 / 20 jobs	6
S96	2%	1 / 39 nodes		39 nodes	100%	1 / 1 jobs	1
G	41%	180 / 431 nodes		435 nodes	100%	44 / 44 jobs	8
V	6%	25 / 361 nodes		385 nodes	97%	41 / 42 jobs	13
U	32%	10 / 31 nodes		195 nodes	100%	10 / 10 jobs	7
L256	100%	8 / 8 nodes		8 nodes	100%	8 / 8 jobs	1
L512	0%	0 / 1 nodes		2 nodes	0%	0 / 0 jobs	0
X	1%	1 / 417 nodes		420 nodes	100%	1 / 1 jobs	1
ALL	17%	272 / 1586 nodes			92%	117 / 126 jobs	36

Assigned Idle Powered off

Illustration of Power Consumption of Parallel Operation of TSUBAME3.0+2.5

Changes of power consumption during a fiscal year (April to next March)



We should do both of (1) limiting power in the summer and (2) limiting the total energy consumption within the given budget

Controlling Total Electric Fee per FY (plan)

Assumption

- System usage reaches the peak in the Winter
- Electric fee per Wh changes every month for surcharge

Strategy

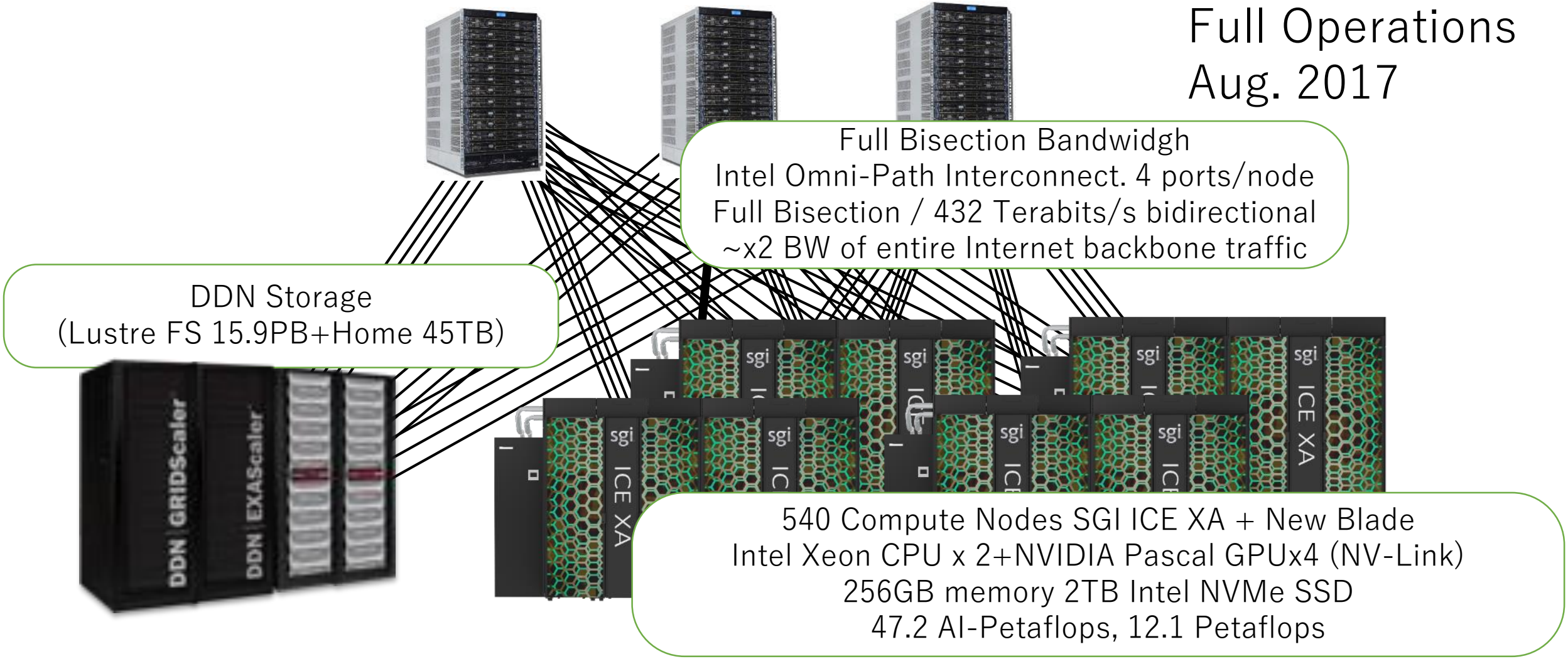
- All T3 nodes are “ON”, and T2.5 nodes become alive on-demand
- We allocate budget for each month
- Based on the month budget, we decide working T2.5 node
- When we have remaining amount, we adjust the following months

	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
x100k yen	60	65	70	50	50	120	120	120	120	120	120	120
Total MWh	288	312	360	245	220	480	500	?	?	?	?	?
T3 MWh	200	250	300	220	210	350	350	?	?	?	?	?
#T2node	400	390	380	40	20	300	360	?	?	?	?	?

Numbers are tentative

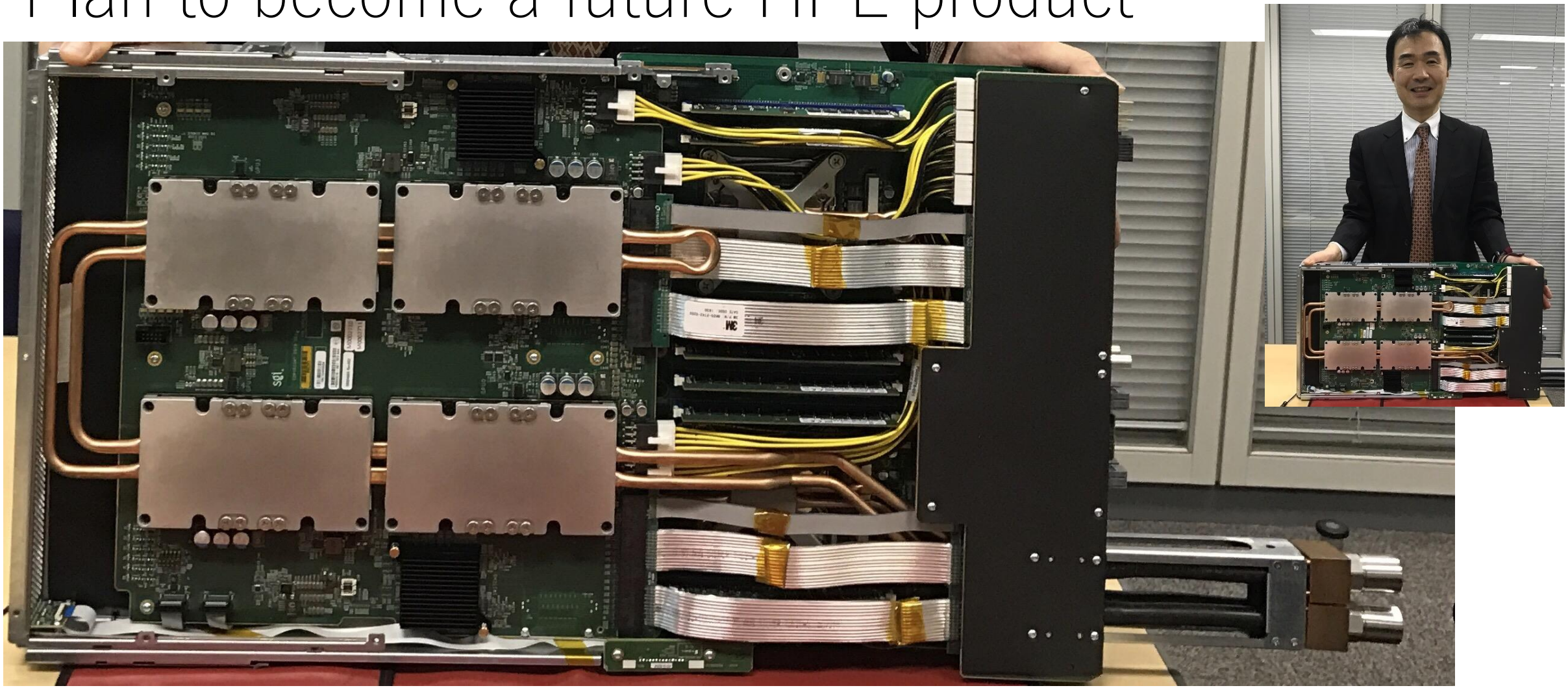
Overview of TSUBAME3.0
BYTES-centric Architecture, Scalaibility to all 2160 GPUs, all nodes, the entire memory hiearchy

Full Operations
Aug. 2017



TSUBAME3.0 Co-Designed SGI ICE-XA Blade (new)

- No exterior cable mess (power, NW, water)
- Plan to become a future HPE product

