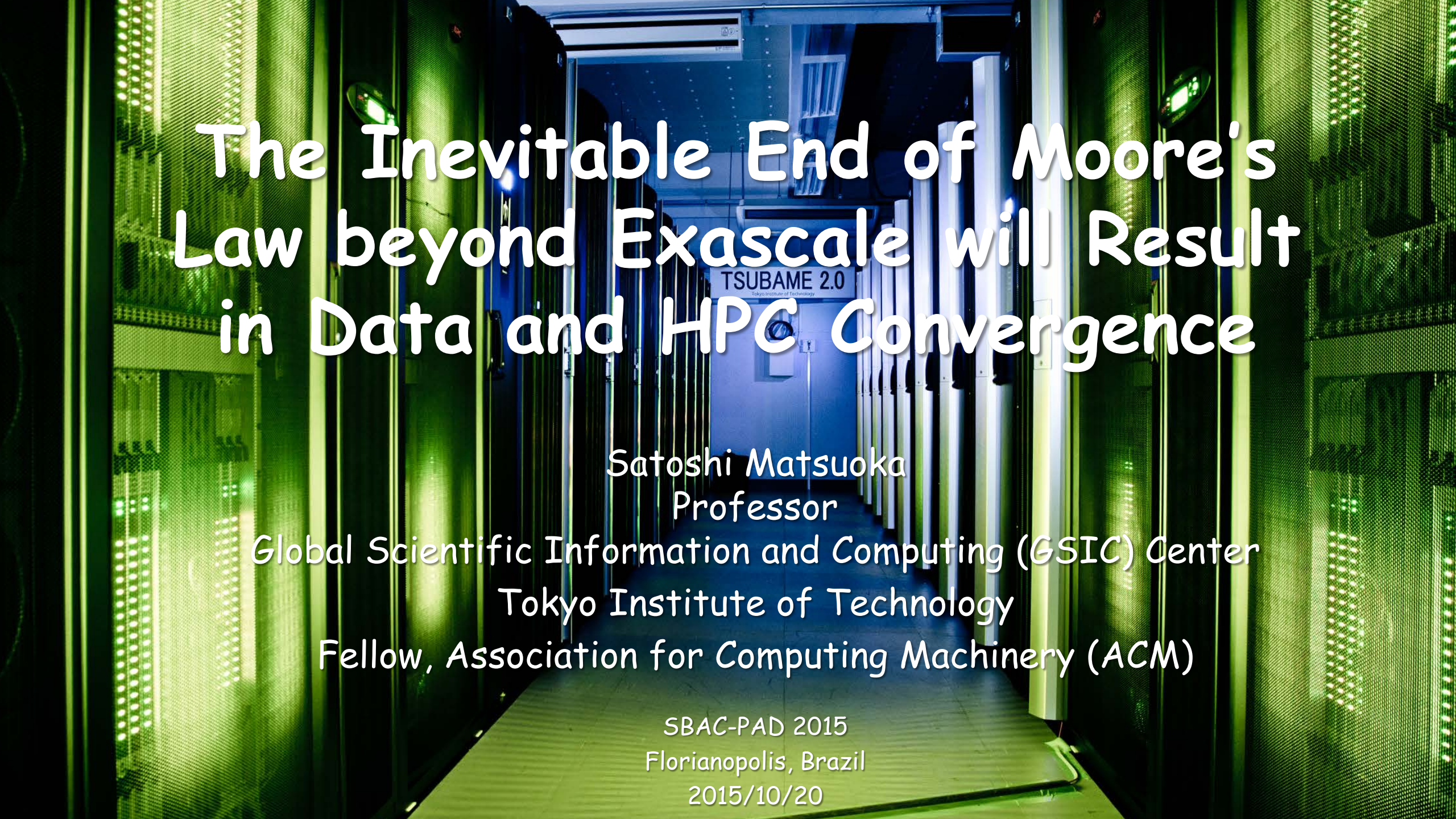


The Inevitable End of Moore's Law beyond Exascale will Result in Data and HPC Convergence

Satoshi Matsuoka
Professor
Global Scientific Information and Computing (GSIC) Center
Tokyo Institute of Technology
Fellow, Association for Computing Machinery (ACM)

SBAC-PAD 2015
Florianopolis, Brazil
2015/10/20

TSUBAME2.0 Nov. 1, 2010
"The Greenest Production Supercomputer in the World"

- GPU-centric (> 4000) high performance & low power
- Small footprint (~200m2 or 2000 sq.ft), low TCO
- High bandwidth memory, optical network, SSD storage…

TSUBAME 2.0
New Development

System
(42 Racks)
1408 GPU Compute Nodes,
34 Nehalem "Fat Memory" Nodes

Rack
(8 Node Chassis)

Node Chassis
(4 Compute Nodes)

Compute Node
(2 CPUs,3 GPUs)

Chip
(CPU ,GPU)

CPU(Westmere EP)
76.8 GFLOPS
32nm

GPUs(Tesla M2050)
515 GFLOPS
3 GB 40nm

1.6 TFLOPS
55 GB/103 GB
>400GB/s Mem BW
80Gbps NW BW
~1KW max

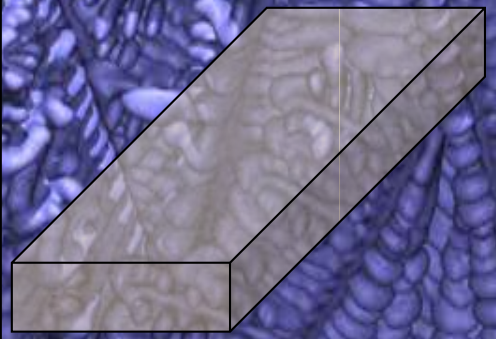
6.7 TFLOPS
220 GB/412 GB
>1.6TB/s Mem BW

53.6 TFLOPS
1.7 TB/3.2 TB
>12TB/s Mem BW
35KW Max

2.4 PFLOPS
80 TB
4224 GPUs
>600TB/s Mem BW
220Tbps NW
Bisecion BW
1.4MW Max

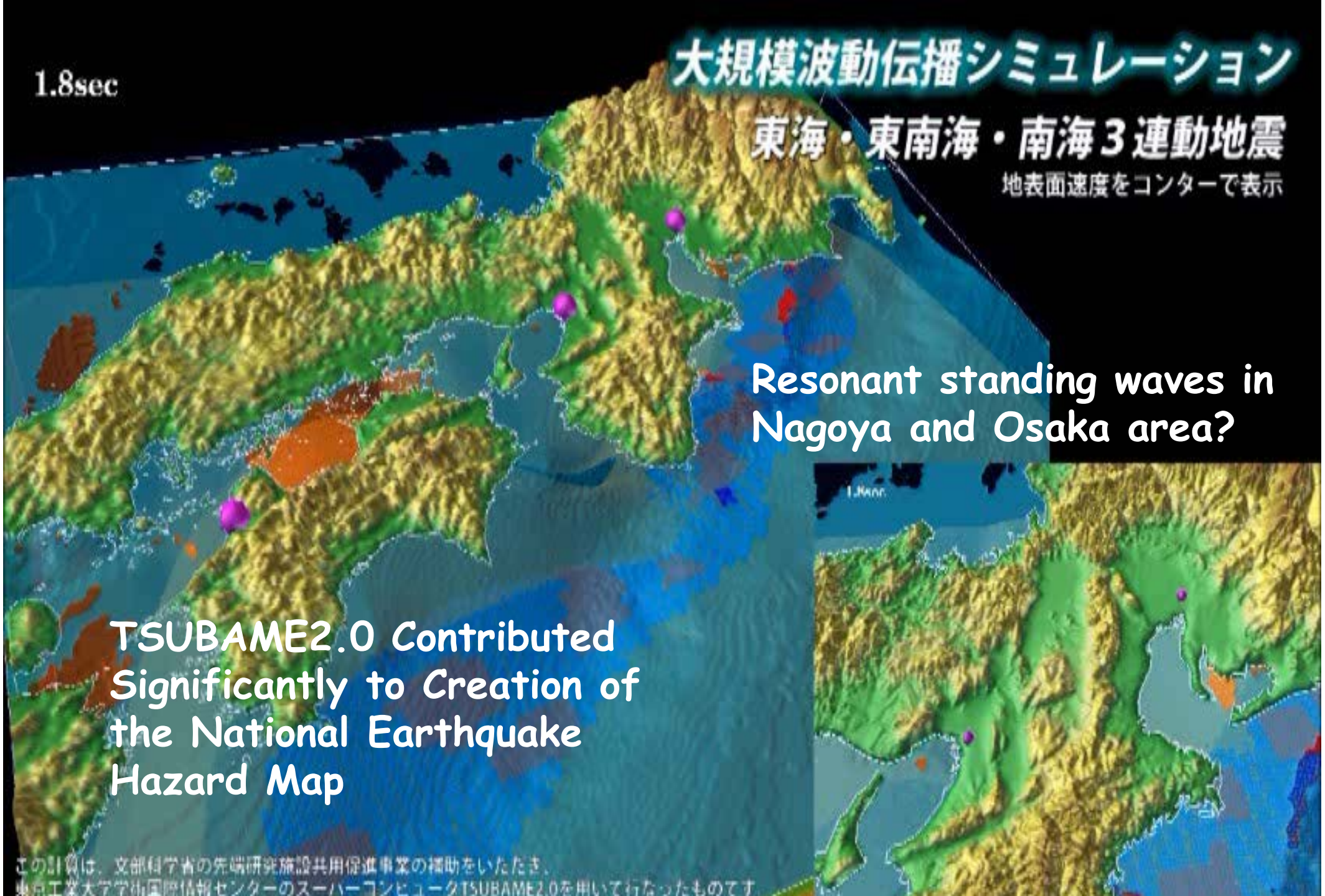
Integrated by NEC Corporation

ACM Gordon Bell Prize 2011
2.0 Petaflops Dendrite Simulation



Lightweight and durable metal alloy for future automobiles

Special Achievements in Scalability and Time-to-Solution
"Peta-Scale Phase-Field Simulation for Dendritic Solidification on the TSUBAME 2.0 Supercomputer"
2 Petaflops (3.4 Petaflops on TSUBAME 2.5)


1.8sec
大規模波動伝播シミュレーション
東海・東南海・南海３連動地震
地表面速度をコンターで表示
Resonant standing waves in Nagoya and Osaka area?
TSUBAME2.0 Contributed Significantly to Creation of the National Earthquake Hazard Map

Comparing K Computer to TSUBAME 2.5

東京工業大学
Tokyo Institute of Technology

GSIC
Global Scientific Information and Computing Center

Perf ≒
Cost ≪

独立行政法人理化学研究所
計算科学研究機構
RIKEN Advanced Institute for Computational Science

K Computer (2011)

11.4 Petaflops SFP/DFP
$1400mil 6 years
(incl. power)
x30 TSUBAME2

TSUBAME2.0(2010)
→ TSUBAME2.5(2013)
17.1 Petaflops SFP
5.76 Petaflops DFP
$45mil / 6 years (incl. power)

2016 H2 TSUBAME3.0 Leading Machine Towards Exa & Big Data

1. **"Everybody's Supercomputer" - High Performance (~20 Petaflops, 4~5PB/s Mem, ~Pbit/s NW), innovative high cost/performance packaging & design, in mere 180m²…**
2. **"Extreme Green" – ~10GFlops/W power-efficient architecture, system-wide power control, advanced cooling, future energy reservoir load leveling & energy recovery**
3. **"Big Data Convergence" – Extreme high BW &capacity, deep memory hierarchy, extreme I/O acceleration, Big Data SW Stack for machine learning, graph processing, …**
4. **"Cloud SC" – dynamic deployment, container-based node co-location & dynamic configuration, resource elasticity, assimilation of public clouds…**
5. **"Transparency" - full monitoring & user visibility of machine & job state, accountability via reproducibility**

2006 TSUBAME1.0
80 Teraflops, #1 Asia #7 World
"Everybody's Supercomputer"

2010 TSUBAME2.0
2.4 Petaflops #4 World
"Greenest Production SC"

2011 ACM Gordon Bell Prize

2013
TSUBAME2.5
upgrade
5.7PF DFP
/17.1PF SFP
20% power
reduction

2013 TSUBAME-KFC
#1 Green 500

2016 TSUBAME3.0
~20PF(DFP) 4~5PB/s Mem BW
10GFlops/W power efficiency
Big Data & Cloud Convergence

Large Scale Simulation
Big Data Analytics
Industrial Apps

6

2020 Global Target

1 Exaflop,
200TeraByte/s@
20MW

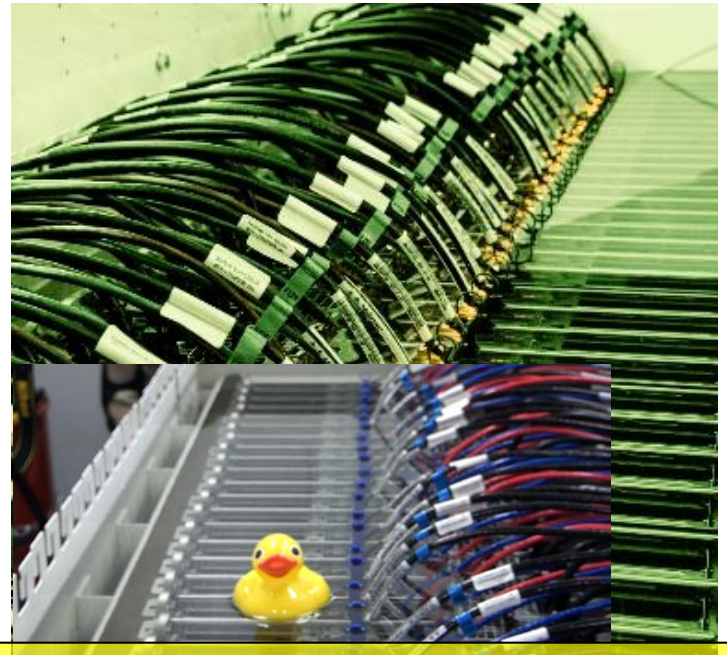
= 50 GigaFlops/W,
10GB/s / W

x1000~x2500
improvement!

Machineg	Power (incl. cooling)	Linpack Perf (PF)	Linpack MFLOPs/W	Factor	Total Mem BW TB/s (STREAM)	Mem BW MByte/S / W	Factor
Earth Simulator 1	10MW	0.036	3.6	13,400	160	16	312
Tsubame1.0 (2006Q2)	1.8MW	0.038	21	2,368	13	7.2	692
Tsubame2.0 (2010Q4)	**1.8MW**	**1.2**	**667**	**75**	**440**	**244**	**20**
			x32			***x34***	
K Computer (2011Q2)	~16MW	10	625	80	3300	206	24
BlueGene/Q (2012Q1)	~12MW?	17	~1400	~35	3000	250	20
TSUBAME2.5 (2013Q3)	**1.4MW**	**~2.8**	**~2000**	**~25**	**802**	**572**	**8.7**
TSUBAME3 (2016Q2)	**1.8MW**	**~18**	**~10,000**	**~5**	**~50K**	**~2800**	**~4**
			~x75			***~x40***	
EXA (2020)	20MW	1000	50,000	1	200K	10,000	1

TSUBAME-KFC: TSUBAME3 Prototype [ICPADS2014]

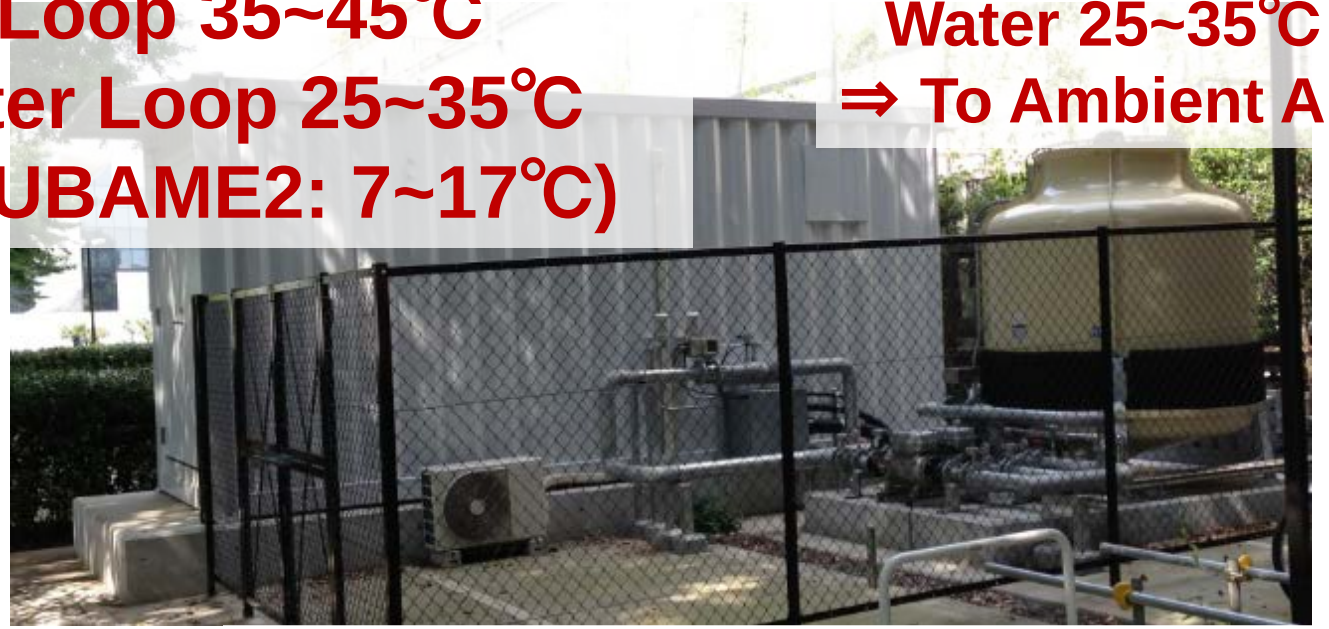
Oil Immersive Cooling＋ Hot Water Cooling + High Density Packaging + Fine-Grained Power Monitoring and Control¥



High Density Oil Immersion
210TFlops (DFP)
630TFlops (SFP)
(=>1.5 PetaFlops upgrade)

High Temperature Cooling
Oil Loop 35~45℃
⇒ Water Loop 25~35℃
(c.f. TSUBAME2: 7~17℃)

Cooling Tower: Water 25~35℃
⇒ To Ambient Air



Experimental Container Facility
20 feet container (16m^2)
Fully Unmanned Operation

TSUBAME-KFC #1 Green 500 List (Nov. 2013)

- 1st achievement as Japanese supercomputer
- #1 again in June 2014
- 10GFLOPS/W incl. Cooling for TSUBAME3, + Energy Recovery

The Green500 List

Listed below are the November 2013 The Green500's energy-efficient supercomputers ranked from 1 to 10.