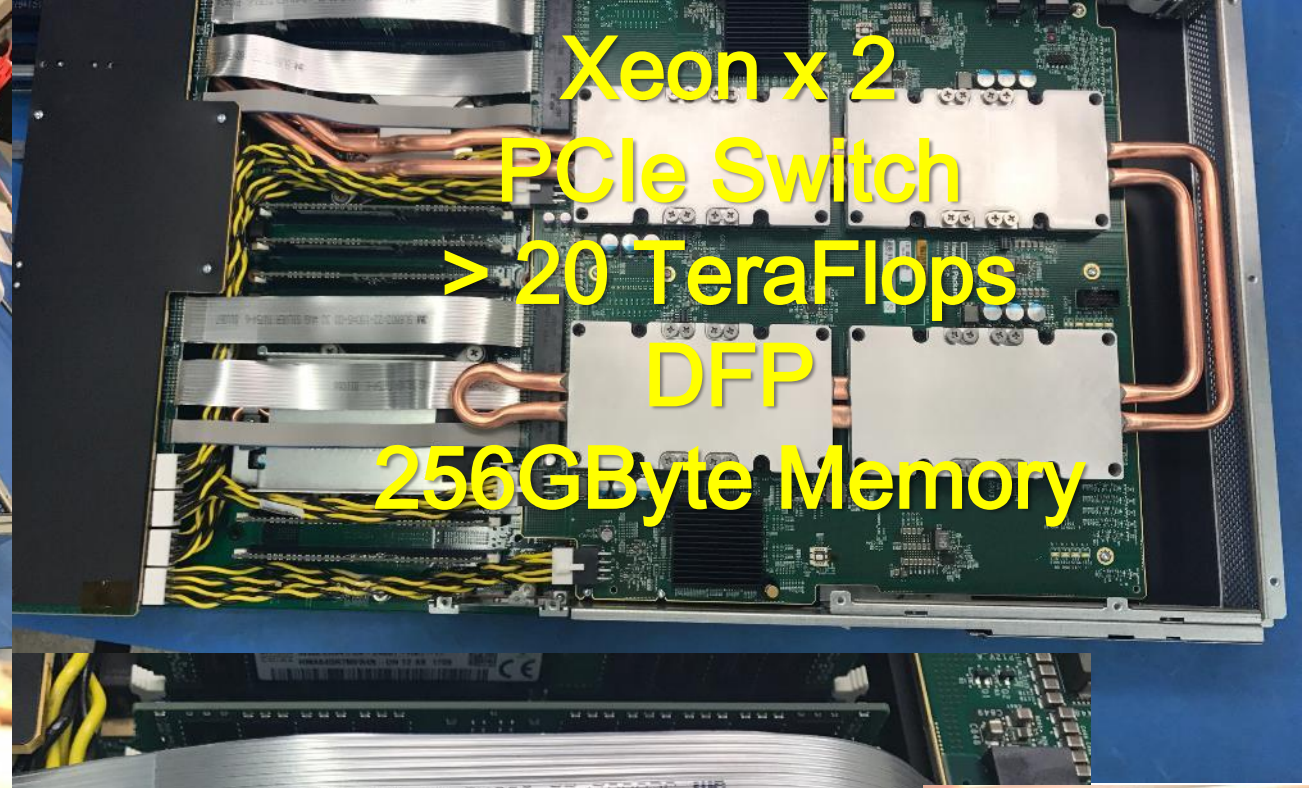


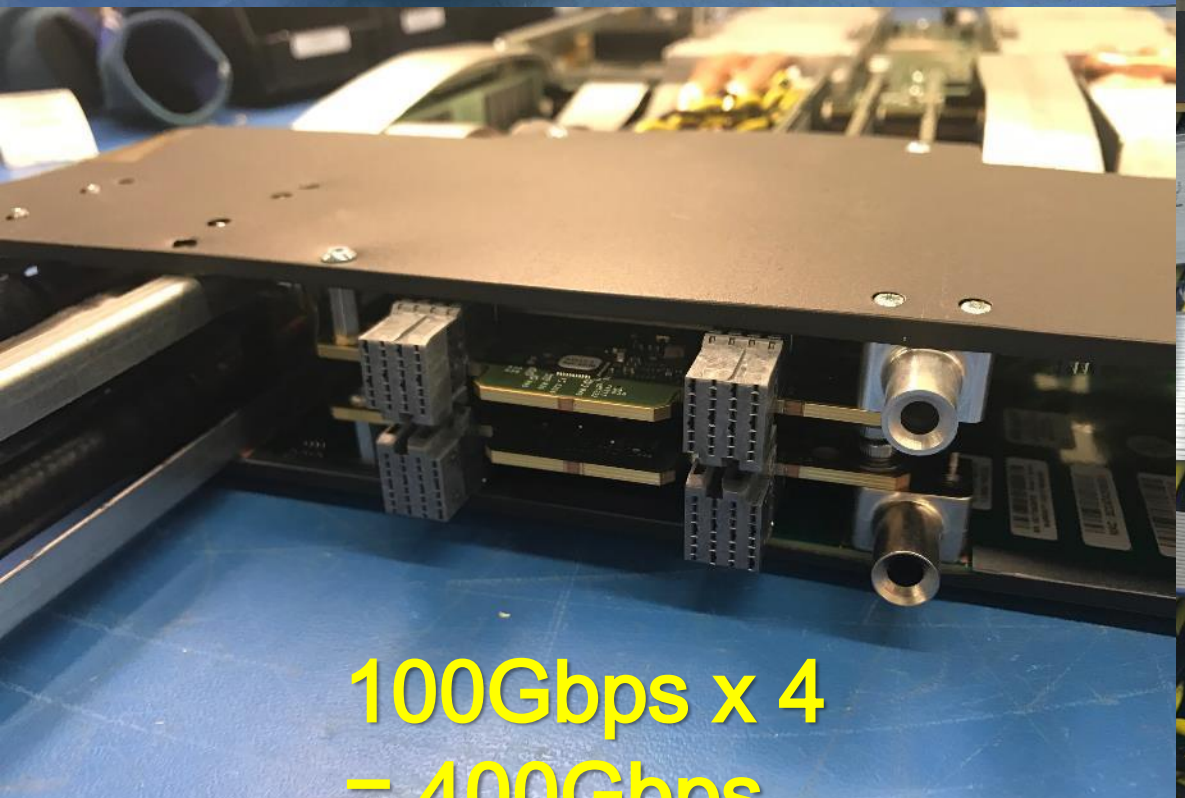
Running Benchmarks (HPL, HPCG)
Tokyo Tech + HPE/SGI Japan Team

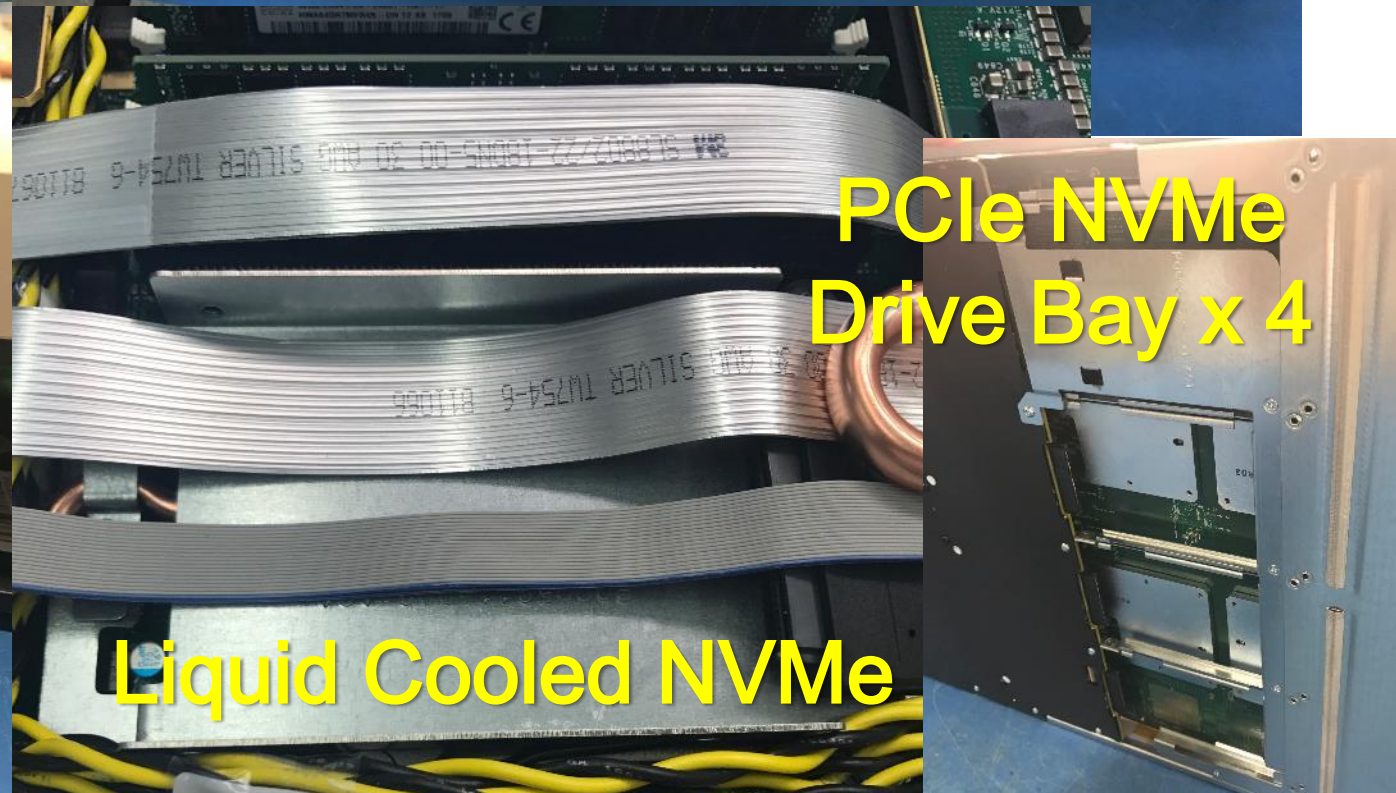
Top 500 Submission =
1.x Petaflops @ 144 Nodes

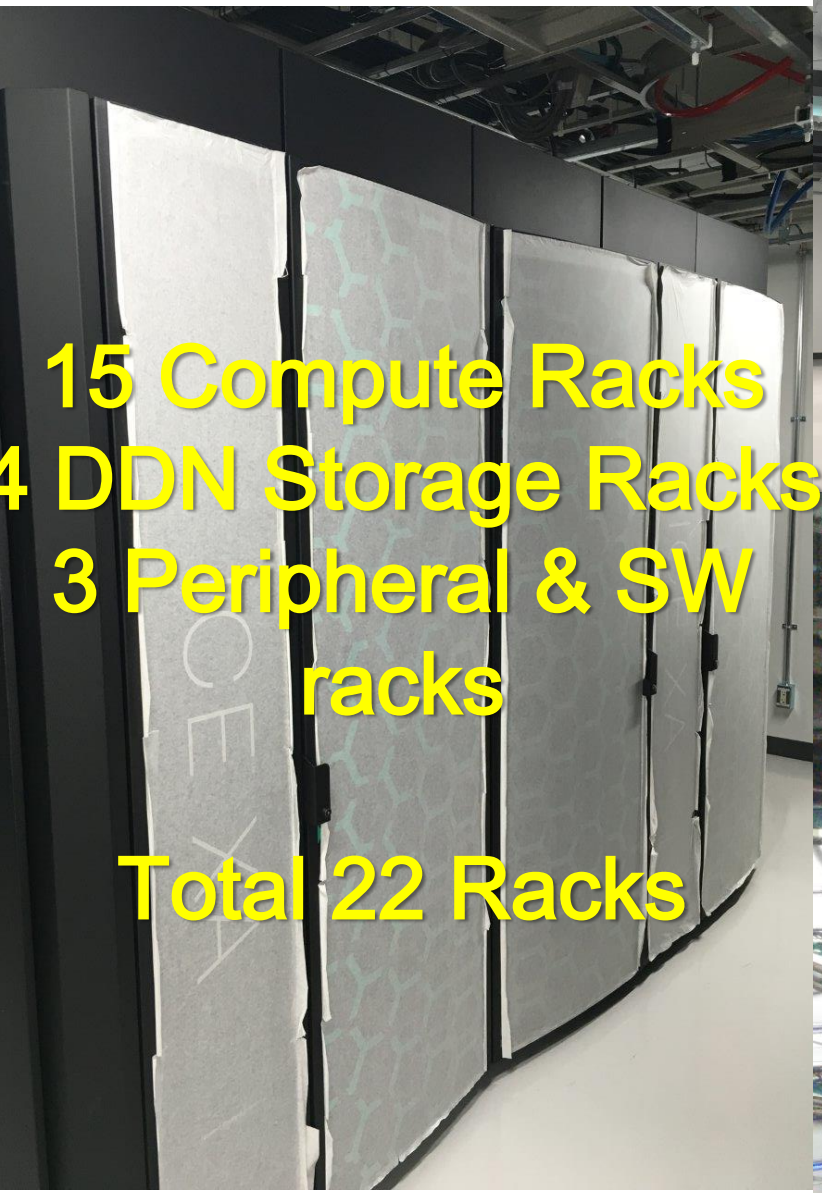
Green 500 Submission =
14.11 Gflops/W (Nov. 2016 #1 = 9.46 Gflops/W)
Announcement ISC17 Opening,
9AM Monday, 19th June World #1 !!!

Liquid Cooled
“Hot Pluggable” ICE-XA Blade
Smaller than 1U server, no cables or pipes