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	4,503.17	GSIC Center, Tokyo Institute of Technology	TSUBAME-KFC - LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.100GHz, Infiniband FDR, NVIDIA K20x	27.78
2	3,631.86	Cambridge University	Wilkes - Dell T620 Cluster, Intel Xeon E5-2630v2 6C 2.600GHz, Infiniband FDR, NVIDIA K20	52.62
3	3,517.84	Center for Computational Sciences, University of Tsukuba	HA-PACS TCA - Cray 3623G4-SM Cluster, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband QDR, NVIDIA K20x	78.77
4	3,185.91	Swiss National Supercomputing Centre (CSCS)	Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Level 3 measurement data available	1,753.66
5	3,130.95	ROMEO HPC Center - Champagne-Ardenne	romeo - Bull R421-E3 Cluster, Intel Xeon E5-2650v2 8C 2.600GHz, Infiniband FDR, NVIDIA K20x	81.41
6	3,068.71	GSIC Center, Tokyo Institute of Technology	TSUBAME 2.5 - Cluster Platform SL390s G7, Xeon X5670 6C 2.930GHz, Infiniband QDR, NVIDIA K20x	922.54
7	2,702.16	University of Arizona	iDataPlex DX360M4, Intel Xeon E5-2650v2 8C 2.600GHz, Infiniband FDR14, NVIDIA K20x	53.62
8	2,629.10	Max-Planck-Gesellschaft MPI/IPP	iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x	269.94
9	2,629.10	Financial Institution	iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x	55.62
10	2,358.69	CSIRO	CSIRO GPU Cluster - Nitro G16 3GPU, Xeon E5-2650 8C 2.000GHz, Infiniband FDR, Nvidia K20m	71.01

TSUBAME-KFC

TSUBAME 2.5

* Performance data obtained from publicly available sources including TOP500

Ranks 1 - 100

Ranks 101 - 200

Ranks 201 - 300

Ranks 301 - 400

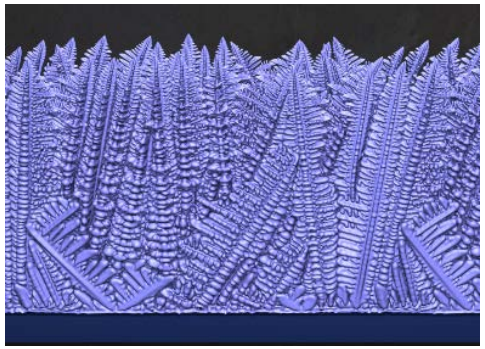
Ranks 401 - 500

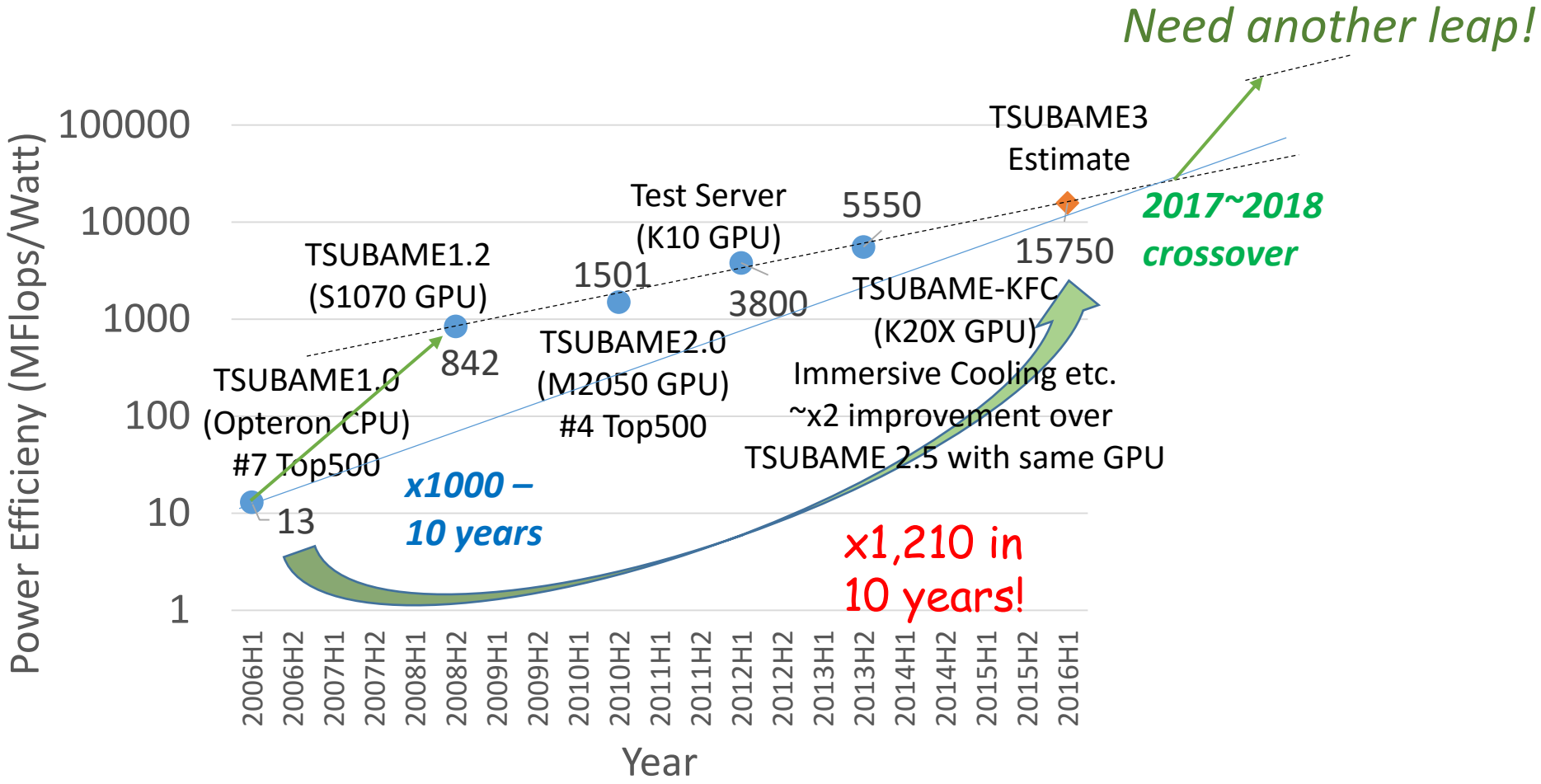
Downloads

Microsoft Excel™ Spreadsheet (.xls)

Comma Separated Values (.csv)

Power Efficiency of GB Dendrite Simulation since 2006
Good news and bad news towards Post-Moore

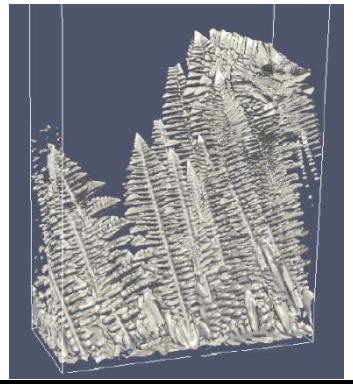




Measured for the 2011 Gordon Bell Award Dendritic Solidification App
Flop/s/W = Total #Flops / J = energy to solution given same problem

Data is the problem now and for Exascale

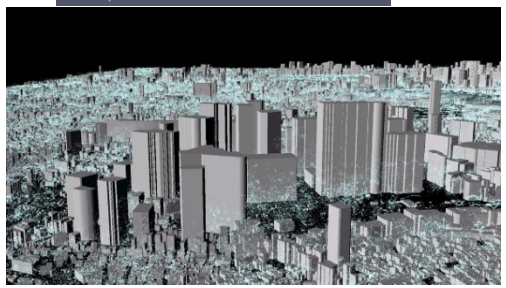
TSUBAME2 Application Examples



Denderite Simuation[Gordon Bell 2011]

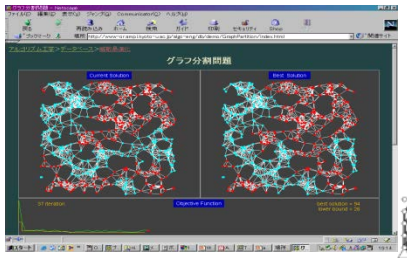
- Stencil-Based
- 4000GPU 2PFlops(SP)
- 3GB × 4000 max size

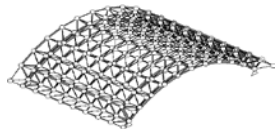
x10 GPU Requirement

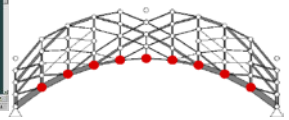


Whole City Simulation

- Stencil-Based (LBM)
- 1000GPU 150TFlops
- 3GB × 1000 max size







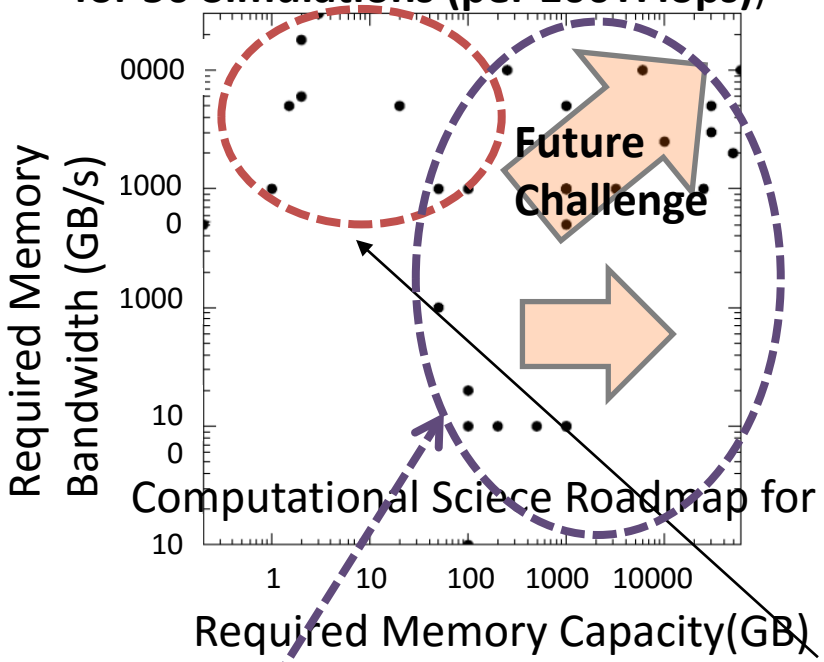
SDPARA Semi-Definite Optim. Solver

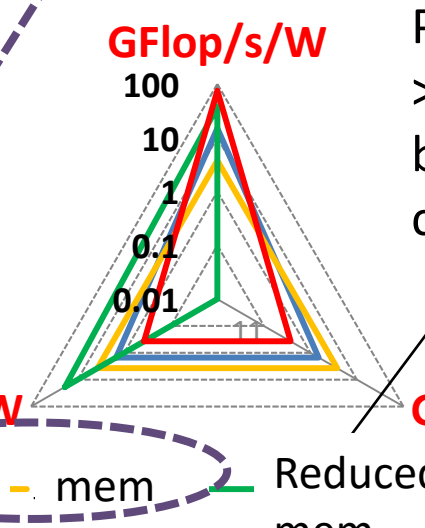
- Matrix (Dense+Sparse)＋EP
- 4000GPU 1PFlops/s
- Uses 70TB Host Memory

- PCI-e Bottleneck
- Lack of Host Memory Capacity

Majority of high-memory applications and next-gen big data apps are not possible

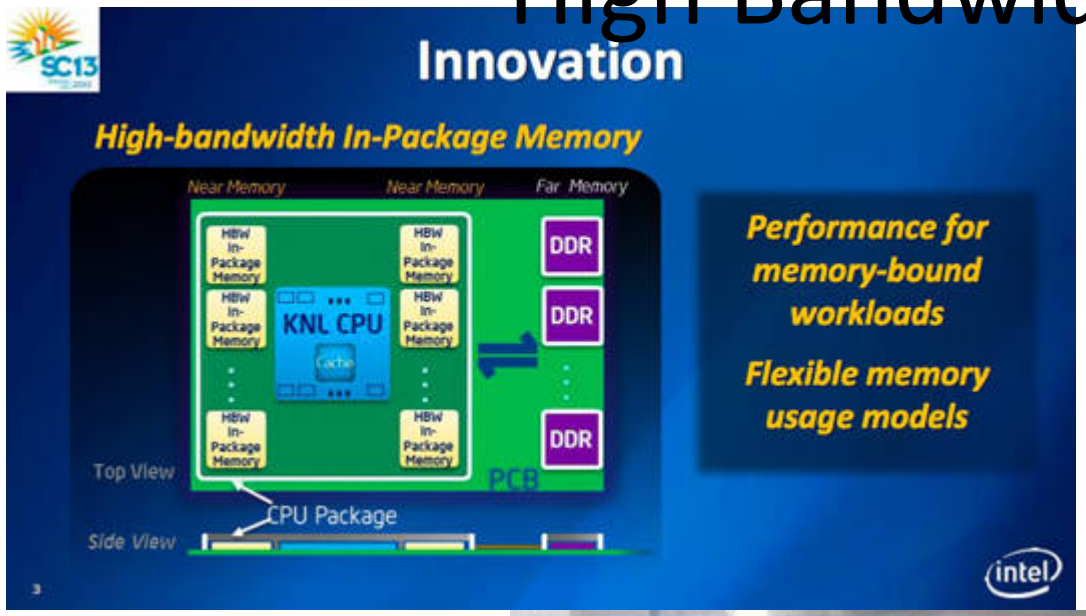
Memory Bandwidth-Capacity Requirement for 50 Simulations (per 100TFlops))