Xeon x 2
PCIe Switch
> 20 TeraFlops DFP
256GByte Memory

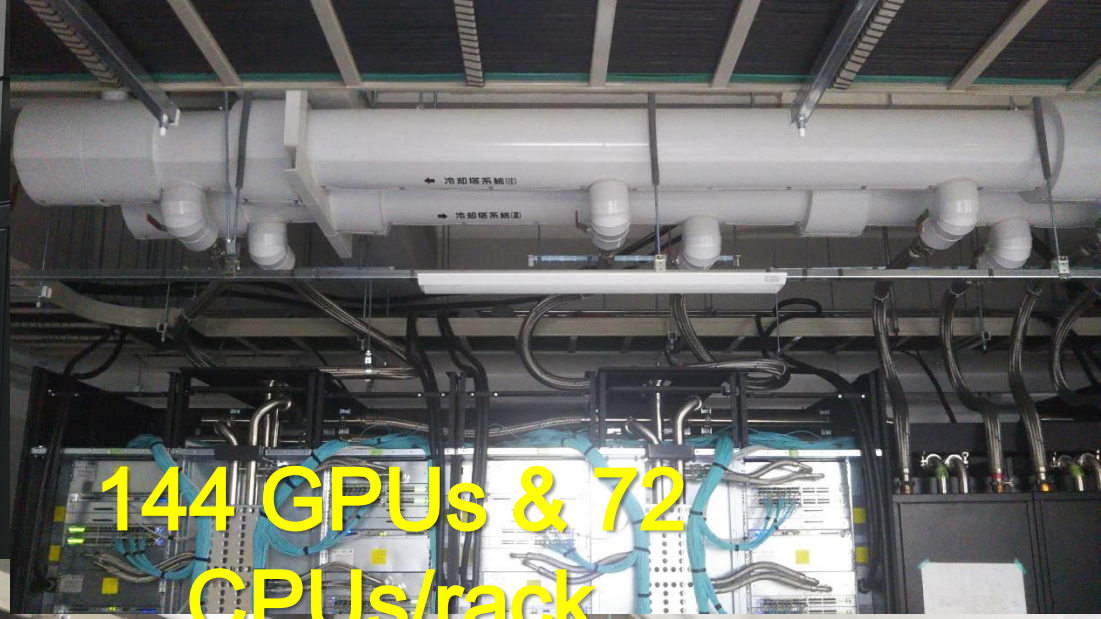

100Gbps x 4
= 400Gbps


PCIe NVMe
Drive Bay x 4
Liquid Cooled NVMe

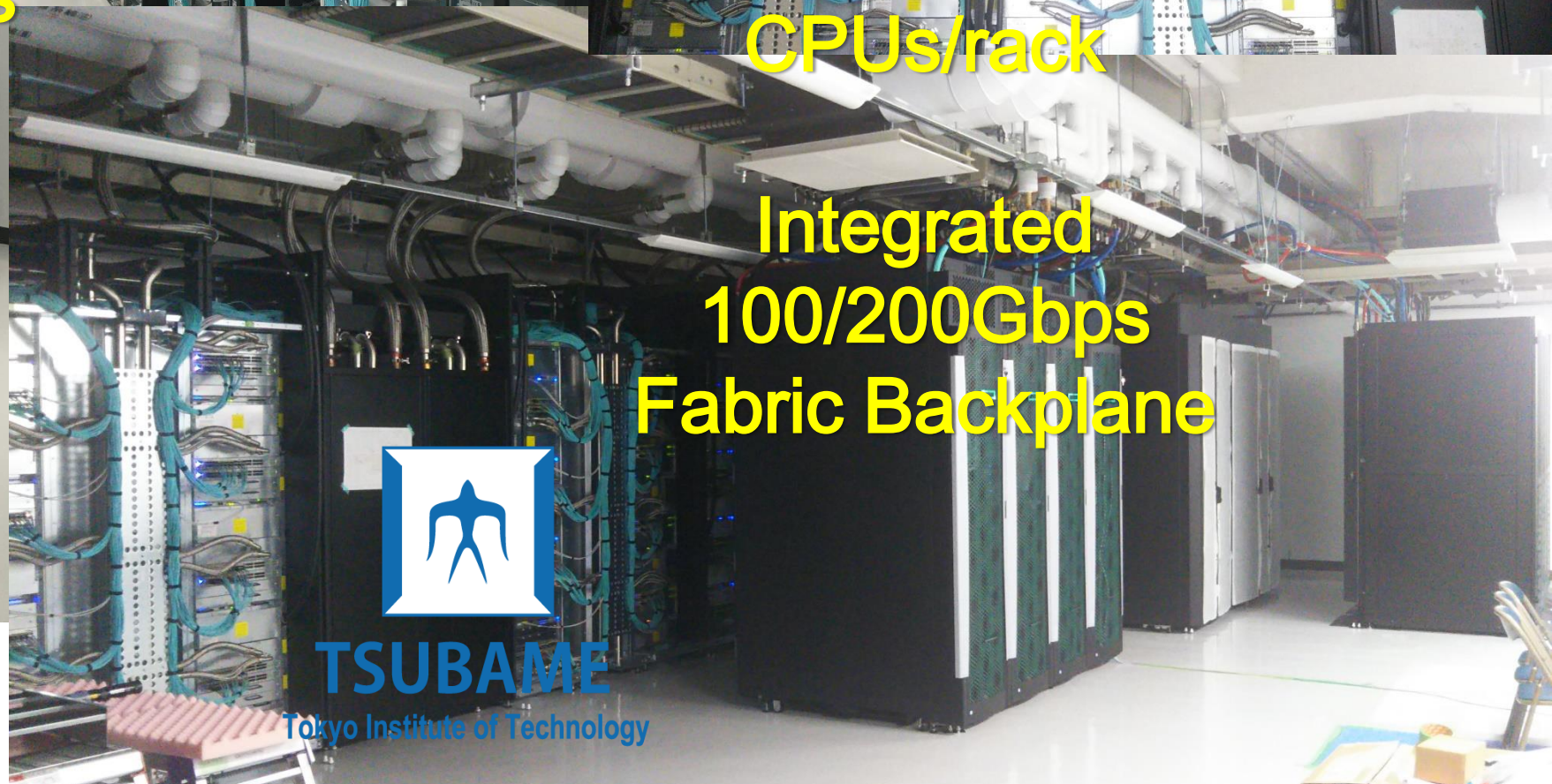


15 Compute Racks
4 DDN Storage Racks
3 Peripheral & SW racks

Total 22 Racks



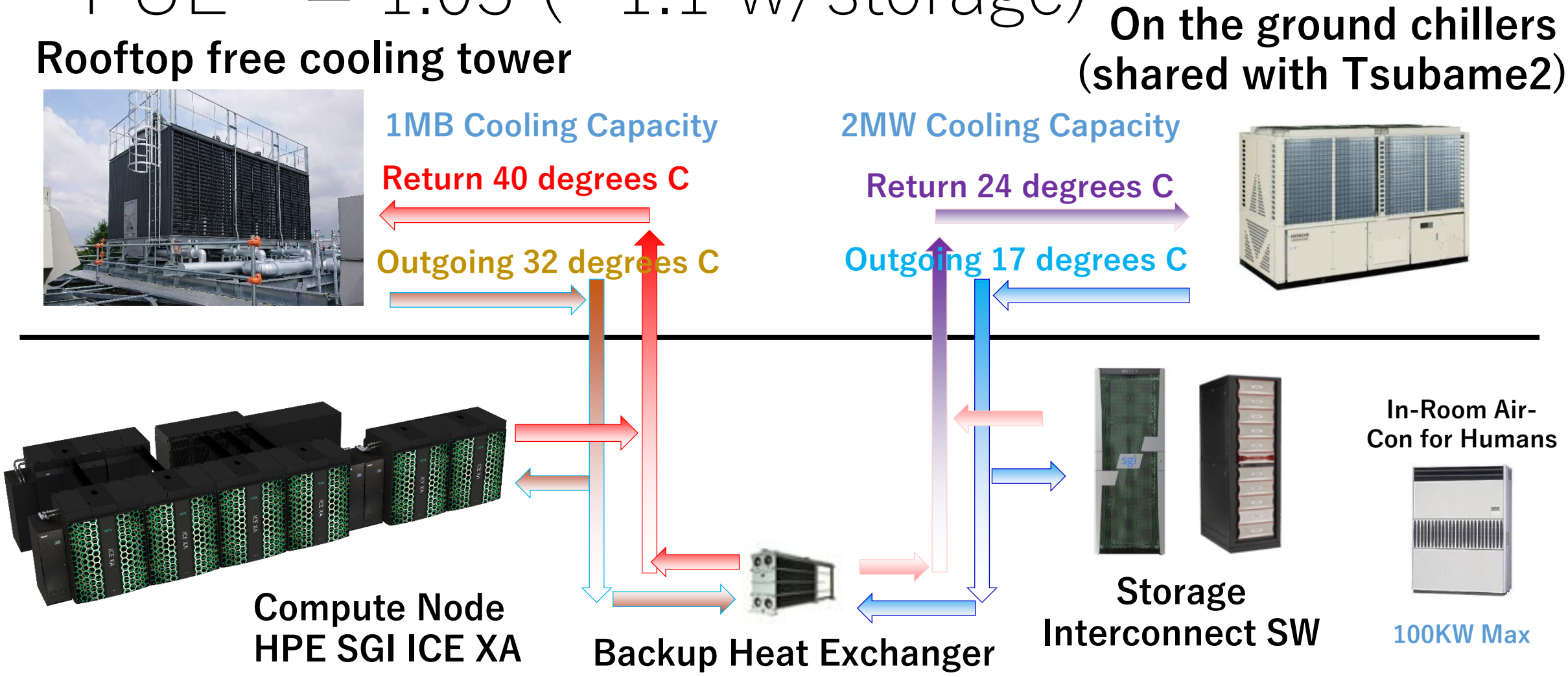
144 GPUs & 72 CPUs/rack



Integrated 100/200Gbps Fabric Backplane

TSUBAME
Tokyo Institute of Technology

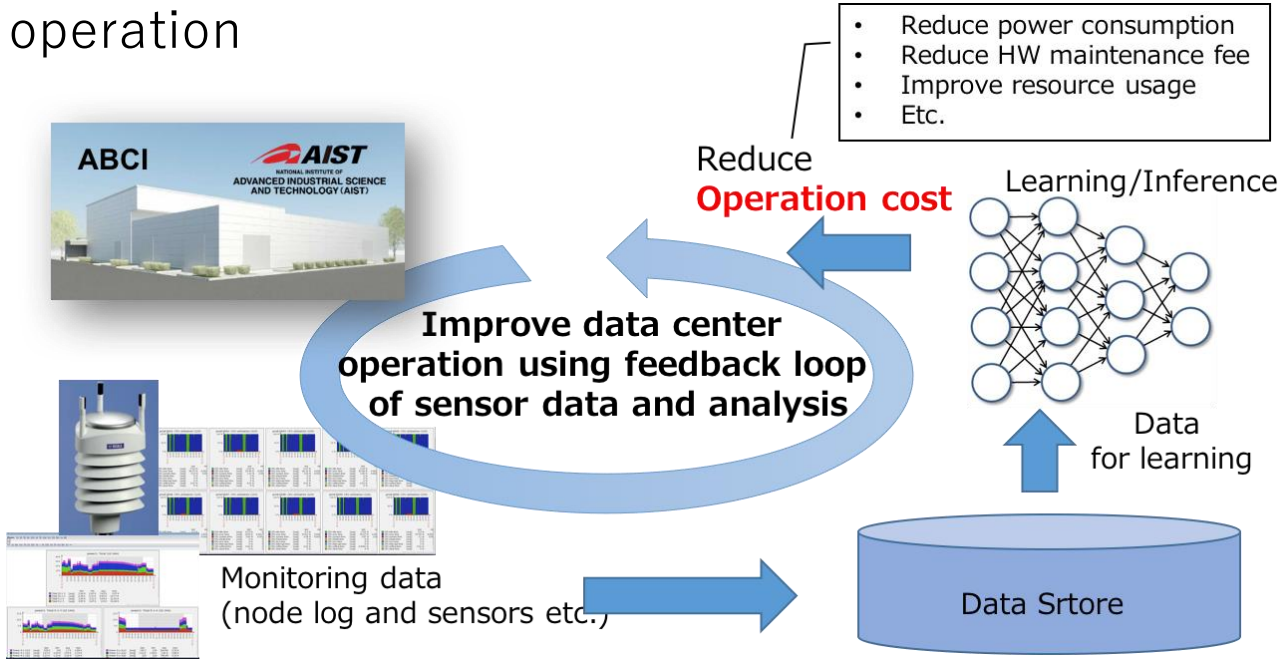
Warm Water Cooling Distribution in T3
PUE ~= 1.03 (~1.1 w/storage)



Smart Data Center Operation for ABCI (NEDO Project 2017-)

- Started to develop a system that achieves a self-sustainable operation and reduce operation cost of data center, especially for ABCI system
  - Data storage for storing sensor data from node, cooling system, etc.
  - ML/DL algorithms to analyze the data and model data center behavior
    - Reduce power consumption, detect errors, etc.
  - Apply the algorithms to improve operation

- Current status
  - Started from Aug. 2017
  - Designing/developing sensor data collector and its storage



Comparing TSUBAME3/ABCI to Classical IDC
AI IDC CAPX/OPEX accelerartion by > x100

Perf > 400~600

Power Eff > 200~300

Traditional Xeon IDC

~10KW/rack PUE 1.5~2

15~20 1U Xeon Servers

2 Tera AI-FLOPS(SFP) / server

30~40 Tera AI-FLOP / rack

TSUBAME3 (+Volta) & ABCI IDC

~60KW/rack PUE 1.0x

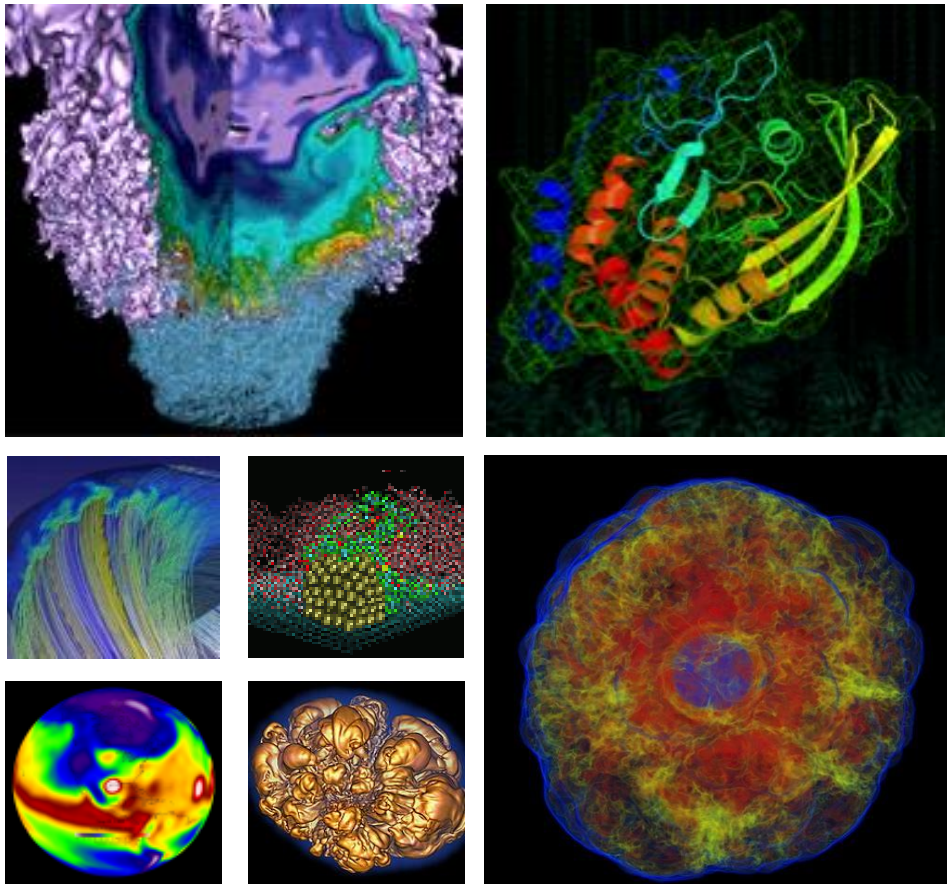
~36 T3 evolution servers

~500 Tera AI-FLOPS(HFP) / server

~17 Peta AI-FLOPs / rack

Energy Efficiency Gains from Software:

Retrospectives and Perspectives



John Shalf
Lawrence Berkeley National Laboratory

SC17
November, 2017

Measurement and Metrics

"If you can not measure it, you can not improve it."