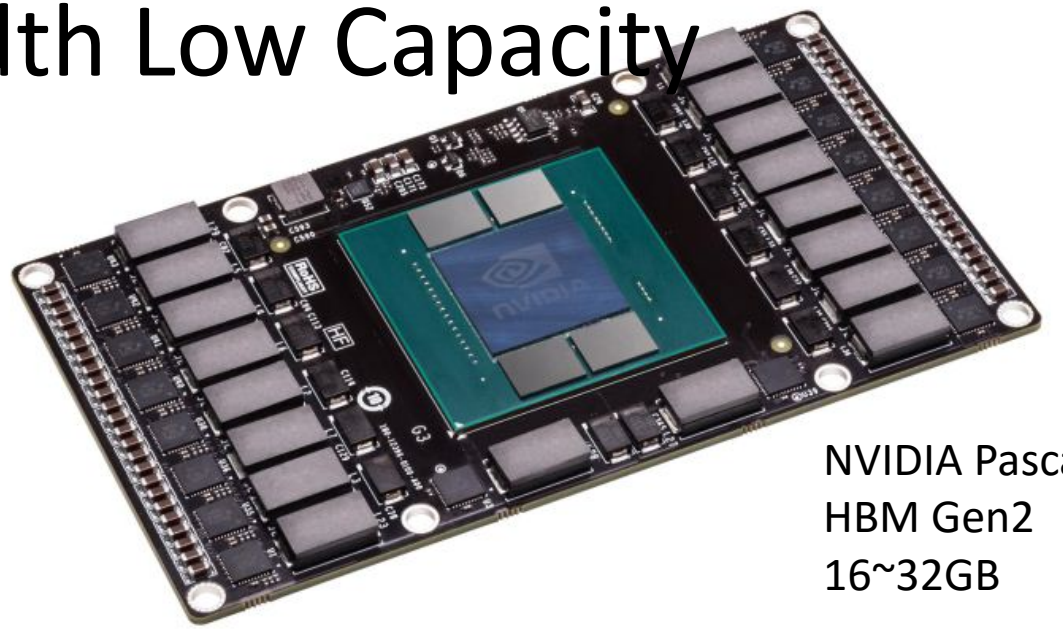


Post-K plan
> x10 FLOPS c.f. K
but < x10 mem capacity

Next Gen High BW 3D Stacked Memory
High Bandwidth Low Capacity



Intel KNL + In package memory 16GB



NVIDIA Pascal + HBM Gen2 16~32GB

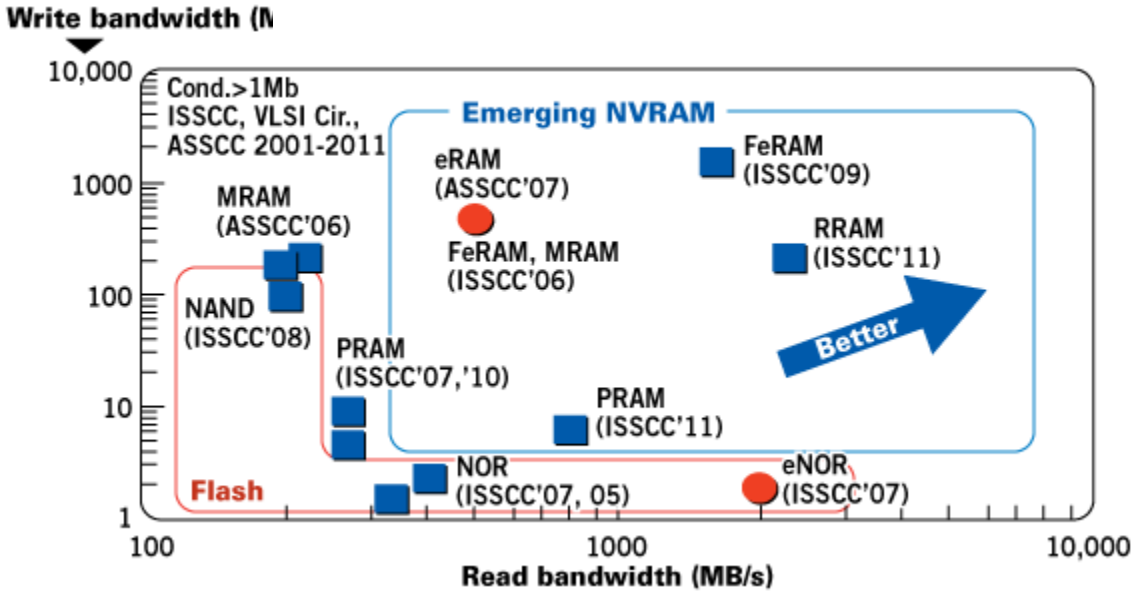
Fujitsu SPARC fxIX + HMC 32GB

Non-Volatile Memory and their Roadmaps

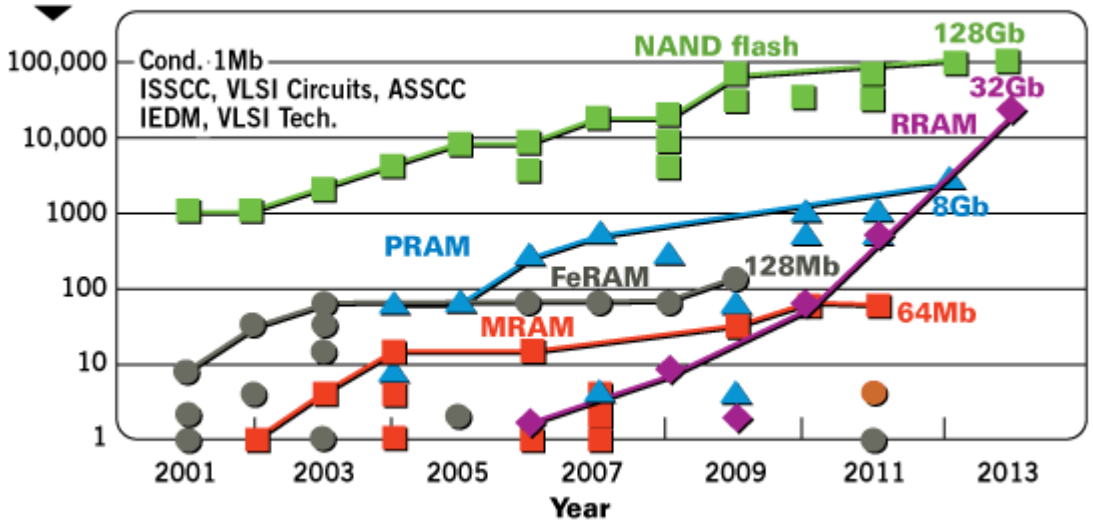
表1 DRAM、NANDフラッシュメモリと次世代メモリの比較

	DRAM	NAND	PCM	ReRAM	STT-RAM	CB-RAM	
セルファクター	4-6F^2	4F^2	4F^2	4F^2	6-14F^2	4-8F^2	6-8F^2
書き込み電圧	1-1.5V	18-20V	1.5-3V	~3V	1.0-1.5V	1.0-1.5V	1.0-1.5V
書き込み電流*	20μA	<1μA	100μA	25μA	49μA	1-20μA	50-80μA
書き込み時間*	5ns	10μs	<100ns	<10ns	<7ns	<10ns	<5μs
書き換え回数	10^{15}	10^5	10^{12}	10^6	10^{15}	10^{10}	10^5
データリテンション	<100ms	10years	10years	10years	10years	Hour/Day	10years
3D積層	No	Yes	Yes	Yes	No	Yes	Yes
マルチレベルセル	No	Yes	Yes	Yes	No	Yes	Yes

＊はビット単位



資料提供：エルピーダメモリ



Already Enterprise is Reducing Machine Requirements Significantly with Data Capacity Acceleration

2015 - Summary Performance Comparison by Storage Class

	Storage Class			IOPS FOB T4Q32	IOPS Steady State PTS-C T2Q16; PTS-E T4Q32; NVDIMM T128Q4			Bandwidth T1Q32; NVDIMM T32Q1		Response Time T1Q1	
	Category	Device Type	Capacity	RND 4KiB 100% W	RND 4KiB 100% W	RND 4KiB 65:35 RW	RND 4KiB 100% R	SEQ 128KiB 100% W	SEQ 128KiB 100% R	RND 4KiB 100% W Ave	RND 4KiB 100% W Max
	HDD & SSHD - PTS-C v1.2										
1	SSHD	7,200 RPM 2.5" SATA Hybrid	500 GB	3,398	77	62	113	79 MB/s	76 MB/s	12.55 mSec	163.10 mSec
2	SAS HDD	15,000 RPM 2.5" SAS HDD	146 GB	652	133	248	514	147 MB/s	174 MB/s	2.24 mSec	24.54 mSec
	CLIENT SSDs - PTS-C v1.2										
3	mSATA	mSATA 1.8" MLC	250 GB	63,127	5,927	13,545	79,423	300 MB/s	526 MB/s	0.13 mSec	11.86 mSec
4	M.2 x4 AHCI	M.2 x4 Gen 3 2280 MLC	128 GB	77,757	6,307	17,041	172,881	623 MB/s	1,895 MB/s	0.16 mSec	54.19 mSec
5	SATA	SATA6Gb/s 2.5" MLC	400 GB	52,723	34,406	50,646	61,178	428 MB/s	475 MB/s	0.082 mSec	3.13 mSec
	ENTERPRISE SSDs - PTS-E v1.1										
6	SATA	SATA 12Gb/s 2.5" eMLC	800 GB	63,185	38,478	46,911	83,648	430 MB/s	507 MB/s	0.05 mSec	0.99 mSec
7	M.2 x4 NVMe	M.2 x4 Gen3 2280 eMLC WCE	256 GB	29,210	12,155	31,695	248,929	1,100 MB/s	2,158 MB/s	0.09 mSec	23.21 mSec
8	SAS	SAS 12Gb/s 2.5" eMLC	400 GB	149,731	97,167	133,738	187,262	651 MB/s	1,051 MB/s	0.072 mSec	4.72 mSec
9	U.2 NVMe	SFF 8639 4 lane 2.5" MLC	1600 GB	470,122	113,260	238,143	585,884	1,286 MB/s	2,232 MB/s	0.032 mSec	1.00 mSec
10	PCIe x8 AHCI	PCIe 8 Lane Edge Card MLC	1400 GB	160,164	83,021	234,205	750,846	614 MB/s	2,903 MB/s	0.014 mSec	0.56 mSec
	NVDIMM-N - PTS-E v1.1										
11	NVDIMM-N x4	Memory Channel DDR4 x4 Modules	32 GB	2,384,692	2,792,176	2,789,921	3,002,085	35,453 MB/s	55,111 MB/s	0.0022 mSec	0.48 mSec

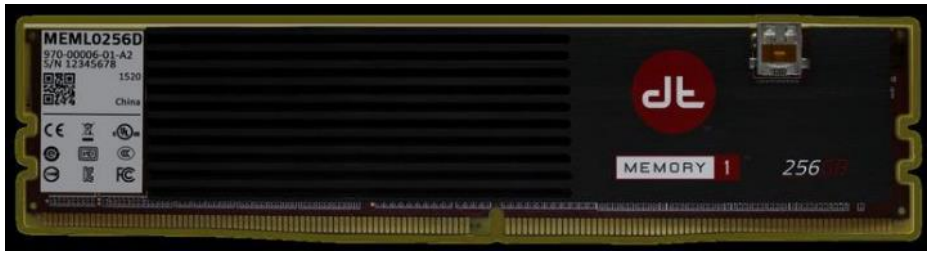
All measurements (rows 1-10) taken by Calypso on the RTP/CTS pursuant to SNIA PTS-E v1.1 / PTS-C v1.2.
IOPS Thread and Queue settings are T2Q16 for HDD/SSHD & Client SSDs, T4Q32 for Enterprise SSDs and T128Q4 for NVDIMM-N.
NVDIMM-N data taken on SuperMicro X10DRI, dual E5 2670V3 8 core cpus, 16GB DRAM using Intel Open Source block IO driver.

SNIA Solid State Storage Initiative — CALYPSO Systems

Source: http://www.storagenewsletter.com/wp-content/uploads/2015/09/sniaSSSINVDIMM-N.jpg

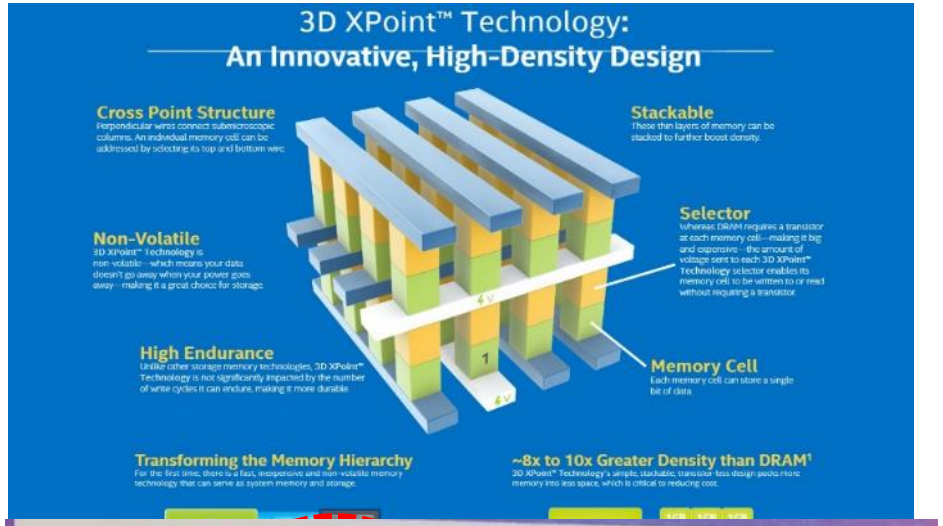
	Current Search Solution	Memory1 Enhanced Solution
Total Index Size	10,000 TB	10,000 TB
Required Servers	100,000	10,000
Memory/Server	128 GB	128 GB + 1 TB Memory1
Index Size/Server	100 GB	1000 GB
Cost/Server	$3,000	$8000
Total Server CAPEX	**$300,000,000**	**$80,000,000**

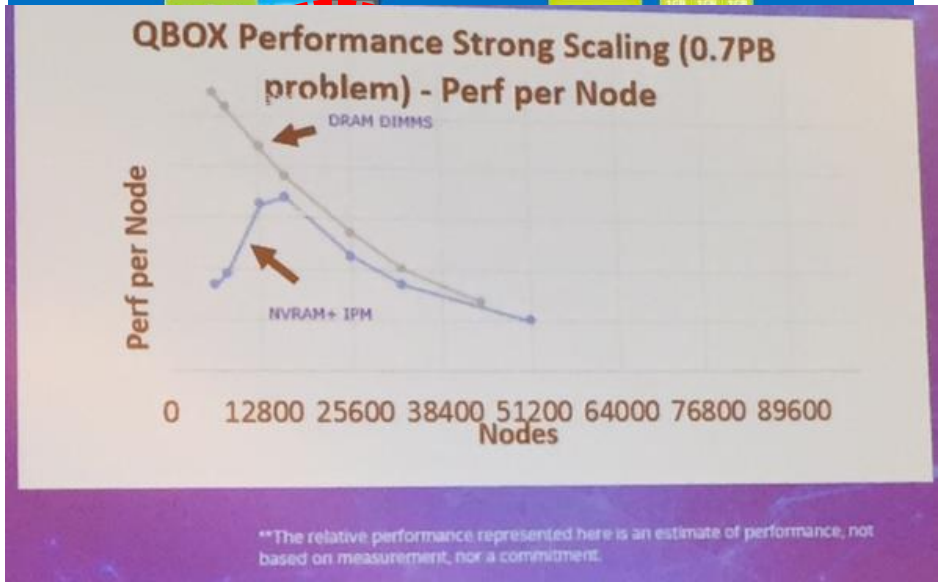
Source: http://www.storagenewsletter.com/wp-content/uploads/2015/09/sniaSSSINVDIMM-N.jp



DIABLO Memory 1 256GB NVDIMM-N

Accelerating “Big Data” Non-Volatile Memory-based Exascale Architectures





From Al Gara Keynote, IEEE Custer 2015

- Accelerating
  - HPC Apps, Big Data-HPC Apps
- Using
  - Exascale machines w/NVM (Flash, ReRAM, 3D Xpoint, …)
  - BYTES, not FLOPS
- While
  - Reducing Bandwidth, Exploiting Locality
  - Dealing with higher write cost
  - Dealing with low durability
  - Maintaining Programmability
  - Exploiting other system assets such as hybrid electro-optical Exabit interconnect
  - …

Software Technology that Deals with Deeper Memory Hierarchy is important for exascale and beyond

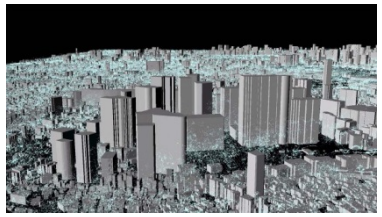
Comm/BW reducing algorithms

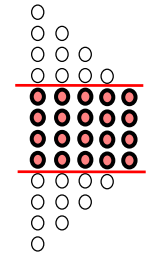
+

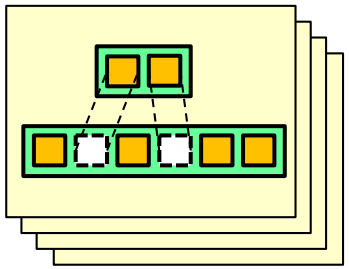
System software for mem hierarchy mgmt

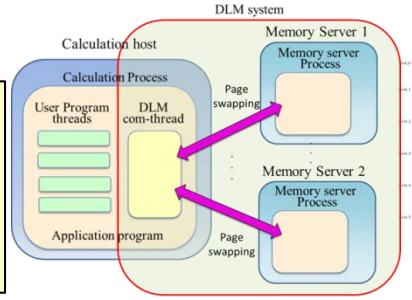
+

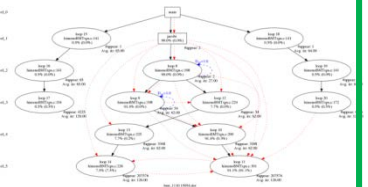
HPC Architecture with hybrid memory devices

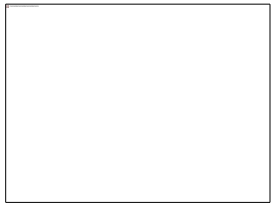






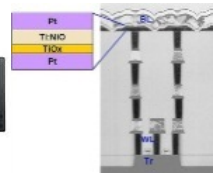


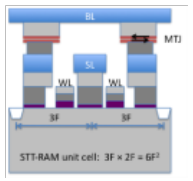




HMC, HBM

O(GB/s) Flash





Next-gen NVM

Problems:

- *Appropriate Architecture Design?*
- *Maintain performance of small memry?*
- *Ease of programming?*
- *Dealing with Legacy Code?*

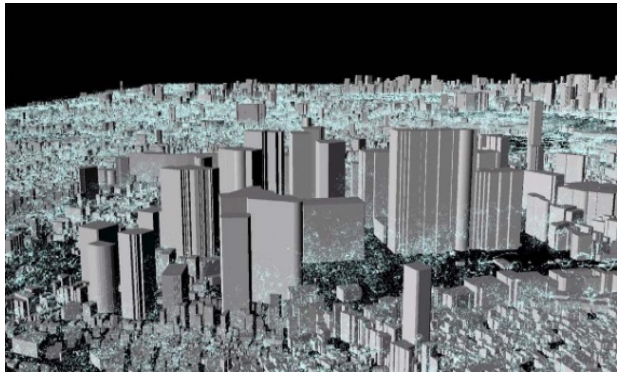
Target: Realizing ***extremely Fast&Big simulations*** of {O(1000PF/s) or O(100PB/s)} & O(100PB)

Bandwidth Reducing Algorithms for High-Bandwidth, High Capacity Applications towards Exascale

- Bandwidth reducing Stencil Algorithms for Hiearchical Memory systems e.g. GPUs via temporal blocking
- Bandwidth reducing format for SpMV on many-core, small cache systems e.g. GPUs
- Bandwidth Big Data Algorithm for GPUs NVMs

Communication Reducing Methods for Large-scale GPU-based CFD Simulations [Endo, Takasaki, Matsuoka, IEEE ICPADS 2015]

- CFD Simulations on GPU supercomputers
  - Stencil computations successfully accelerated by high FLOPS and high BW of GPUs
  - However, problem sizes (Bytes) are limited by device memory capacity
    - LES Wind Simulation
      - by Onodera, Aoki et al.
      - 10km^2 area in Tokyo was simulated with 1m resolution, by using 4,032 GPUs of TSUBAME2.5 supercomputer
      - However, the simulation size is still limited by (aggregated) device memory capacity. How can we exceed the limit?



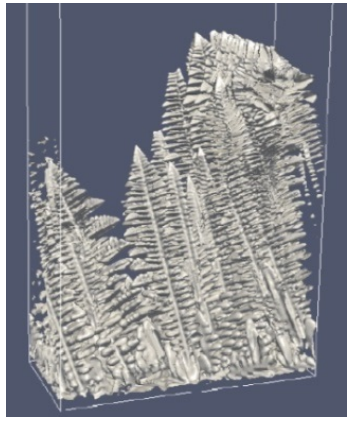
Wind Simulation of Tokyo Area

Stencil Computations

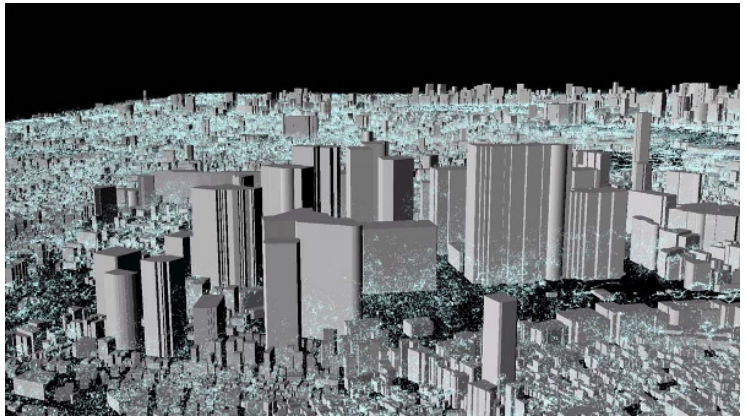
Important kernels for various simulations (CFD, material…)

- Memory intensive → successful on GPU accelerators

ASUCA weather simulator

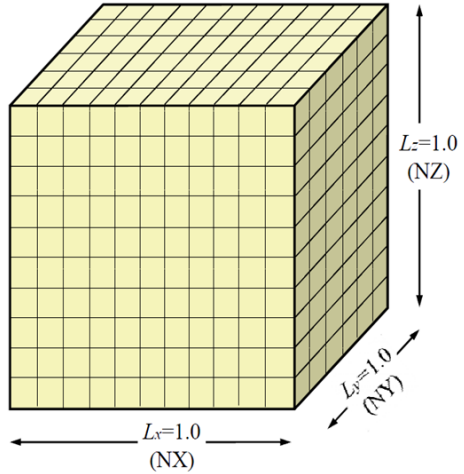


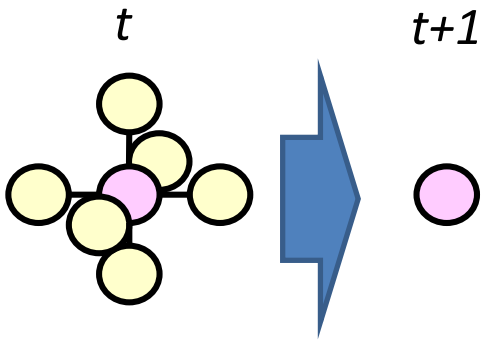
Phase-Field computation (2011 Gordon Bell)



Air flow simulation

=





Advanced Lattice-Boltzmann (LBM) Simulation of the Entire Downtown Tokyo [Onodeara&Aoki]

Recreate “Virtual Tokyo Downtown”

10km x 10km x 500m area @ 1m resolution

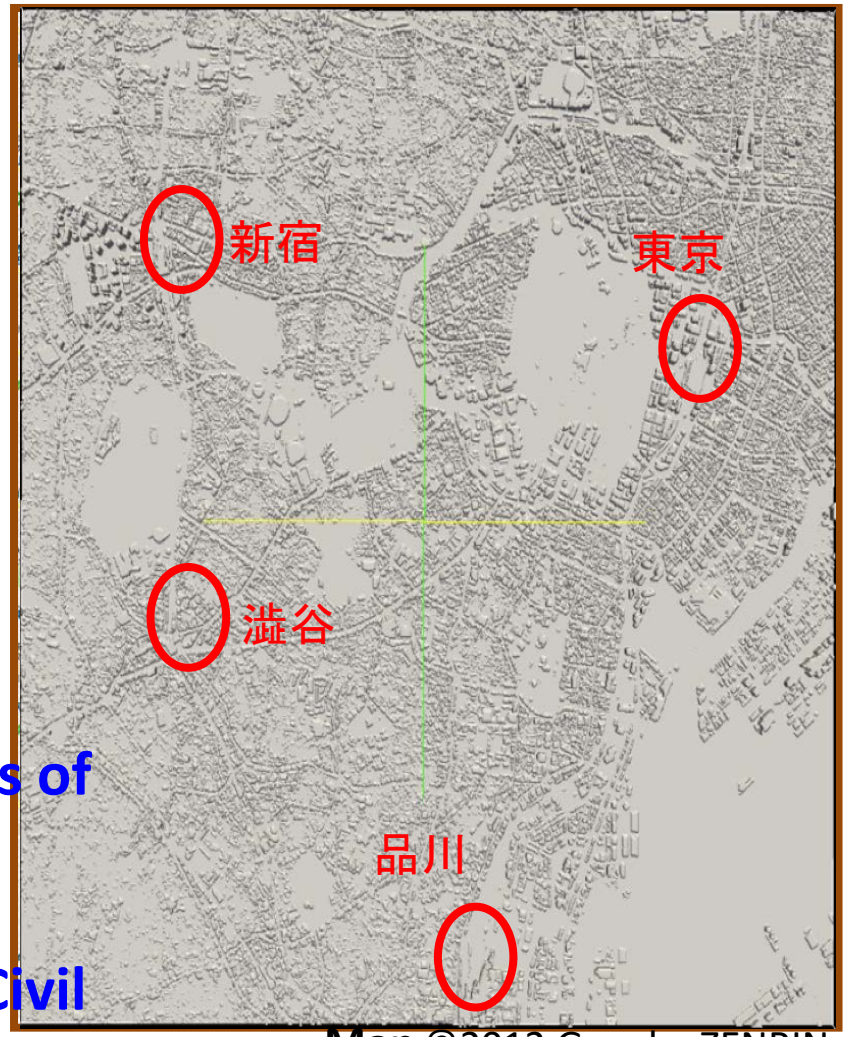
Map Data: Zenrin, Google

Building Data: Pasco Co. Ltd.TDM 3D

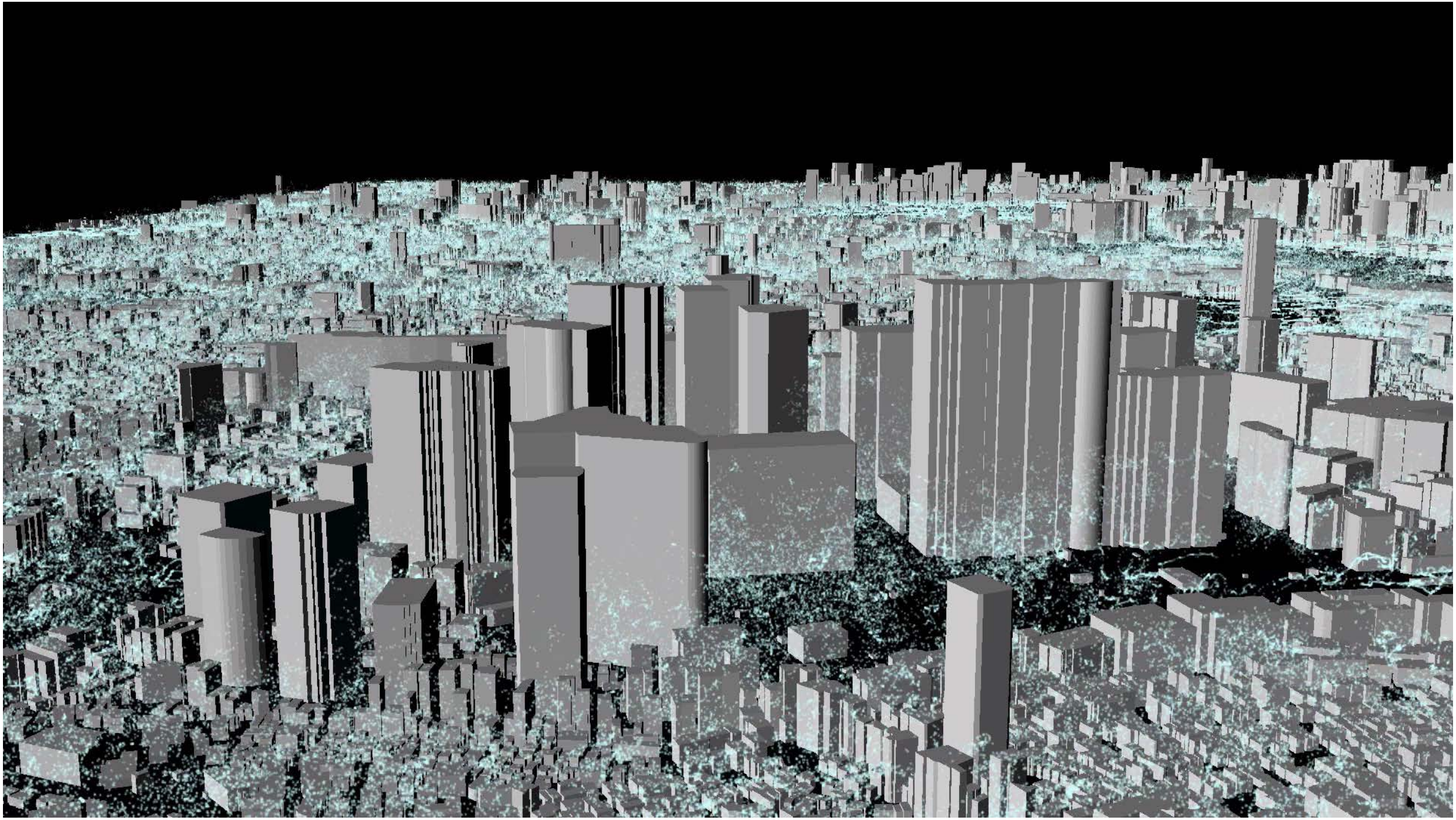
Massively parallel LBM-LES turbulent flow simulation utilize coherent Structured SGS model

TSUBAME2.0 4000GPU 0.592 Petaflops (15% of peak), millions of parallelism

Many “Heat Island” Phenomenon, Dissipation of Pollutants, Civil Planning for Skyskrapers, etc.

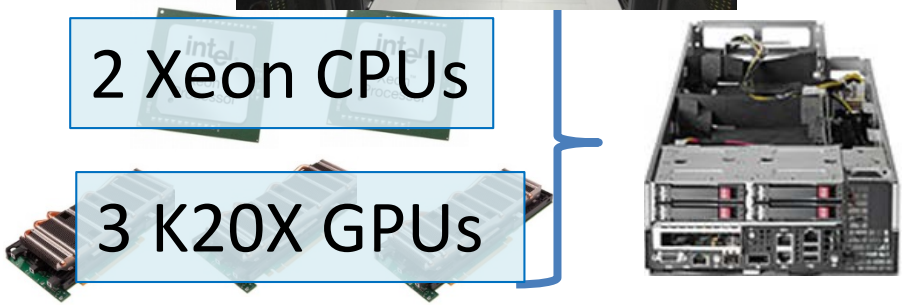


Map ©2012 Google, ZENRIN



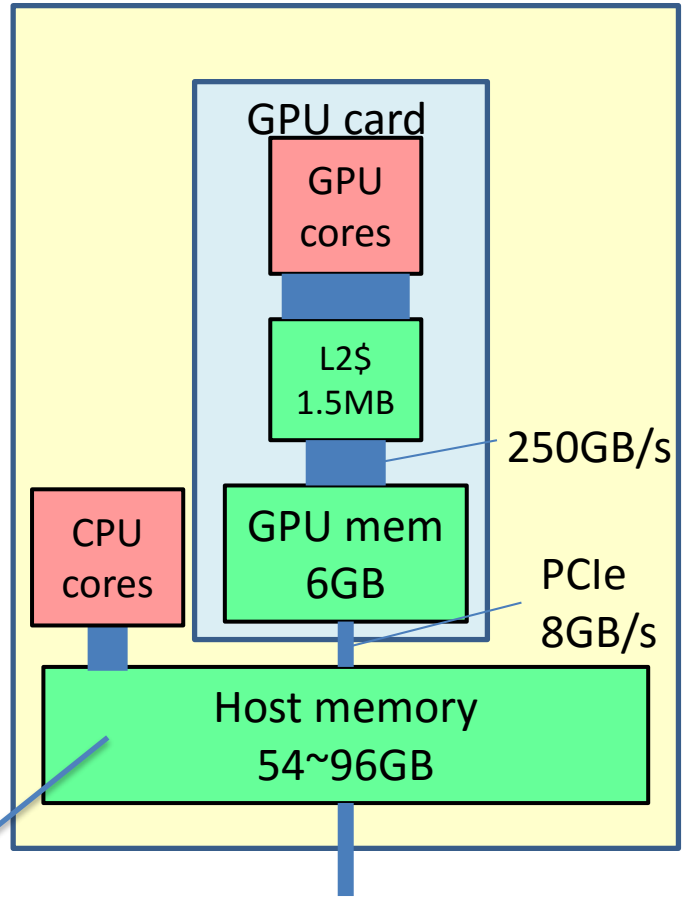
Hybrid Supercomputers with Memory Hierarchy

Tokyo Tech TSUBAME2.5 Supercomputer with 5.7PFlops performance



× 1408

Simplified Node Architecture

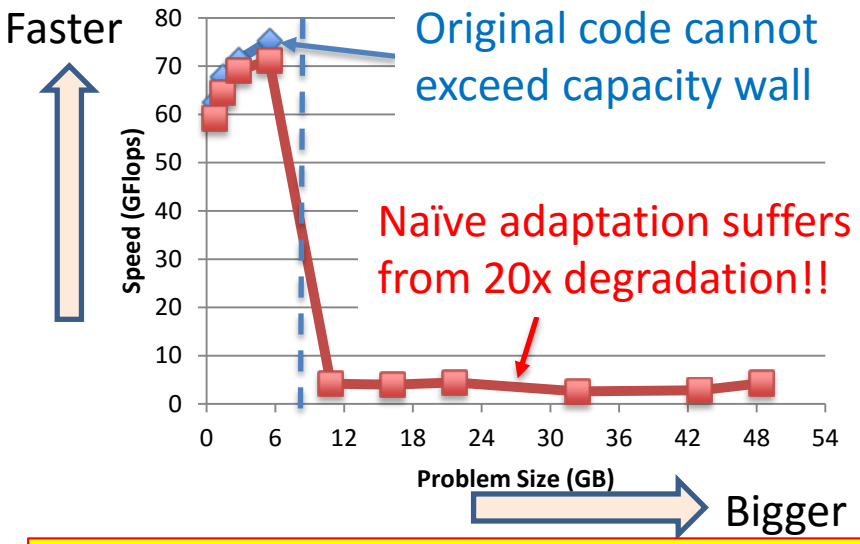


Harnessing capacity of larger host memory is necessary, but naïve method is harmful

Two Issues for Extreme Big Simulations

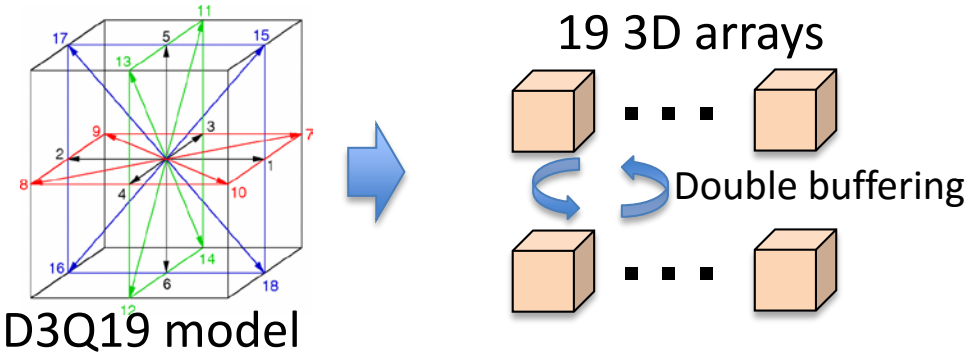
Performance issue

- Naïve usage of memory hierarchy causes 20x performance degradation
- Due to large PCIe costs