-Lord Kelvin (1883)

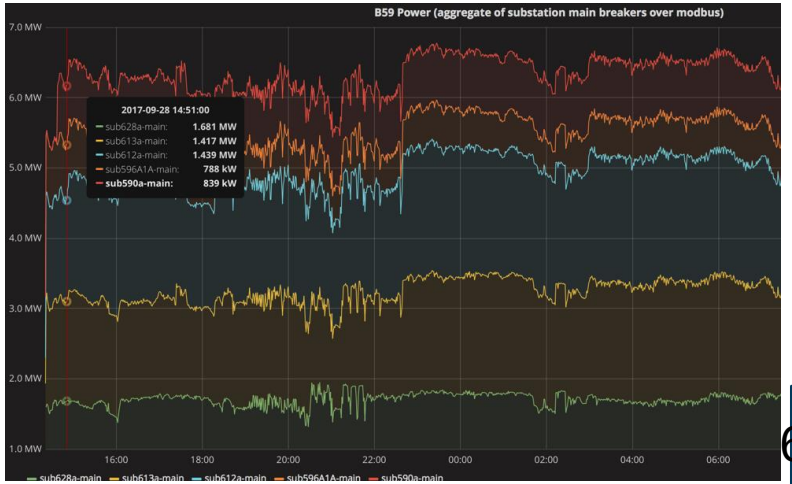
High Data Rate Collection System

Current Data Sources
Substations, panels, PDUs, UPS
Cori & Edison SEDC
Onewire Temp & RH sensors
BMS through BACNET
Indoor Particle counters
Weather station

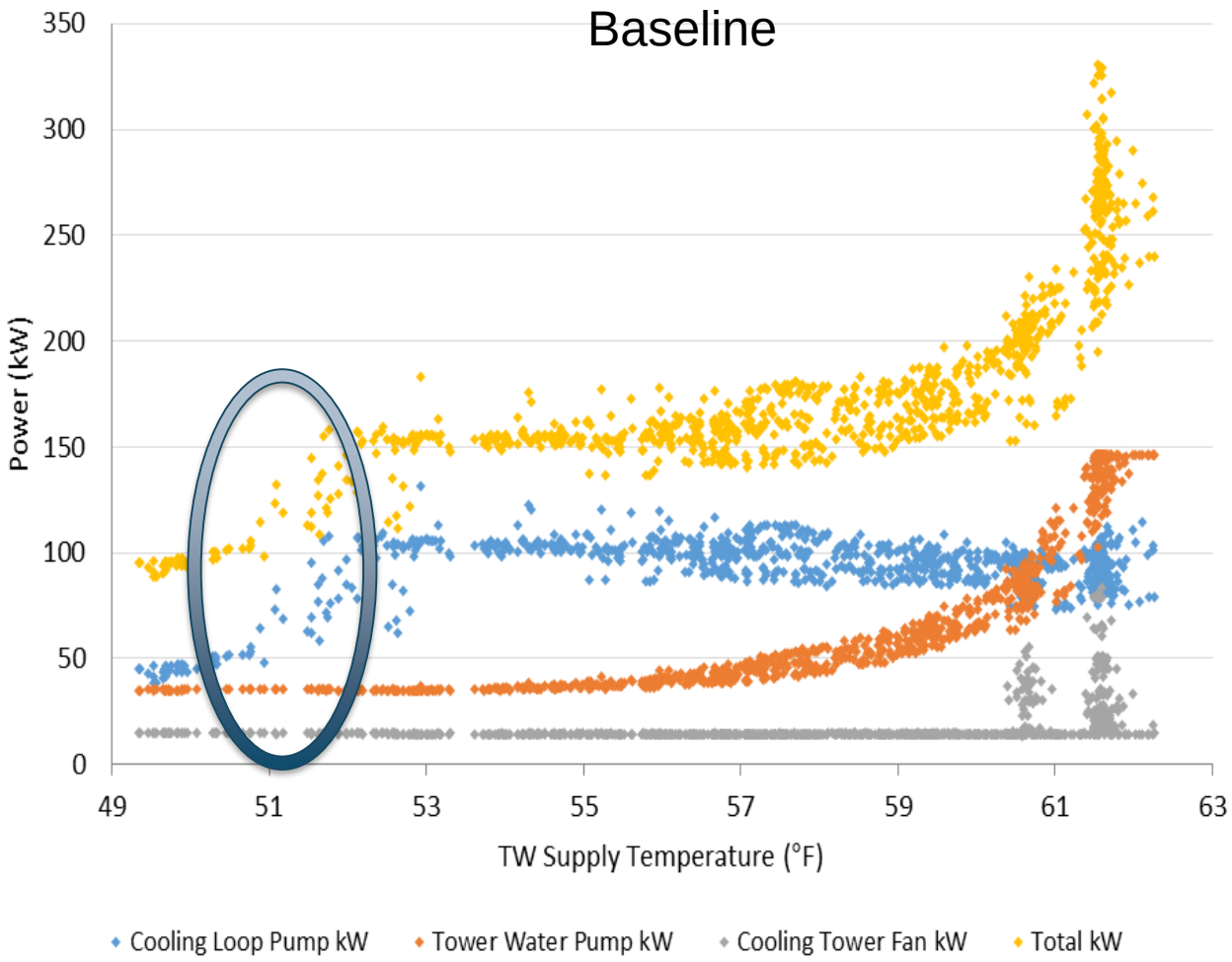
Future Data Sources
Syslog
Job Data
Lustre + GFPS statistics
LDMS
Outdoor particle counters
??

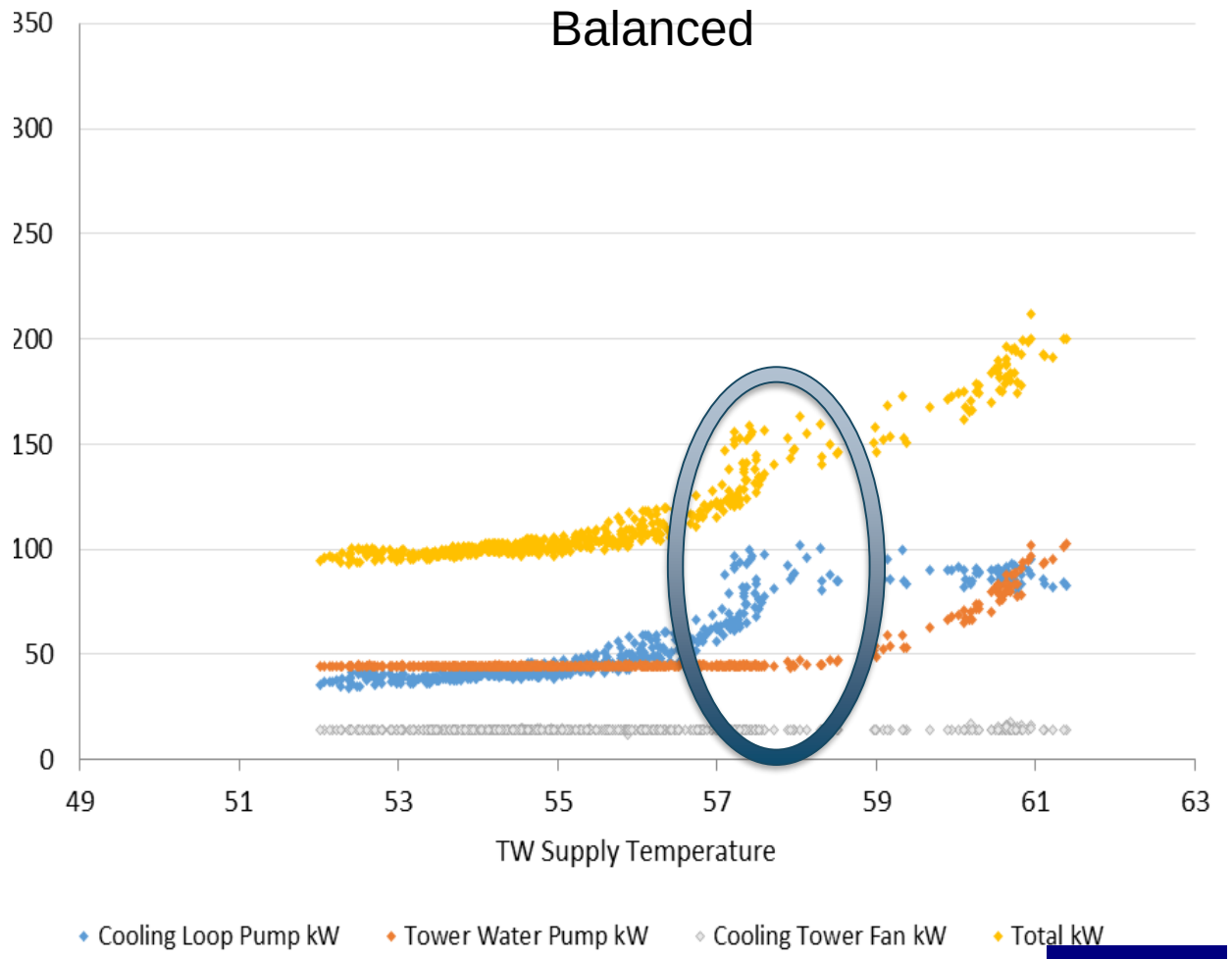
- **Rabbit MQ, Elastic, Linux**
  - Collects ~20K data items per second
  - Over 40TB data online (100TB capacity)
  - 45 days of SEDC (versus 3 hours on SMW)
  - 180 days of BMS data (6X more than BMS)
  - 260 days of power data

Kibana, Grafana



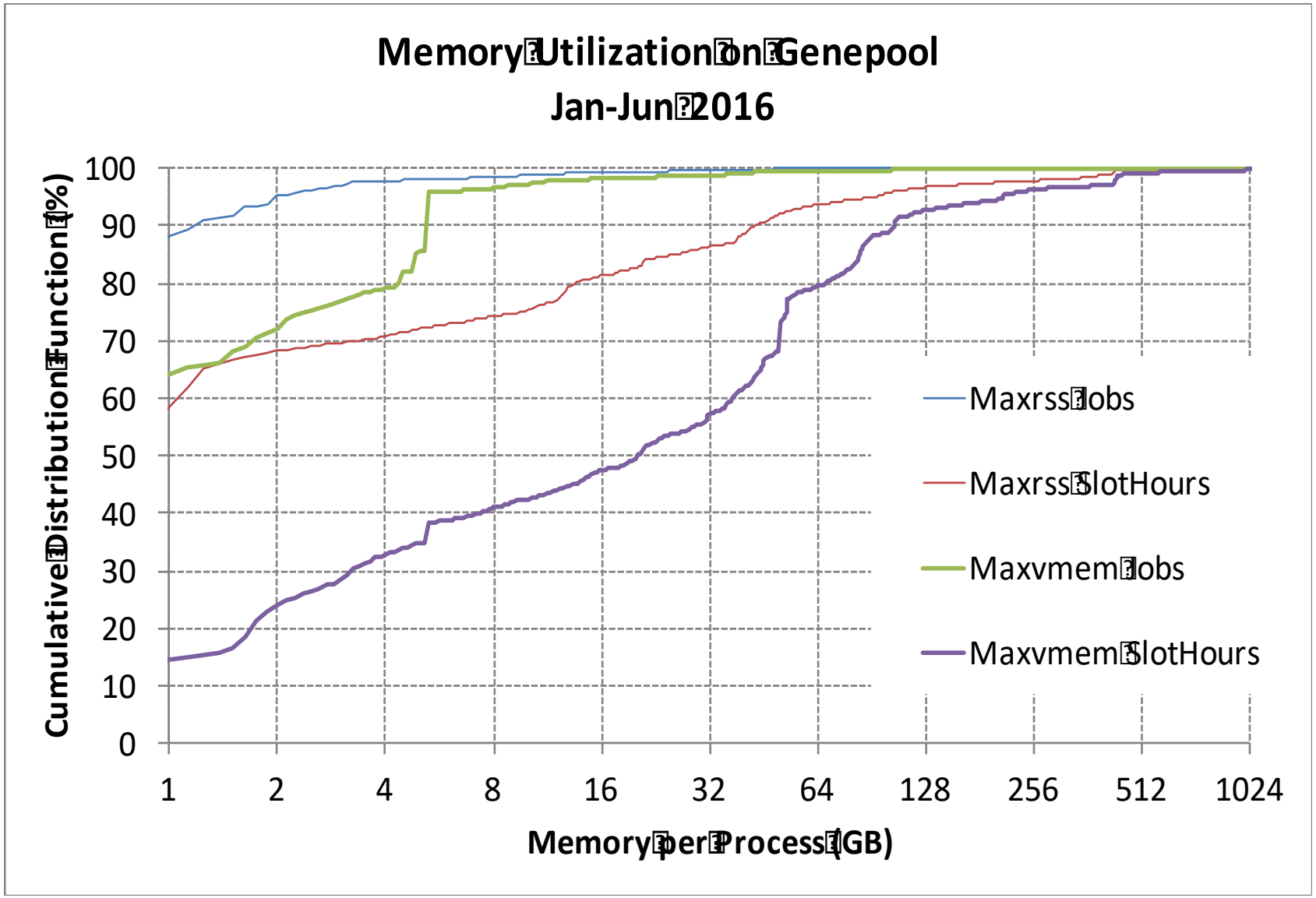
Liquid Cooling Performance





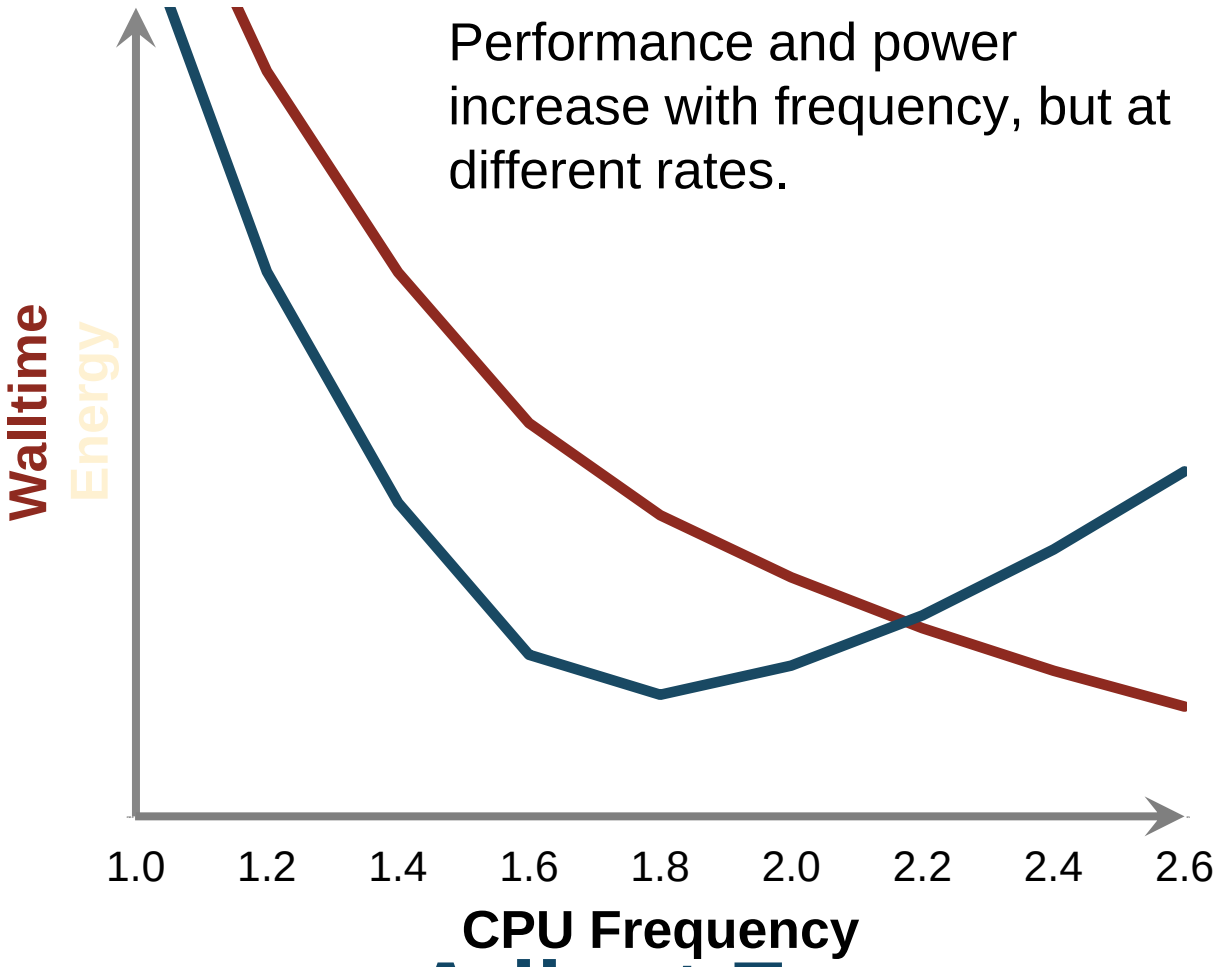
Have saved 700,000 kWh/year

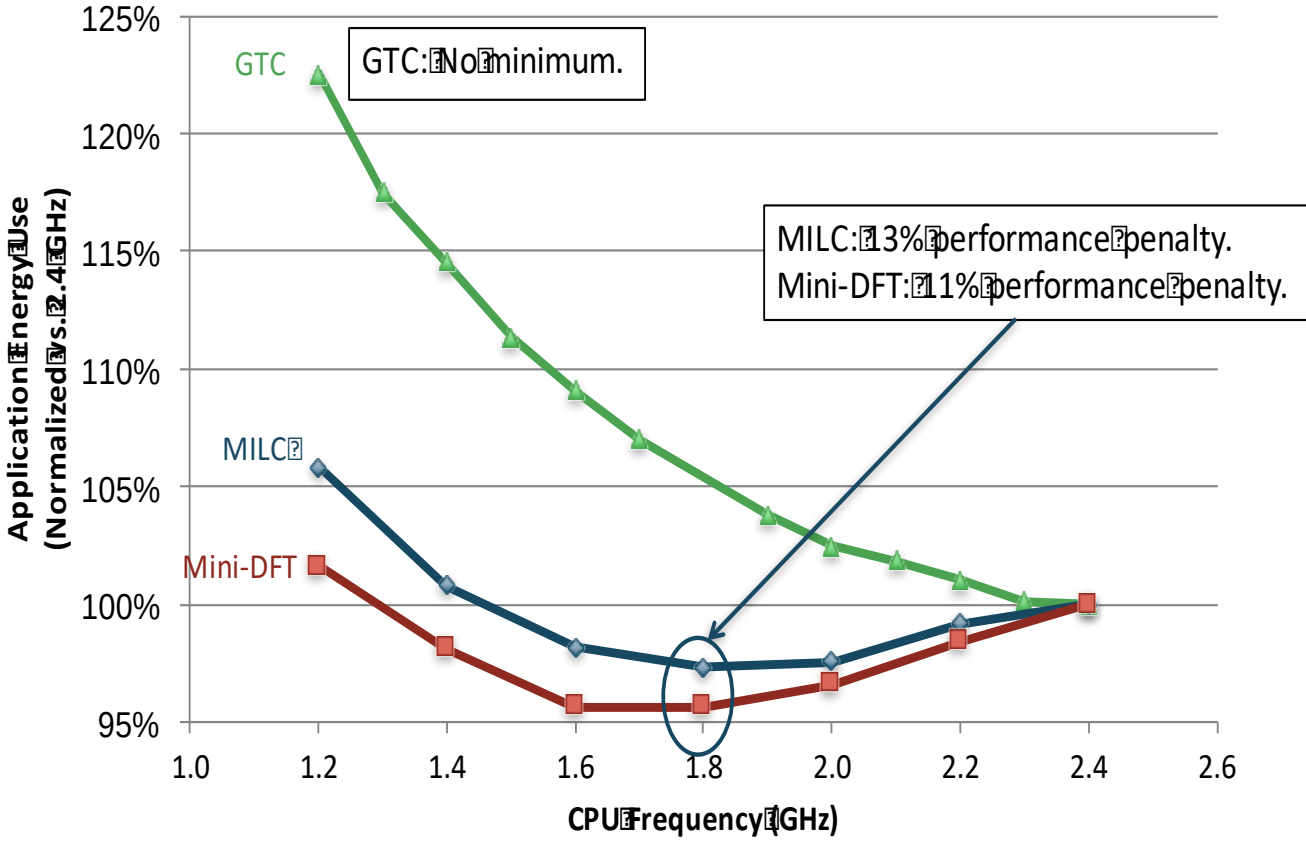
Analytics for resource utilization



Introspective business analytics can save you a LOT of MONEY and POWER

Realtime Feedback and Tuning for Energy Efficiency





Adjust Frequency & Measure Energy

Anatomy of a “Value” Metric

$$\frac{\text{Good Stuff}}{\text{Bad Stuff}}$$

Anatomy of a "Value" Metric

$$\frac{\text{FLOP/s}}{\text{Watts}}$$

Bogus!!!

Potentially Bogus!!