Productivity issue

- Typically , temporal blocking introduces heavy code rewriting
- Original Wind simulation code is already complex
  - ~15,000 Lines of code with MPI/CUDA
  - LBM with hybrid boundary conditions (periodic and Neumann)

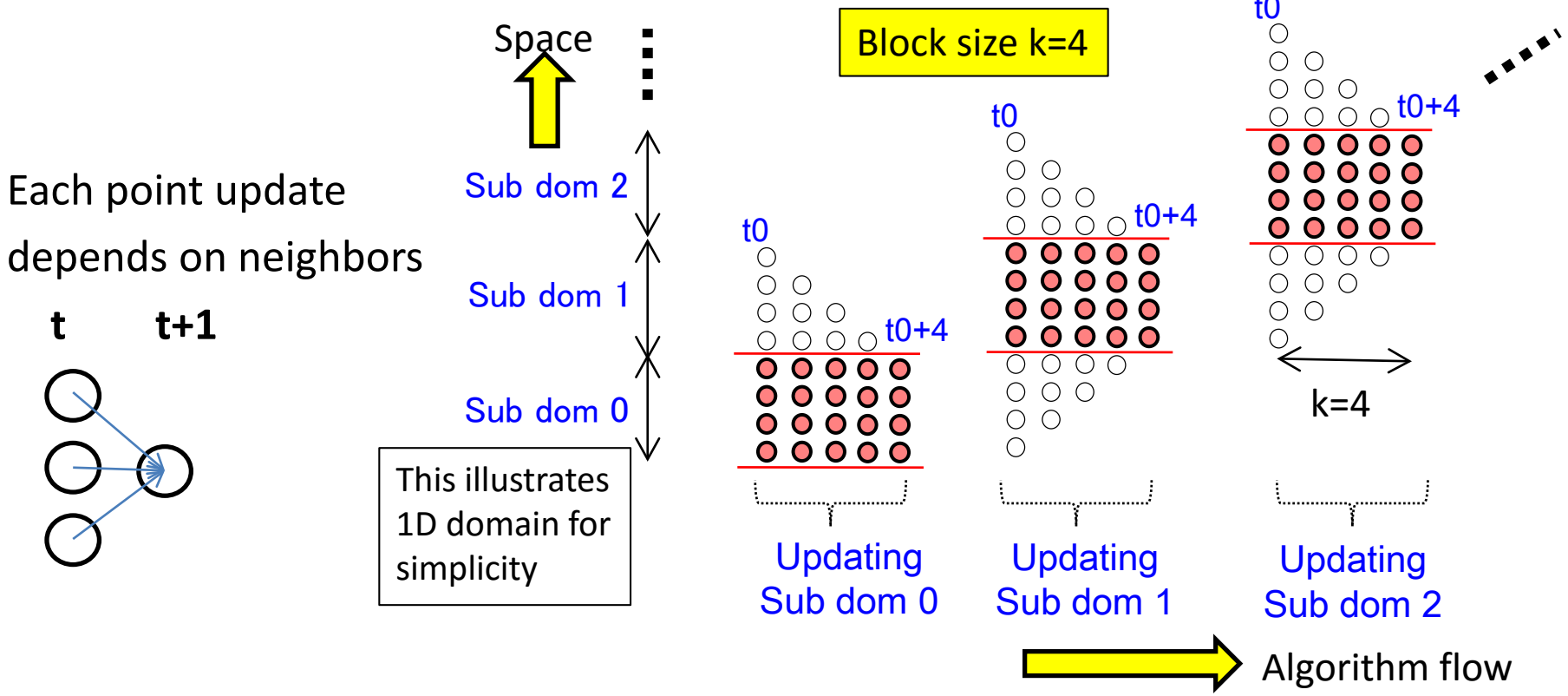


We need locality improvement / comm. reducing → *Temporal blocking* technique is adopted

We reduce programming costs by using runtime library for memory swapping, HHRT

Temporal Blocking (TB)[Wolf 91]: Locality Improvement of Stencil Computations

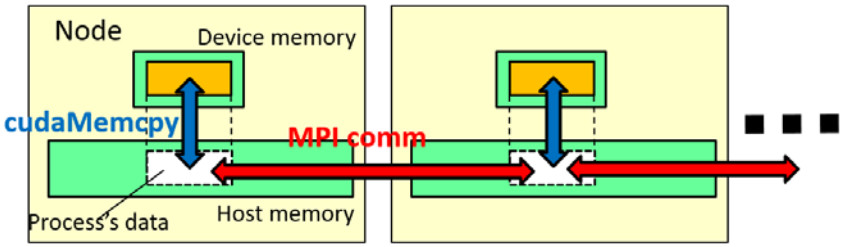
- The domain is divided into sub-domains
- When we pick up a sub-domain, we proceed its computation for several (=k) time steps at once
- This optimization introduces
  - re-structuring of loop structure
  - redundant computations



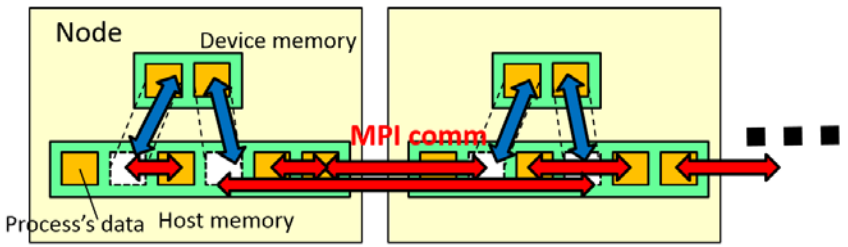
Using HHRT Library for Reducing Code Re-structuring Costs

- HHRT (Hybrid Hierarchical RunTime) [Endo, IEEE Cluster 13]
  - Supports transparent data swapping between device memory and host memory
  - Supports oversubscribed execution of processes
  - Works as a wrapper library of MPI and CUDA
    - MPI/CUDA applications run on HHRT with very small modifications

Typical execution w/o HHRT

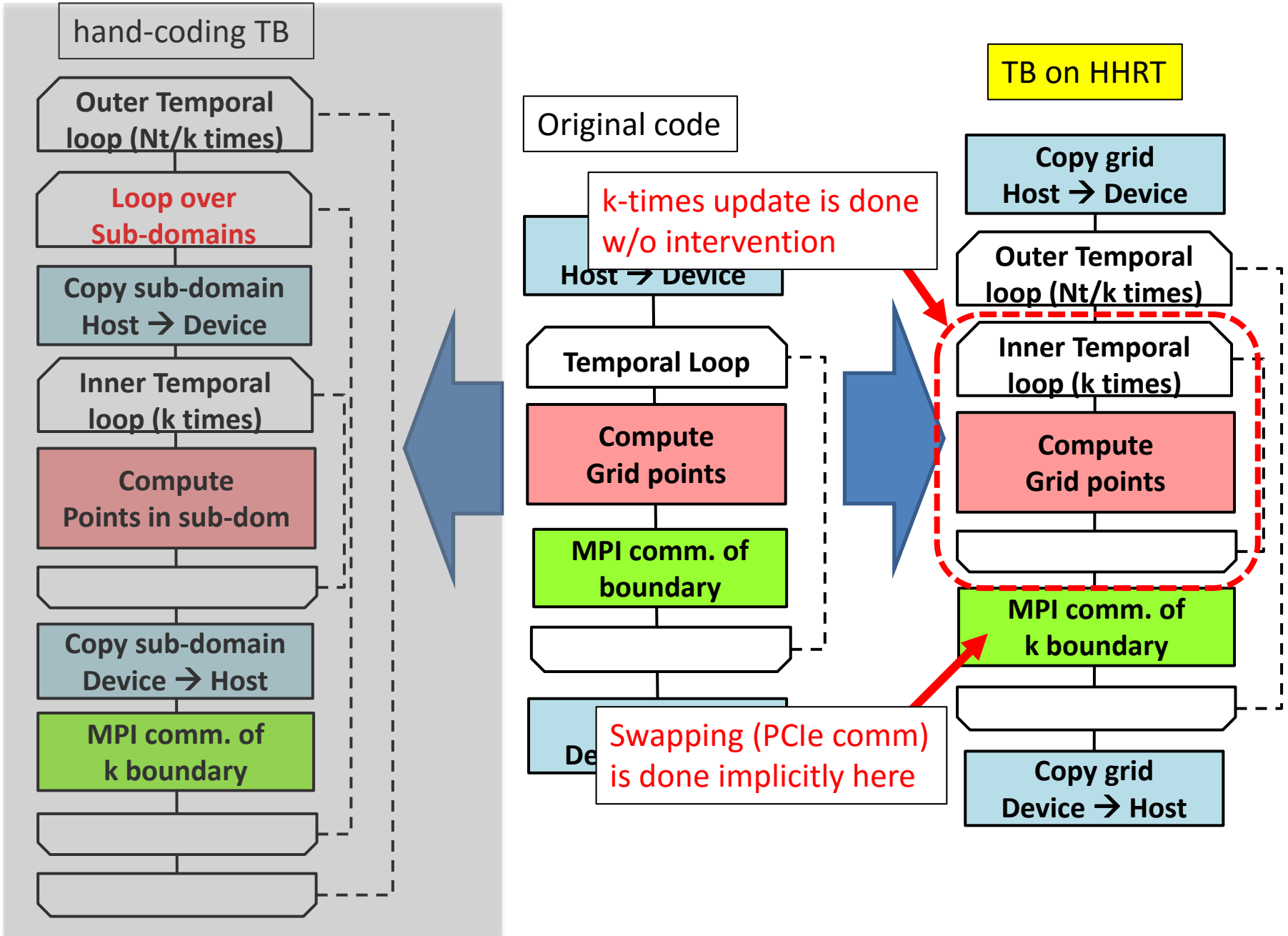


Oversubscribed execution with HHRT



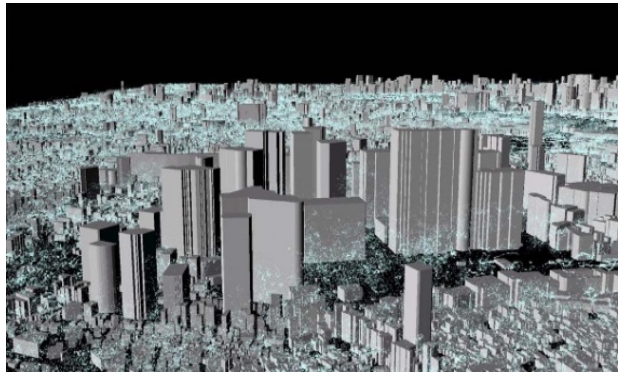
- Several processes time-share the capacity of device memory
- Aggregated data size among local processes can exceed device memory

Implementing TB on HHRT (1)



Implementing TB on HHRT (2)

- Is it easier with HHRT?
  - No additional sub-domain loop required
  - On the other hand, changing temporal loop is programmers' responsibility



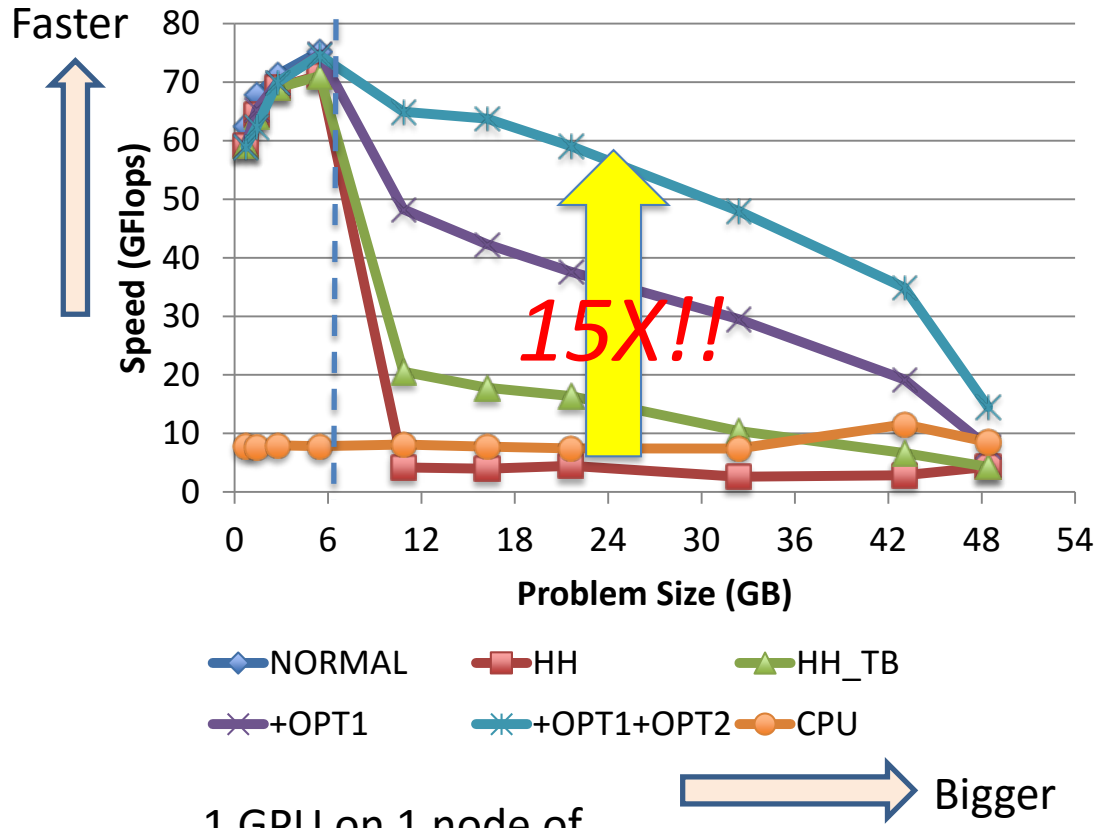
Original code	15,561 (lines)
For basic temporal blocking	+148
Optimization 1	+969
Optimization 2	+52

Optimization 1: To reduce swapping frequency, we re-structured communication part of the original code
Optimization 2: To reduce cost of each swapping, we annotated the lifespan information of array data

To be automated with future version of HHRT

Performance

- Both problem scale and high performance are achieved with the proposed approach



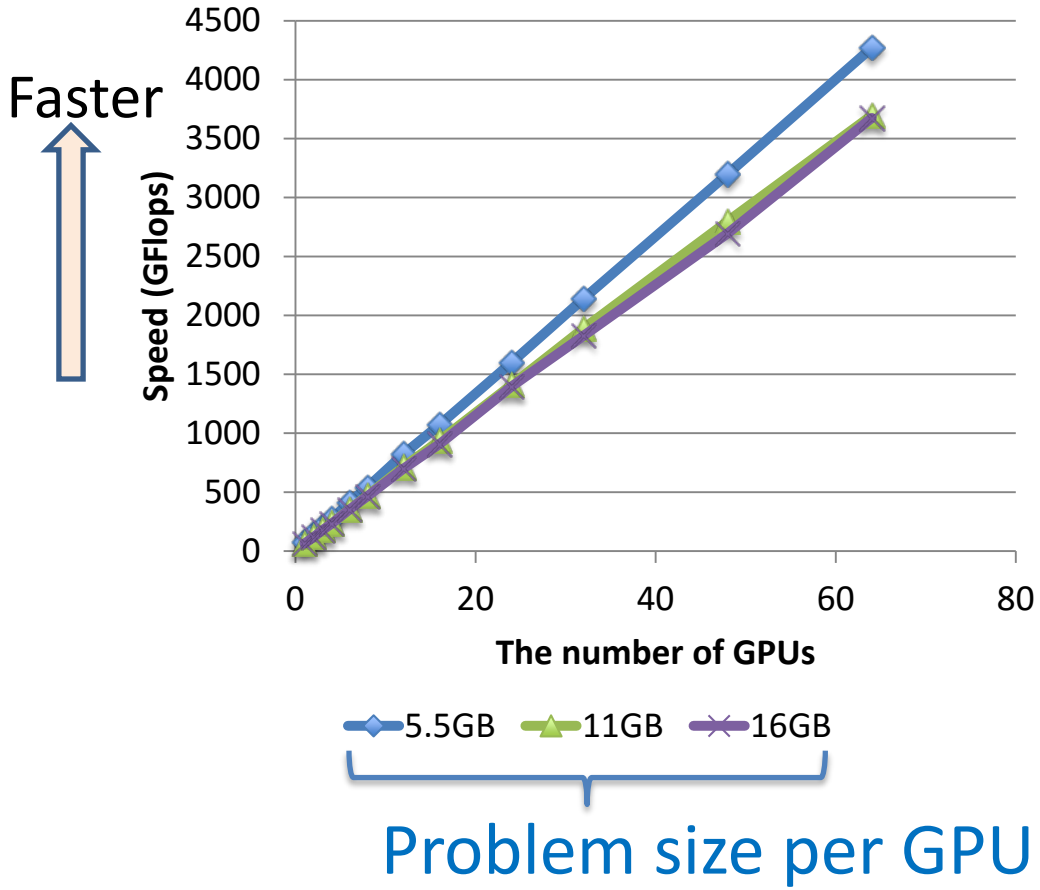
1 GPU on 1 node of TSUBAME2.5 is used

- With 4x larger problem sizes, the performance is kept to 79% of the original version

→ 15x seedup over naïve big version!

- Still tradeoff between size and performance
  - Larger cost when larger number of processes are sharing a GPU

Weak Scalability



1~64 TSUBAME2.5 nodes (1 GPU per node) are used

- We observe high scalability
- 3.7TFlops is achieved with problem size of 1TB (= 16GB x 64)

=> x3 bigger problem size, 13% performance loss

Similar result for Gordon Bell Dandrite Simulation

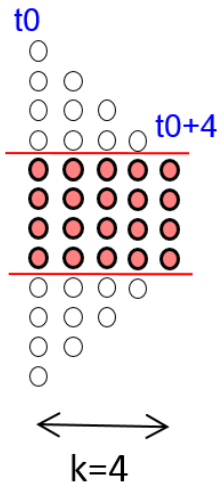
Performance Model towards Automatic Parameters Tuning

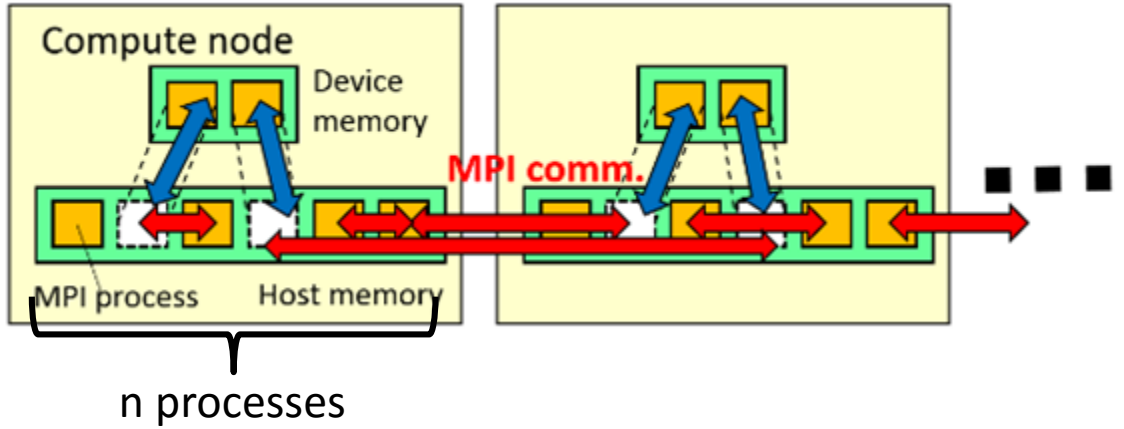
Several runtime parameters should be tuned for better performance

- Previous graphs show performance with tuned parameters

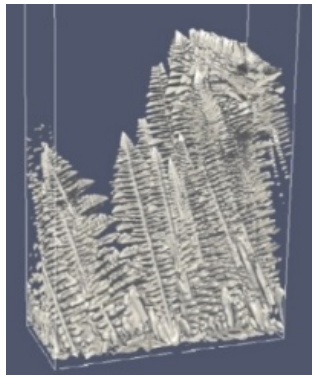
- *k*: Temporal blocking factor
  - If k is too small, we suffer from PCIe cost
  - If k is too large, we suffer from redundant computation
- *n*: The number of processes per GPU
  - If n is too small, data per process may not fit device memory
  - If n is too large, we suffer from larger redundant computation, etc.

→ We constructed a performance model to obtain the best parameters

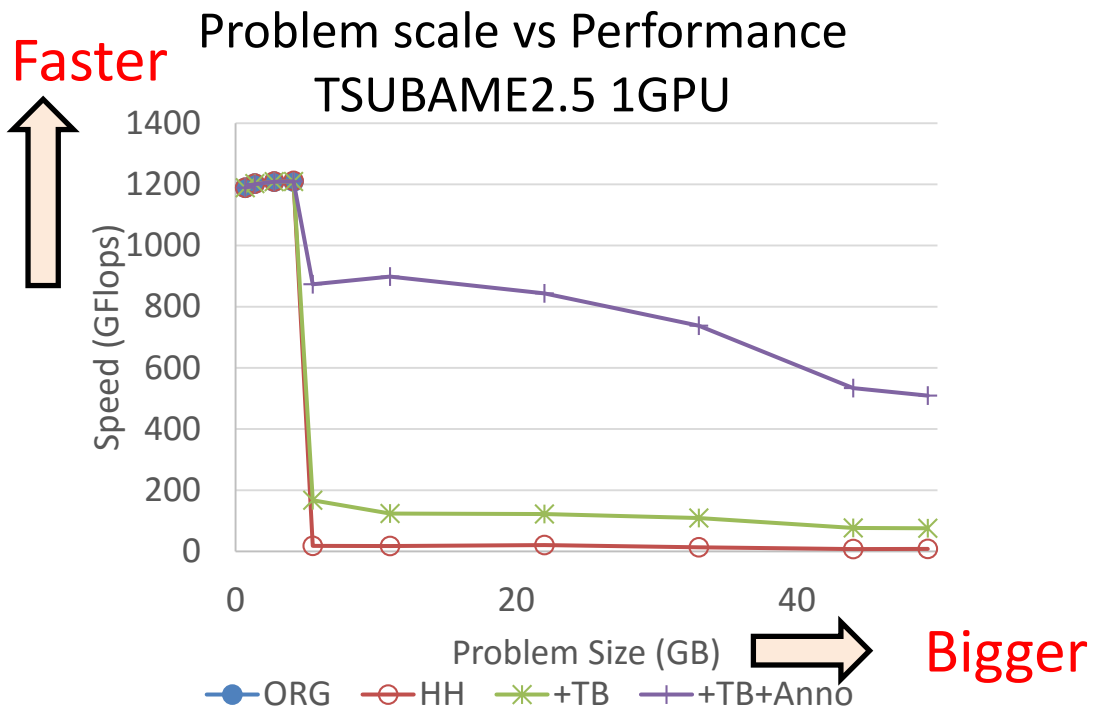


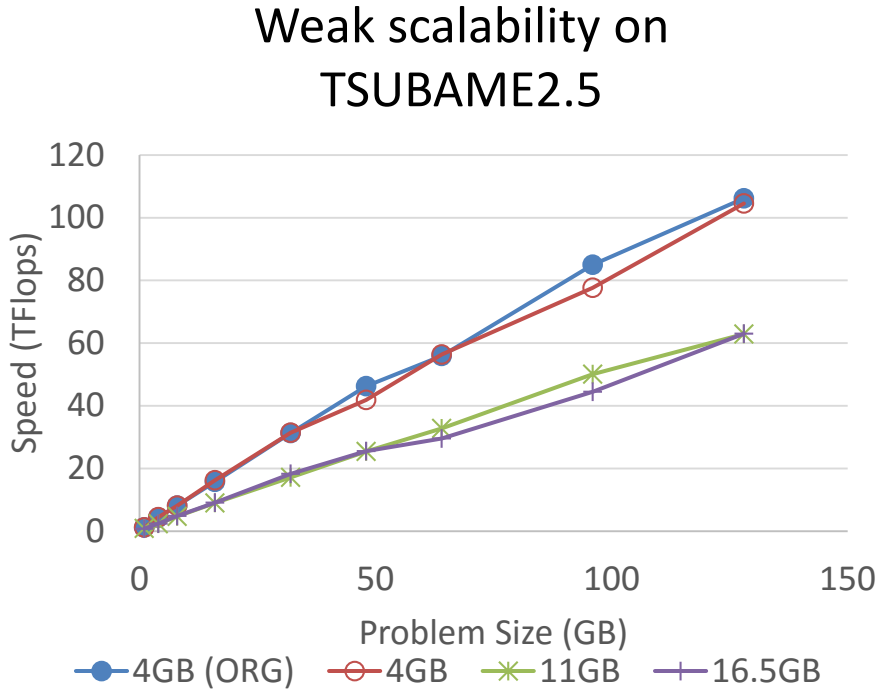


Performance of Dendrite Simulation Application



- We applied our approach to another stencil application, Dendrite simulation (Gordon Bell Prize 2011)

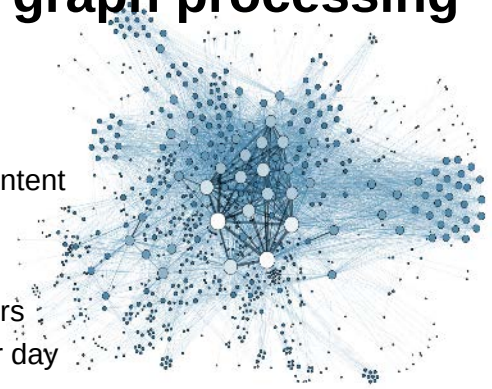




Extreme Big Data Examples

Rates and Volumes are extremely immense

Social NW – large graph processing

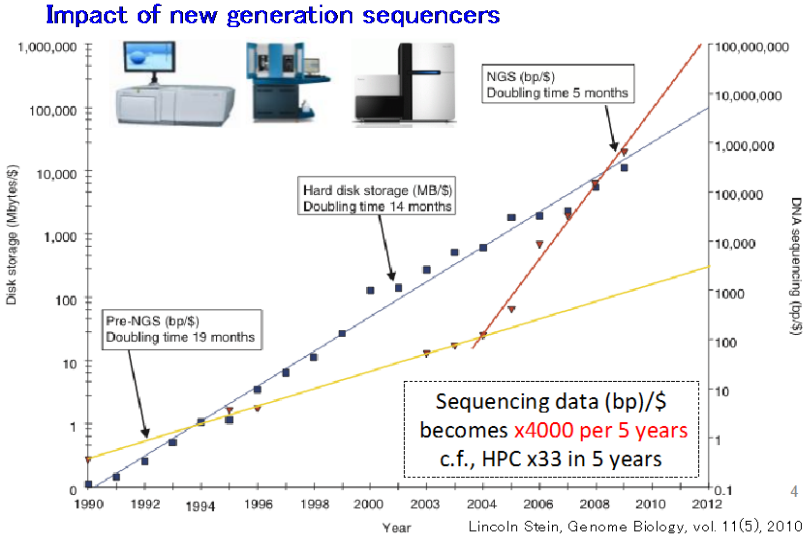


- Facebook
  - ~1 billion users
  - Average 130 friends
  - 30 billion pieces of content shared per month
- Twitter
  - 500 million active users
  - 340 million tweets per day
- Internet
  - 300 million new websites per year
  - 48 hours of video to YouTube per minute
  - 30,000 YouTube videos played per second

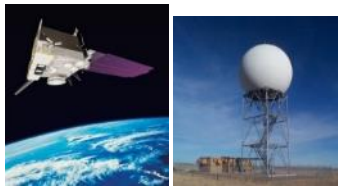
Social Simulation

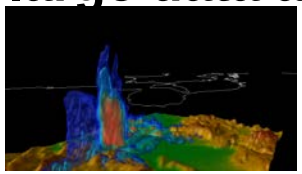
- Applications
  - Target Area: Planet (Open Street Map)
  - 7 billion people
- Input Data
  - Road Network for Planet: 300GB (XML)
  - Trip data for 7 billion people 10KB (1trip) x 7 billion = 70TB
  - Real-Time Streaming Data (e.g., Social sensor, physical data)
- Simulated Output for 1 Iteration
  - 700TB

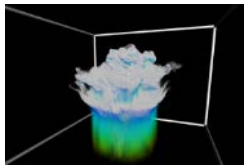
Genomics advanced sequence matching

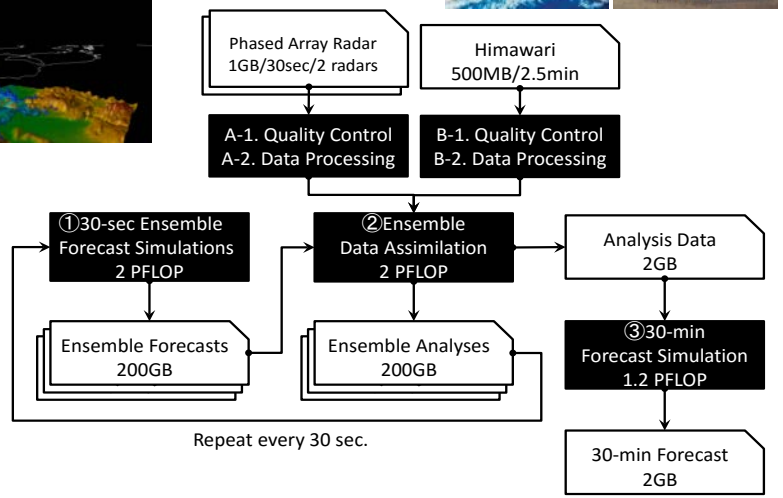


Weather – real time large data assimilation









NOT simply mining Tbytes Silo Data

Peta~Zetabytes Data

Ultra High-BW Data Stream

Highly Unstructured, Irregular

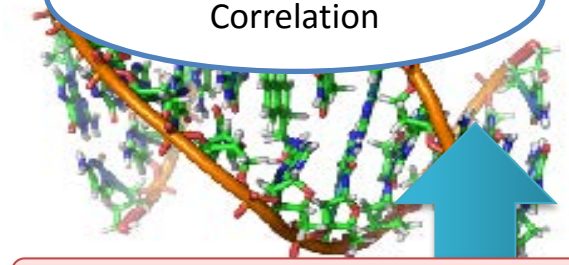
Complex correlations between data from multiple sources

Extreme Capacity, Bandwidth, Compute All Required

100,000 Times Fold EBD "Convergent" System Architecture

Akiyama Group

Large Scale Genomic Correlation

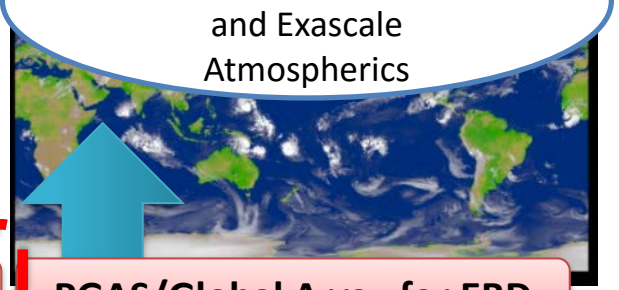


Suzumura Group

Large Scale Graphs and Social Infrastructure Apps

Miyoshi Group

Data Assimilation in Large Scale Sensors and Exascale Atmospherics



SQL for EBD | **Graph Framework** | **PGAS/Global Array for EBD**

Message Passing (MPI, X10) for EBD | **Workflow/Scripting Languages for EBD** | **MapReduce for EBD**

Programming Layer

Matsuoka Group

EBD Abstract Data Models
(Distributed Array, Key Value, Sparse Data Model, Tree, etc.)

EBD Algorithm Kernels
(Search/ Sort. Matching, Graph Traversals, , etc.)

Basic Algorithms Layer

EBD File System | **EBD Data Object** | **EBD Burst I/O Buffer** | **EBD Network Topology and Routing**

System SW Layer

Tatebe Group

Kolbuchi Group

NVM
(FLASH, PCM, STT-MRAM, ReRAM, HMC, etc.)

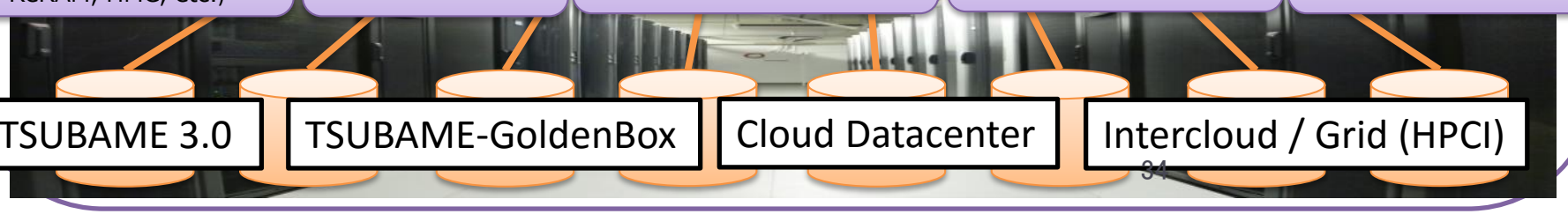
HPC Storage

Web Object Storage

Interconnect
(InfiniBand 100GbE)

Network
(SINET5)

Big Data & SC HW Layer

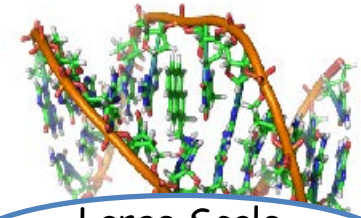


TSUBAME 3.0 | TSUBAME-GoldenBox | Cloud Datacenter | Intercloud / Grid (HPCI)

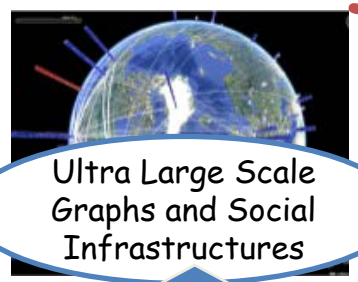
Big Data & SC HW Layer

JST-CREST "Extreme Big Data" Project (2013-2018)

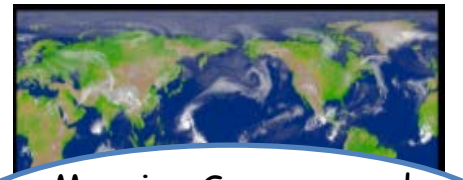
Future Non-Silo Extreme Big Data Scientific Apps



Large Scale Metagenomics



Ultra Large Scale Graphs and Social Infrastructures



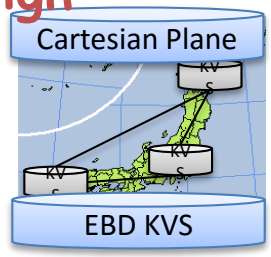
Massive Sensors and Data Assimilation in Weather Prediction

Co-Design Co-Design Co-Design

EBD System Software incl. EBD Object System

EBD Bag

Graph Store



Given a top-class supercomputer, how fast can we accelerate next generation big data c.f. Clouds?