Anatomy of a "Value" Metric

Optimizing performance has substantial room for improvement

$$\frac{\text{Measured Performance}}{\text{Measured Watt}}$$

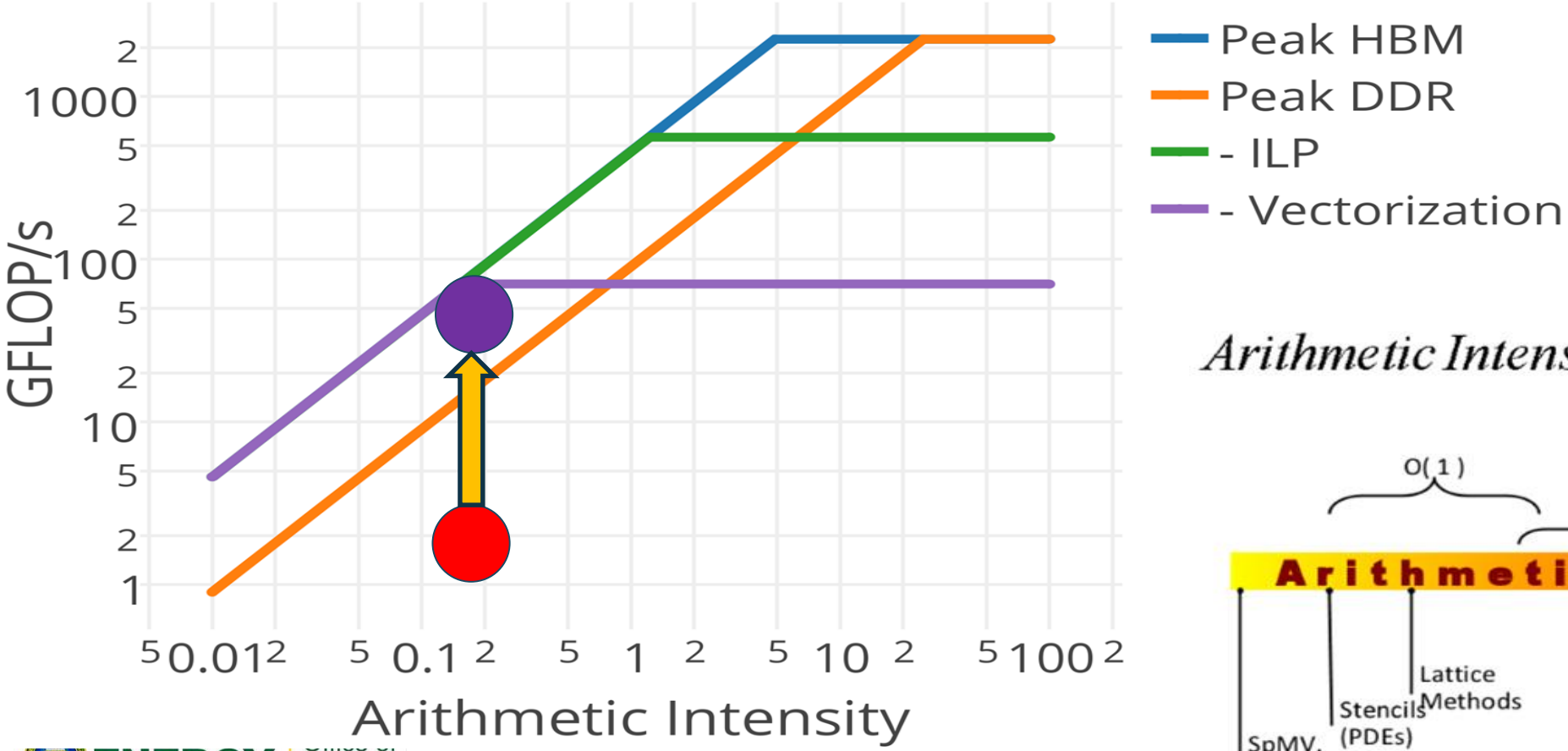
Have mined this to a limit case

At PUE of 1.1, hard to squeeze out more

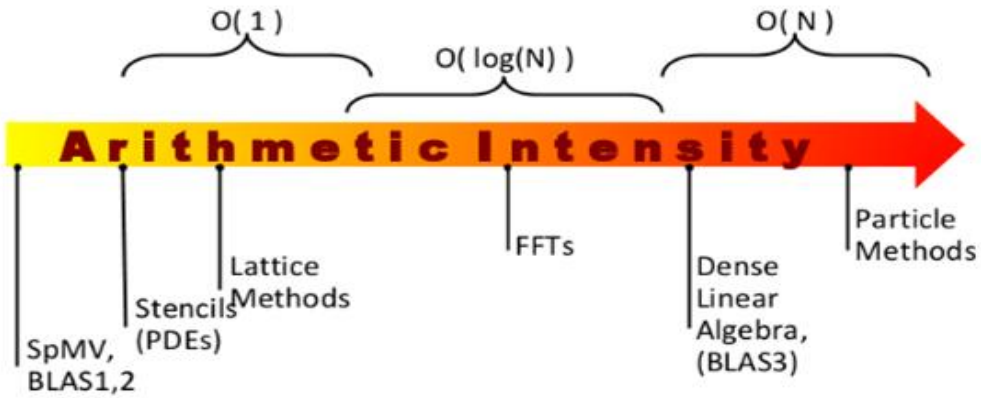
But must remain vigilant

Roofline Model: *"Houston, is there a problem?"*

Am I under-performing? *(FLOP rate is a poor metric!)*
And how bad is it???



$$\text{Arithmetic Intensity} = \frac{\text{Total FLOPs}}{\text{Total Bytes}}$$

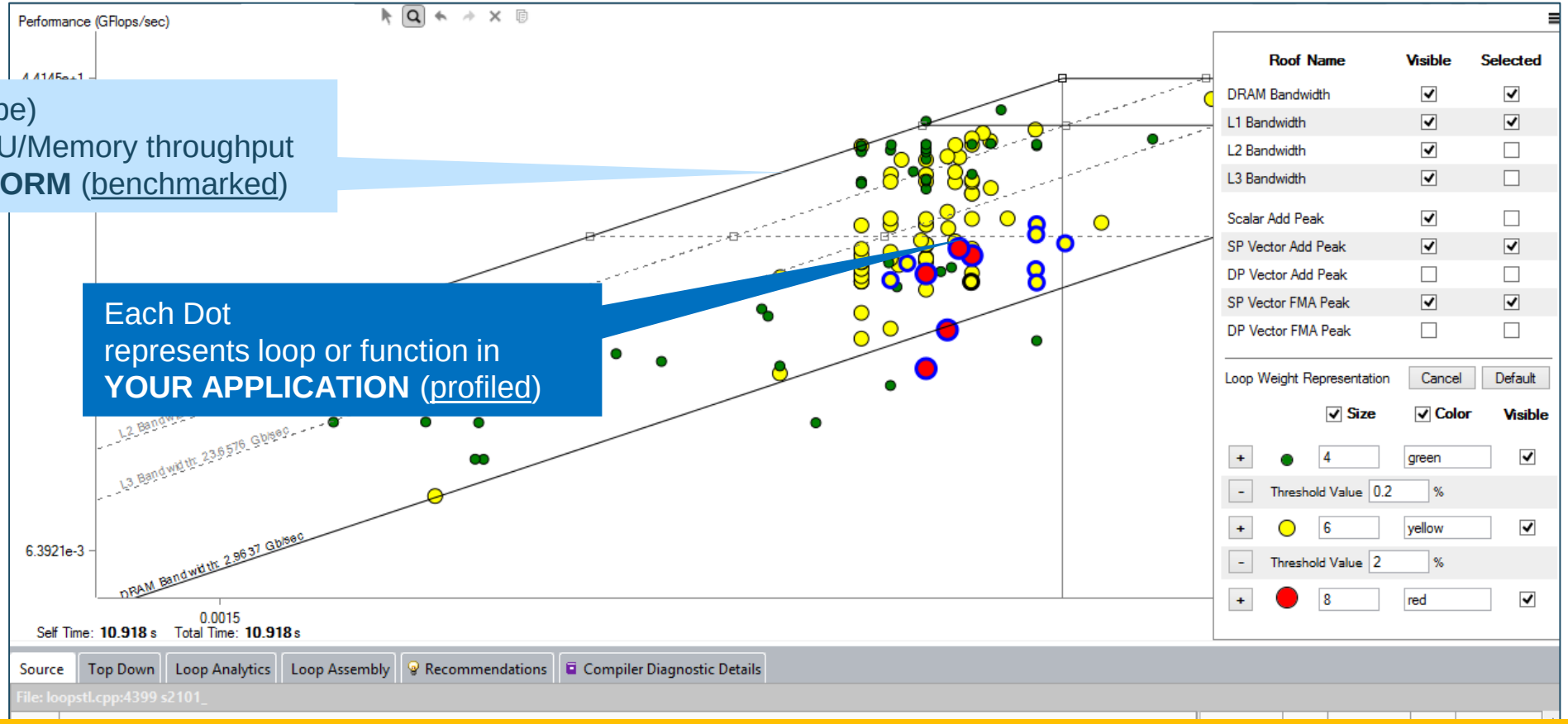


Roofline Automation in Intel® Advisor 2017

Improving under collaboration with NERSC, LBNL, DOE.

Each Roof (slope)
Gives peak CPU/Memory throughput of your **PLATFORM** (benchmarked)

Each Dot
represents loop or function in
YOUR APPLICATION (profiled)