Issues regading Architectural, algorithmic, and system software evolution?

Use of GPUs?

Convergent Architecture (Phases 1~4)
Large Capacity NVM, High-Bisection NW

Cloud IDC
Very low BW & Efficiency
Highly available, resilient

Supercomputers
Compute&Batch-Oriented
More fragile

Graph500 “Big Data” Benchmark

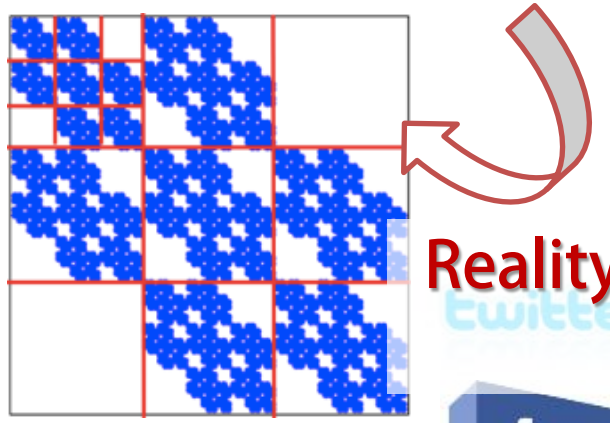
Kronecker graph BSP Problem

$$\arg\max_{\Theta} P(\quad | \quad \leftarrow \text{Kronecker} \ \Theta)$$

A: 0.57, B: 0.19
C: 0.19, D: 0.05

1	1	0
1	1	1
0	1	1

G_1



G_4 adjacency matrix

November 15, 2010

HPCwire

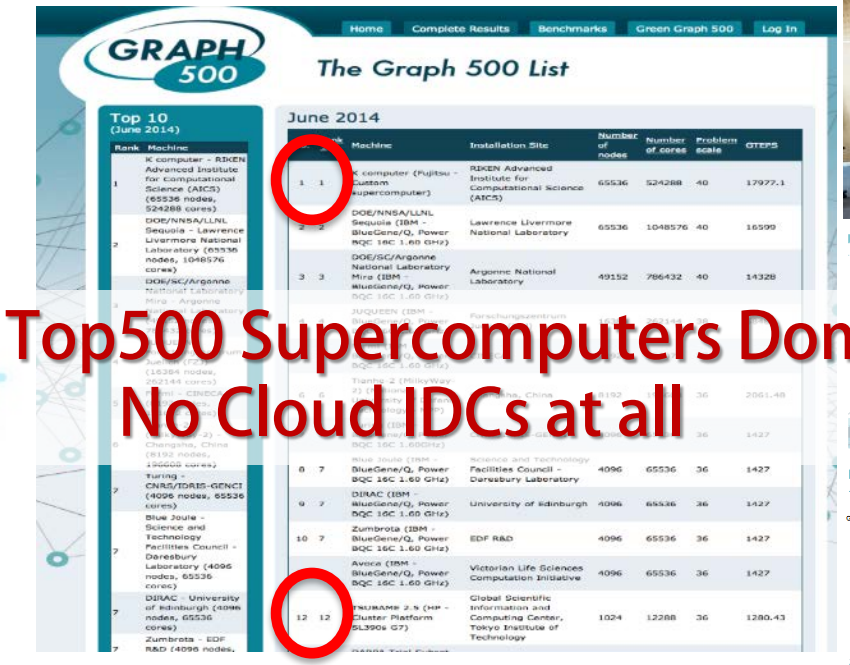
Graph 500 Takes Aim at a New Kind of HPC

Richard Murphy (Sandia NL => Micron)

“ **I expect that this ranking may at times look very different from the TOP500 list. Cloud architectures will almost certainly dominate a major chunk of part of the list.**”

The 8th Graph500 List (June2014): K Computer #1, TSUBAME2 #12

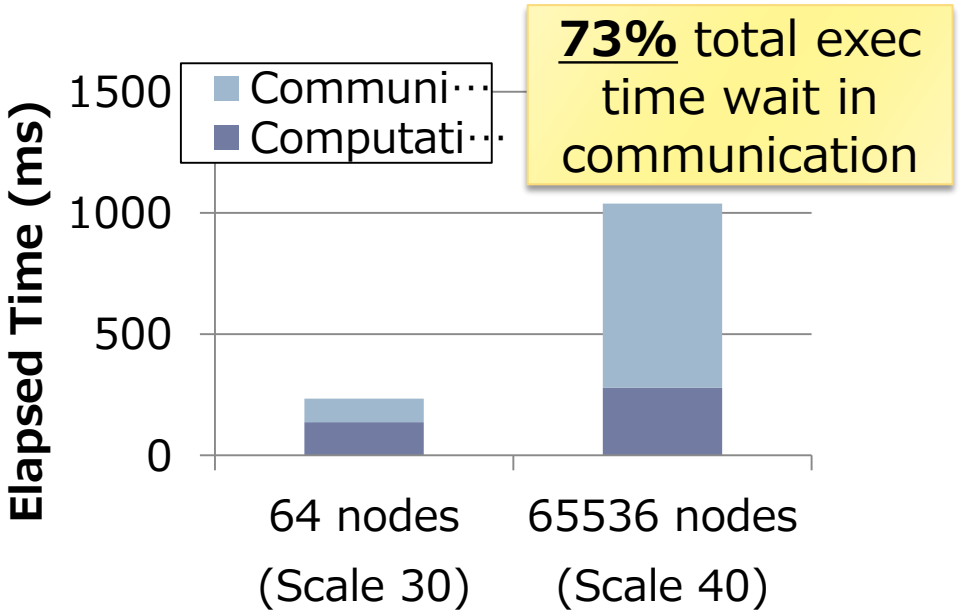
Koji Ueno, Tokyo Institute of Technology/RIKEN AICS



Reality: Top500 Supercomputers Dominate
No Cloud IDCs at all

The Graph500 – June 2014 and June 2015

K Computer #1 Tokyo Tech[EBD CREST] Univ. Kyushu [Fujisawa Graph CREST], Riken AICS, Fujitsu



73% total exec time wait in communication

88,000 nodes,
700,000 CPU Cores
1.6 Petabyte mem
20GB/s Tofu NW

K computer

LLNL-IBM Sequoia
1.6 million CPUs
1.6 Petabyte mem

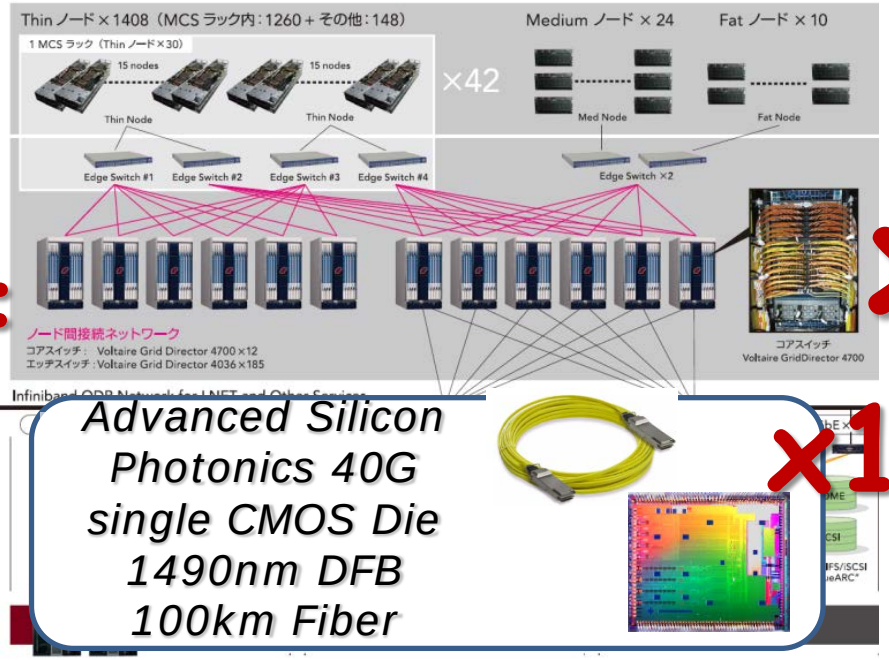
IBM

List	Rank	GTEPS	Implementation
November 2013	4	5524.12	Top-down only
June 2014	1	17977.05	Efficient hybrid
November 2014	2		Efficient hybrid
June 2015	**1**	**38621.4**	Hybrid + Node Compression

*Problem size is weak scaling "Brain-class" graph

Supercomputer Tokyo Tech. Tsubame 2.0 #4 Top500 (2010)

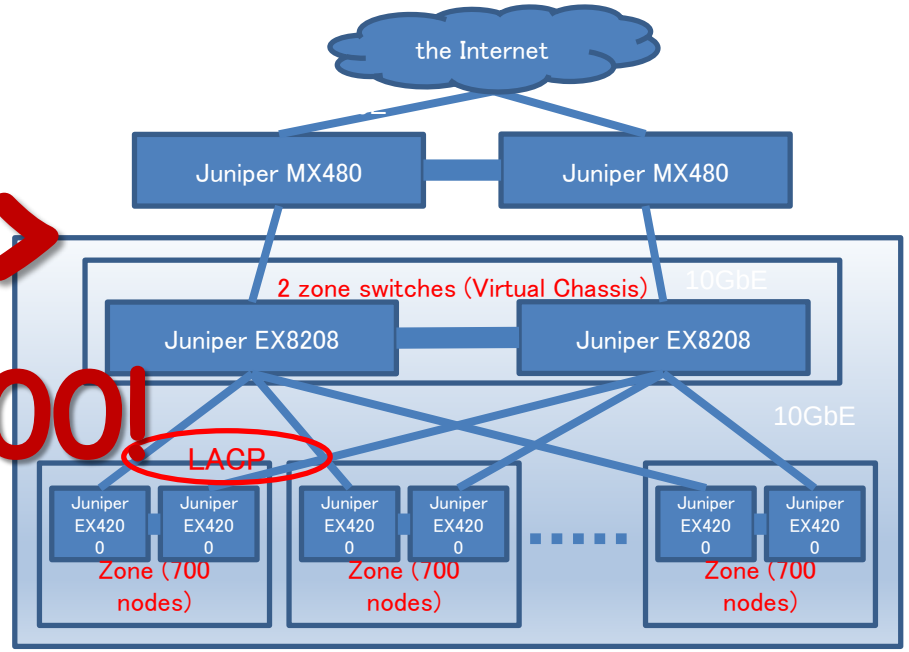
~=



>>

x1000!

A Major Northern Japanese Cloud Datacenter (2013)



Entire Global Internet Average Data BW ~200 Tbps (source: CISCO)

~1500 nodes compute & storage
Full Bisection Multi-Rail Optical Network
Injection 80GBps/Node
Bisection 220Terabps

8 zones, Total 5600 nodes,
Injection 1GBps/Node
Bisection 160Gigabps

Acceleration of EBD Processing (1)

- Large Capacity – Multi-Terabytes, Petabytes, Exabytes
- Kernel algorithms for discrete data – graph, sort, etc.
  - EBD Characteristics
    - Sparse and random data structure
    - Involve frequent and abundant data transfer

    } Implies low latency and high bandwidth access
  - EBD Solutions (research)
    - High capacity at low power: non-volatile memory, deep memory hierarchy
    - High bandwidth: fast on-package memory + memory hierarchy+ Supercomputer Network (>100Gbps injection, Petabits bisection) + **bandwidth reducing algorithms for EBD**
    - Low Latency
      - latency reduction => memory 3-D stacking, fast on-package memory + low latency network
      - Latency hiding => many core + many threading + **latency reducing algorithms for EBD**

    } Our research: define & invent EBD architecture + algorithm + system SW

Acceleration of EBD Processing (2)

- Classification algorithms – statistical modeling/optimization, Machine Learning
  - EBD Characteristics: iterative numerical optimization
    - Kernel may be sparse (e.g., SVM) or dense (e.g., Deep Learning)
    - Parallelism difficult due to massive sample size (10~100 billion images)
  - EBD Solutions (our research)
    - Approach: Employ traditional and new HPC/supercomputer parallelization and acceleration strategies
    - Sparse algorithms – high bandwidth processors (e.g., GPU) w/stacked memory and on-package memory + memory hierarchy + supercomputing network + **bandwidth reducing algorithms (sparse linear algebra)**
    - Dense algorithms – many-core high FLOPS processor (e.g., GPU) + **algorithmic advances for strong scaling**
    - High volume data – **utilize "burst buffer" technology (incl. Clouds)**

Limited showing today

Optimized Graph500 program (1) – Bandwidth Reducing Algorithm
Sparse Matrix Representation with Bitmap

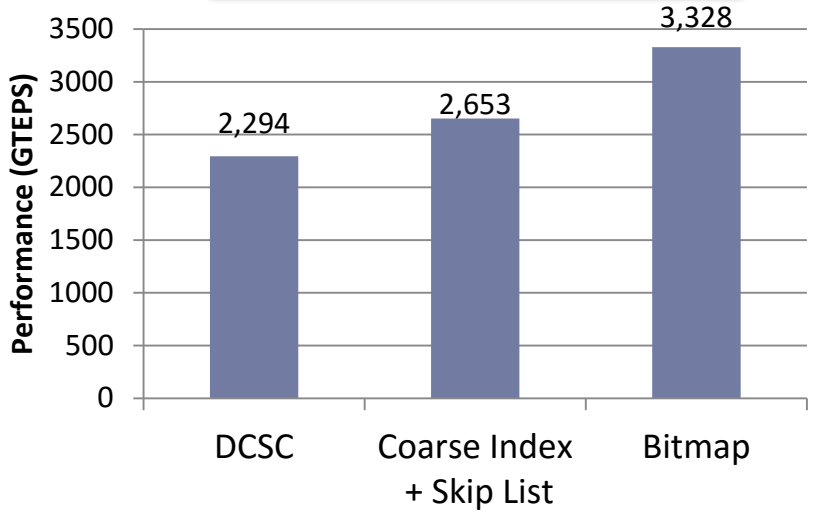
- Problem
  - Since the partitioned graph is a hyper sparse matrix, we need efficient hyper sparse matrix representation for large scale distributed graph processing.
- Our proposal: Sparse Matrix Representation with Bitmap
  - Enables **compression of row indexes** and **fast access to each row**.

Comparison with other methods

Data size of row index (MB/node)
(8064 partition, Scale 36)

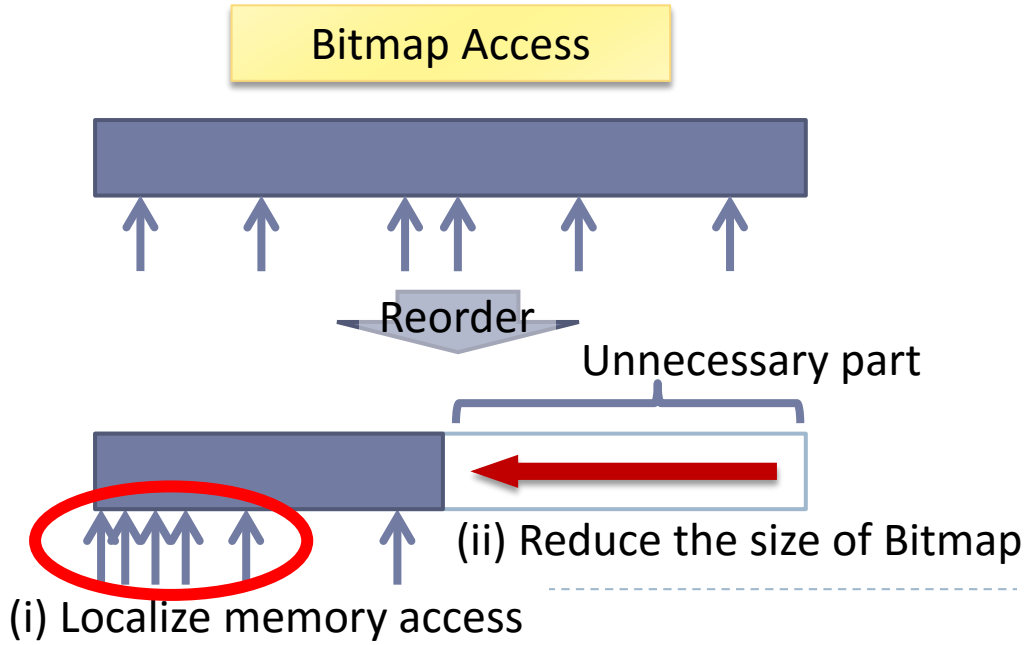
CSR (Compressed Sparse Row)	1806
DCSC	861
Coarse Index + Skip List	309
Bitmap (Proposal)	**337**