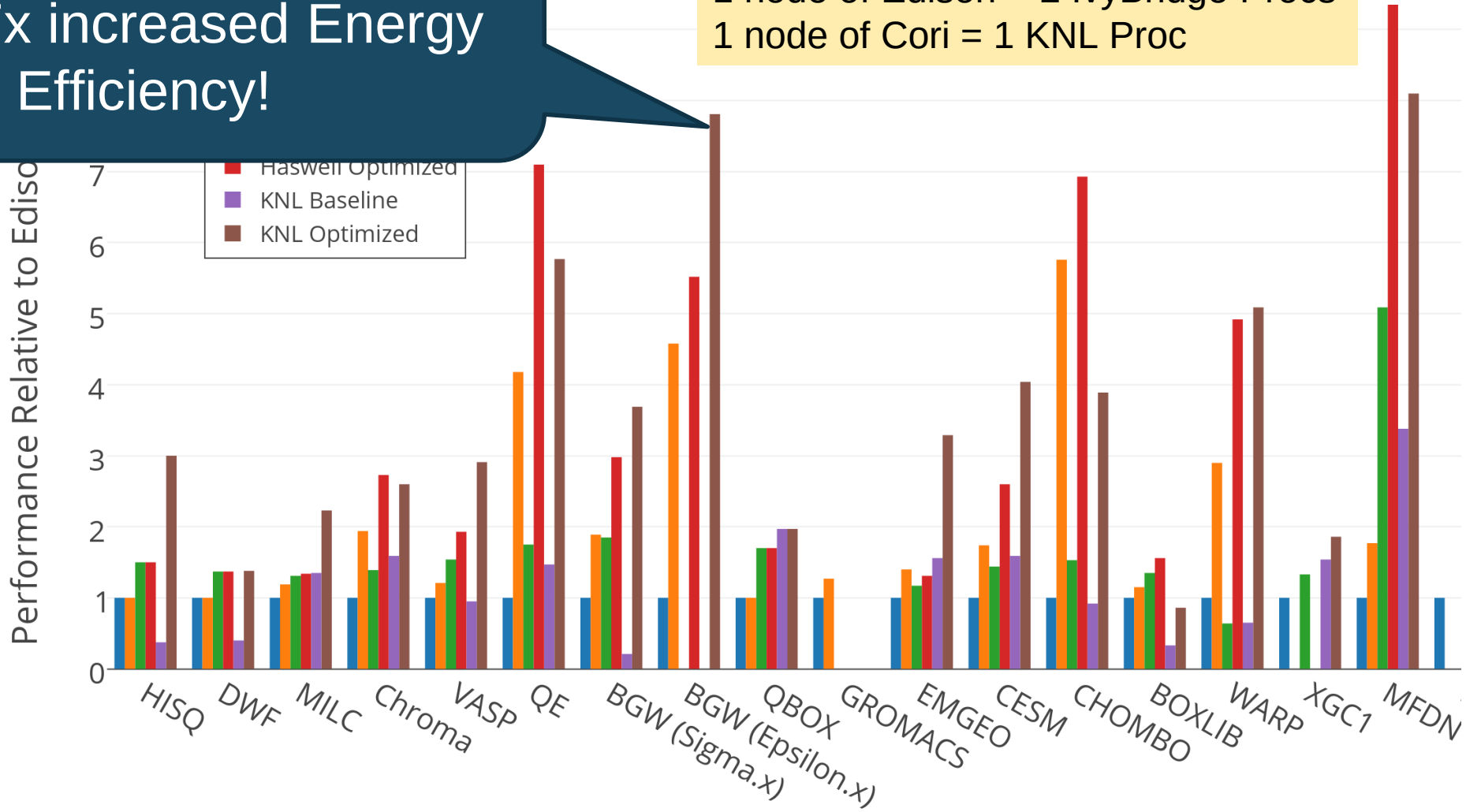


Automatic and integrated – first class citizen in Intel® Advisor

Speed-ups from NESAP Investments

7x Increase in Performance is nearly 7x increased Energy Efficiency!

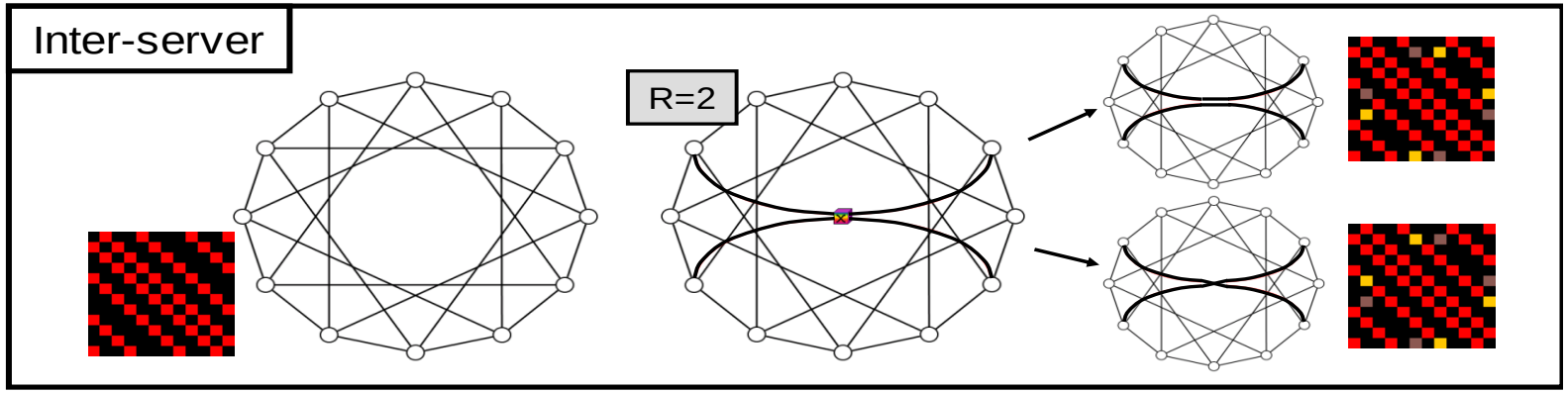
Comparison is node to node
1 node of Edison = 2 IvyBridge Procs
1 node of Cori = 1 KNL Proc

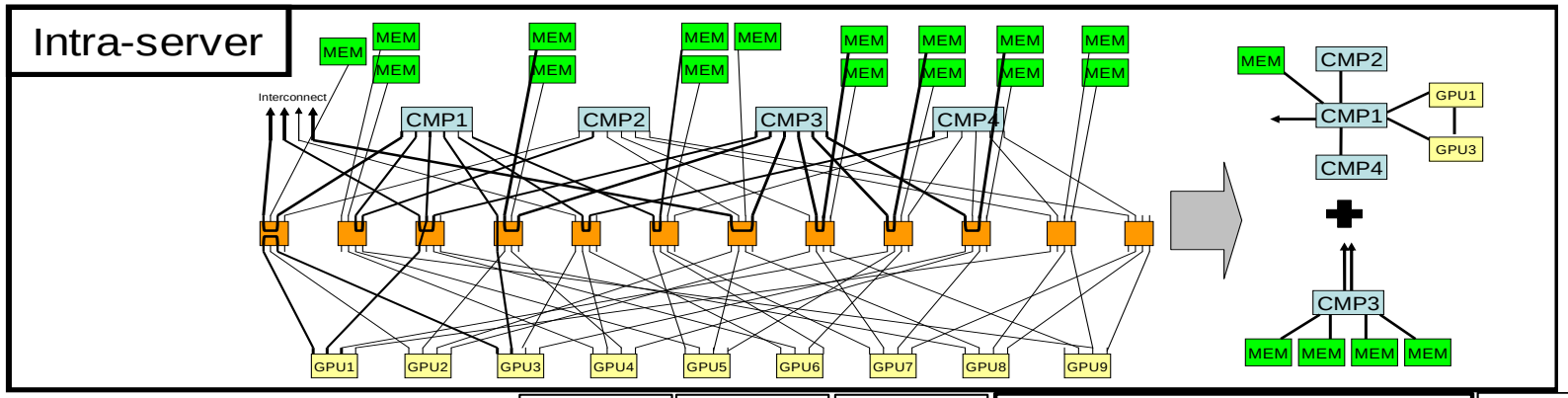


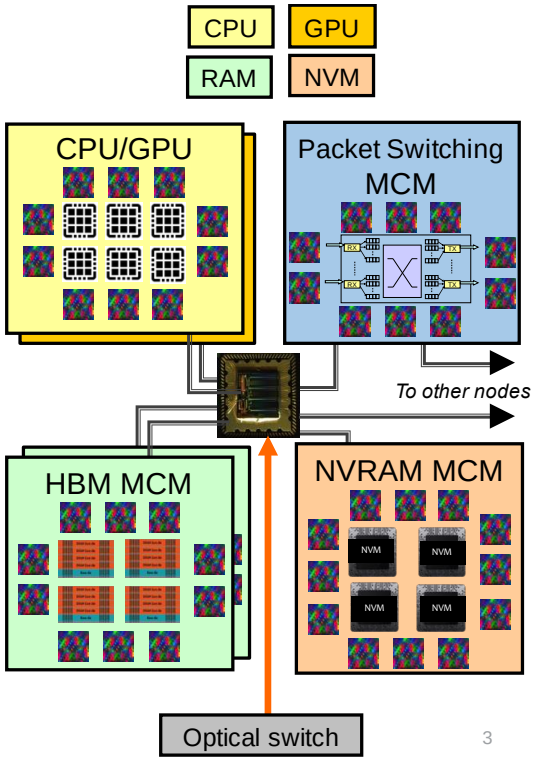
U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

ARPAe PINE *Key Concept: dynamic Bandwidth Steering for energy efficiency (use case for power API?)*







Johansson Coolbaugh Bowers Gaeta Bergman Lipson Patel Dennison Shalf Ghobadi

Conclusions

- **Measuring datacenter operational efficiency has resulted in huge improvements**
  - At PUE 1.1, it is hard to mine MORE efficiency
  - But must stay vigilant with continuous monitoring
- **Runtime System Optimizations driven by real-time feedback may work**
  - Currently many programming structures are fixed at compile time (currently limited)
  - Software mediated control loop may be too slow for nanosecond events
  - Runtime system will always know less about application than application scientist
- **Tuning codes is HUGE energy benefit! (energy efficiency = performance/watts)**
  - NESAP code tuning offered largest (3x – 7x) efficiency improvements!
  - Need tools that do a better job identifying the performance ***opportunity*** (*roofline*)
- **Integrated analytics at every level is essential for identifying efficiency opportunities**
  - But we are collecting data faster than we can formulate questions to pose against it!
  - Perhaps an opportunity for machine learning as “anomaly detection”

Some panel framing questions

$$\text{PUE} = \frac{\textbf{Total Facility Energy}}{\textbf{IT Equipment Energy}}$$

What system software has been most helpful in improving system energy efficiency?

Which software stack layers should support energy awareness?

- System software, libraries, MPI runtime/job schedulers, applications, …

Is there user willingness to exploit libraries for energy efficiency?

What might encourage adoption of APIs for measurement/management?

- Sandia Power API, Intel GEOPM, Redfish, …

Where are the biggest opportunities for energy efficiency improvements?

Can software-based runtime introspection mechanisms react fast enough?

- Or is dedicated hardware required for the control loop?

What is the user reward for developing energy efficient codes?

- Versus investing time in making the software more scientifically capable

What is the total cost of ownership (TCO) tradeoff in tuning codes for *(a) improved performance* or *(b) reduced energy consumption*

What are the fundamental properties of energy efficient applications?