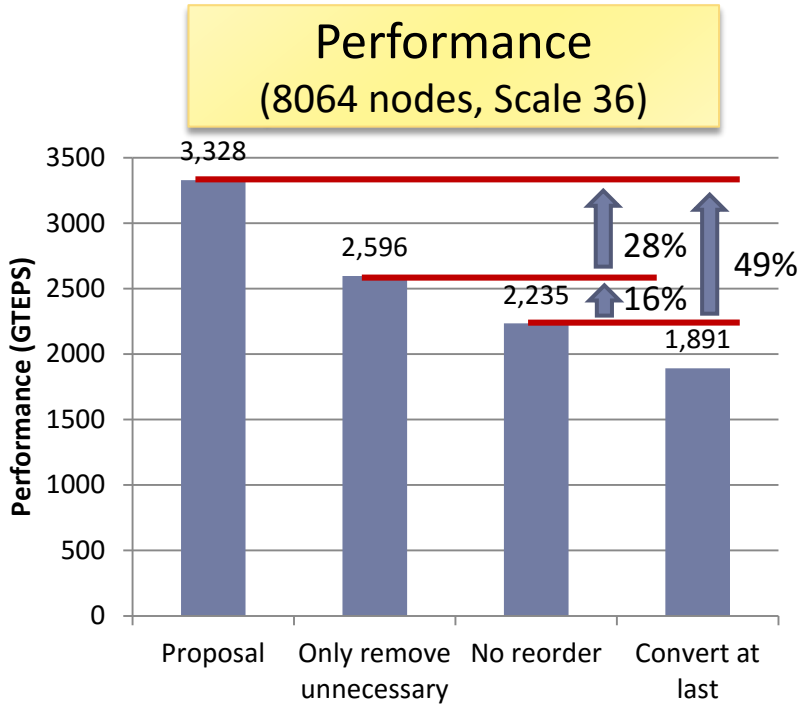
Performance
(8064 nodes, Scale 36)



Optimized Graph500 program (2) – Bandwidth Reducing Algorithm Vertex Reordering for Bitmap Optimization

- Our idea
  - Creates reordered vertex number by sorting vertices by degree.
  - Use reordered number for bitmap access and original number for other processing.
- Result
  - 16% speedup by reduction of bitmap data, 28% speedup by localized memory access, and 49% speedup in total. (8064 nodes)





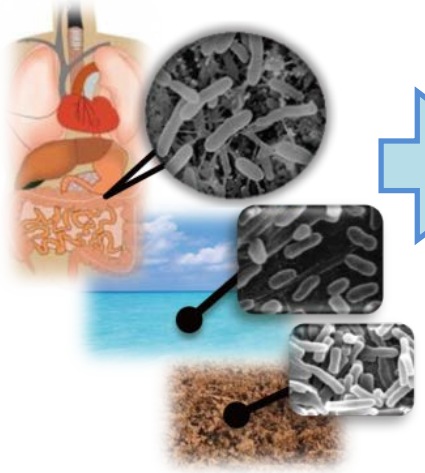
EBD App1: Ultra-fast and High-sensitive Homology Search for Metagenomics

[Akiyama Group]

GOAL

1. To develop ultra-fast and high-sensitive metagenomic analyzer.

2. To generalize the invented scheme, as a common tool for "EBD vs. EBD".



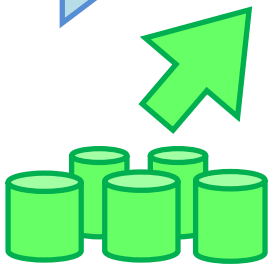
Environmental samples

...CCTTATCTTCG...

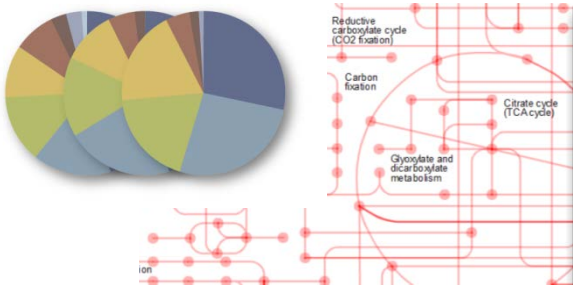
...CCACATAAACT...

...ATGGTCGATGTT...

DNA sequencing



Reference Databases

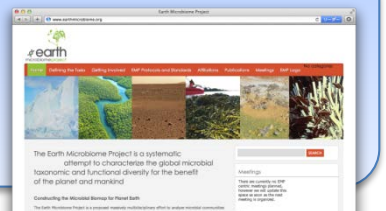


Phylogenetic composition
and
Metabolic pathway analysis

Hospital Microbiome

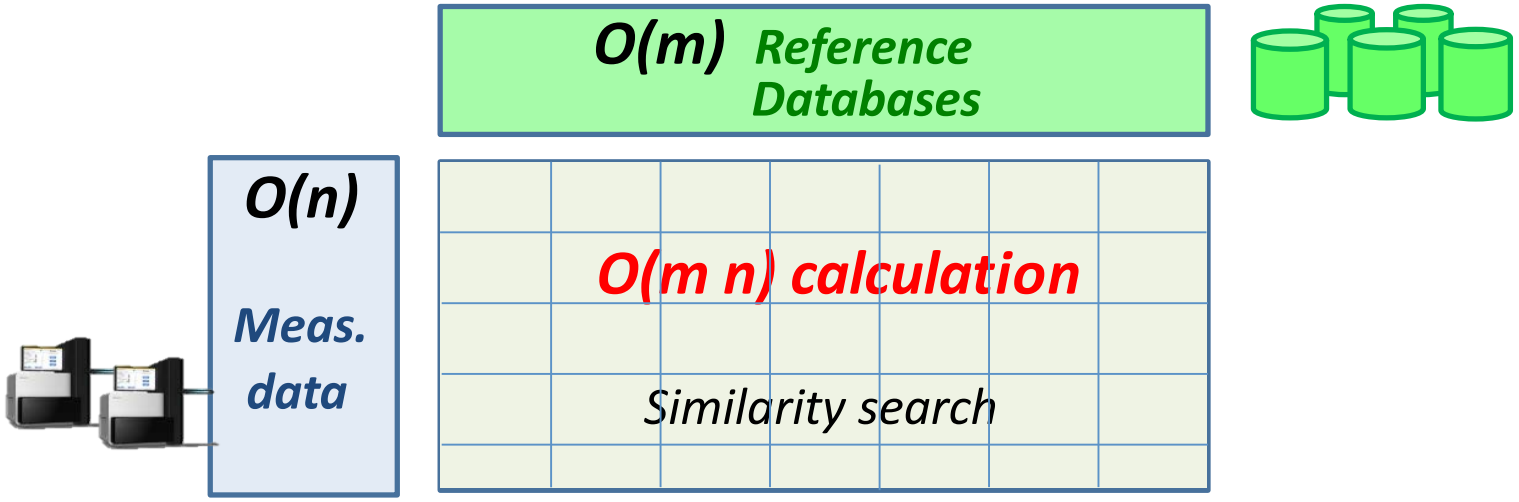
Home Microbiome

Earth Microbiome



High-sensitive Homology Search required in Metagenomics:

A typical situation where EBD vs. EBD calculation required.



In case of a human oral metagenome analysis:

Input:

Measured Data: 1,284 GB
(18 billion DNA reads)

A Reference DB: 15 GB
(KEGG GENES DB only)

Output:

Search Result: 1,223 GB
(Similarity, Ontology)

in 2009 on a Xeon cluster w/ 144 cores
0.18 M reads/hour
(100,000 hours / **18 billion reads**)

x3178 faster

in 2013 on K-computer
572M reads/hour
(32 hours / **18 billion reads**)

and beyond...

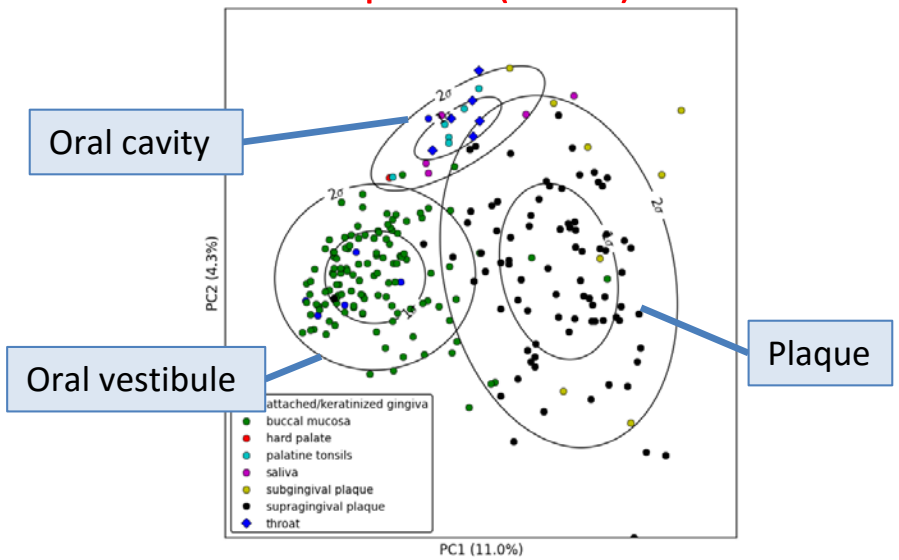
Results from Akiyama group@Tokyo Tech

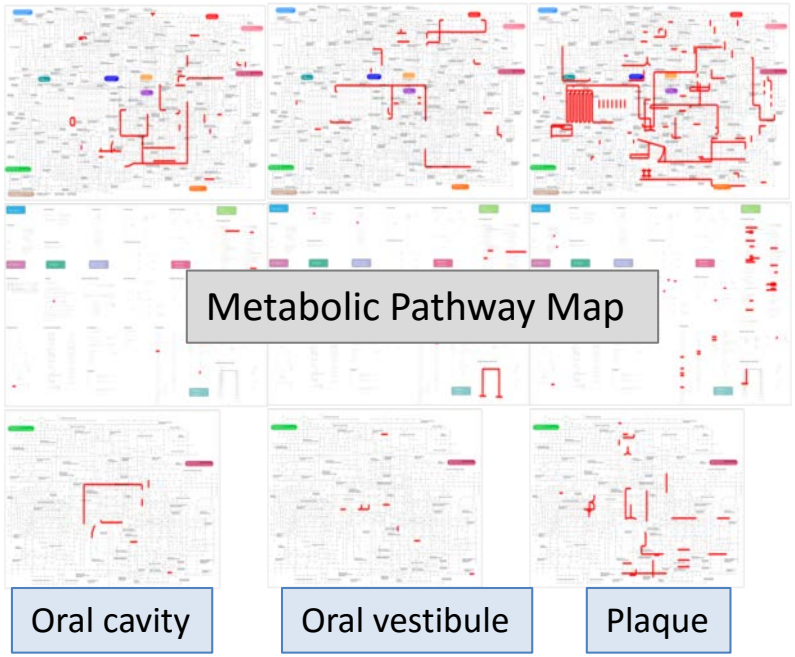
Ultra high-sensitive “big data” metagenome sequence analysis of human oral microbiome

- Required > **1 million node*hour product** on K-computer
- World’s most sensitive sequence analysis (based on amino acid similarity matrix)

- Discovered at least three microbiome clusters with <u>functional</u> differences.
  (Integrated 422 experiment samples taken from 9 different oral parts)

572.8 M Reads / hour
82,944 node (663,552 Cores)
K-computer (2012)

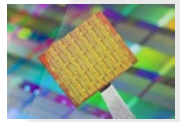


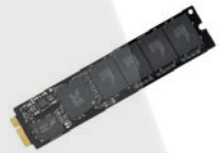


EBD Programming Framework

Hamar (Highly Accelerated Map Reduce) [Shirahata, Sato, … IEEE CCGrid 2013]

- A software framework for large-scale supercomputers w/ many-core accelerators and local NVM devices
  - Abstraction for deepening memory hierarchy
    - Device memory on GPUs, DRAM, Flash devices, etc.
- Features
  - Object-oriented
    - C++-based implementation
    - Easy adaptation to modern commodity many-core accelerator/Flash devices w/ SDKs
      - CUDA, OpenNVM, etc.
  - Weak-scaling over 1000 GPUs
    - TSUBAME2
  - Out-of-core GPU data management
    - Optimized data streaming between device/host memory
    - GPU-based external sorting
  - Optimized data formats for many-core accelerators
    - Similar to JDS format





EBD Programming Framework

Out-of-core GPU-MapReduce for Large-scale Graph Processing [IEEE Cluster 2014]

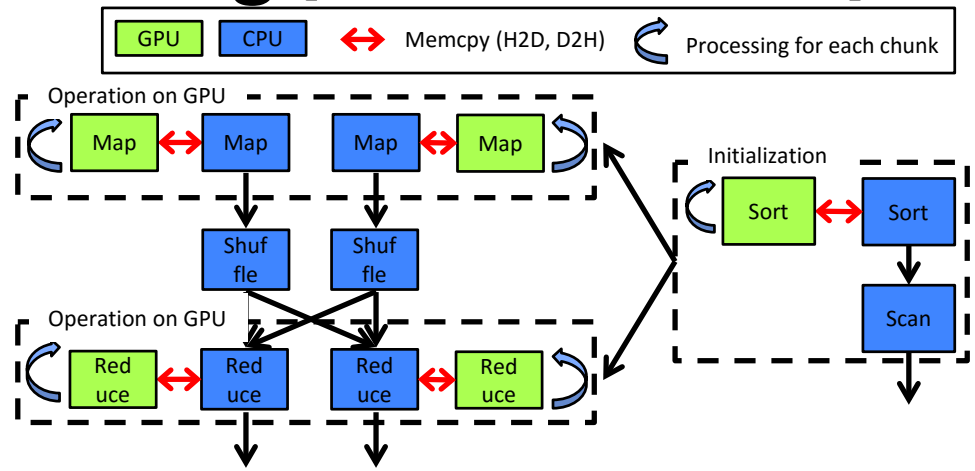
Emergence of large-scale graphs
- SNS, road network, smart grid, etc.
- Millions to trillions of vertices/edges

→ Need for fast graph processing on supercomputers

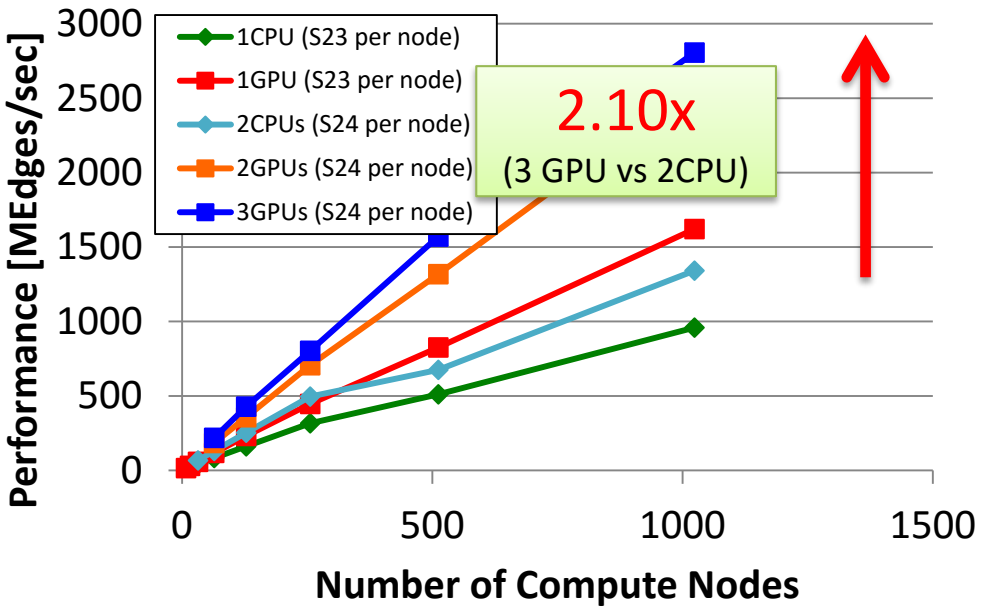
Problem: GPU memory capacity limits scalable large-scale graph processing

Proposal: Out-of-core GPU memory management on MapReduce
- Stream-based GPU MapReduce
- Out-of-core GPU sorting

Experimental Results:
performance improvement over CPUs
- Map: 1.41x, Reduce: 1.49x, Sort: 4.95x speedup
- Overlapping communication effectively

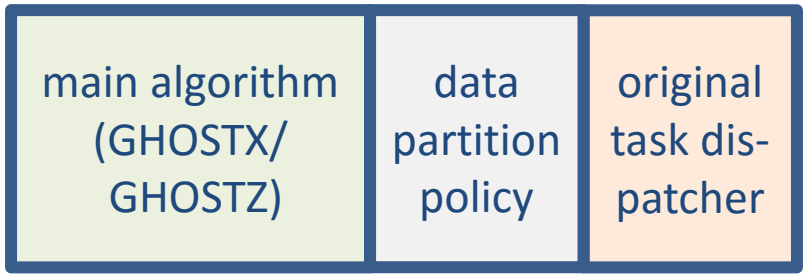


Weak scaling on TSUBAME2.5

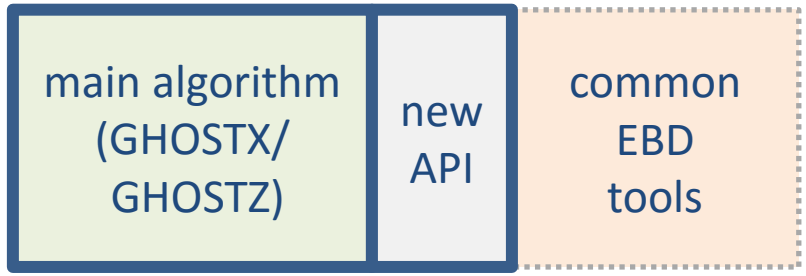


Toward Co-design: Generalization of scheme

Current GHOST-MP structure

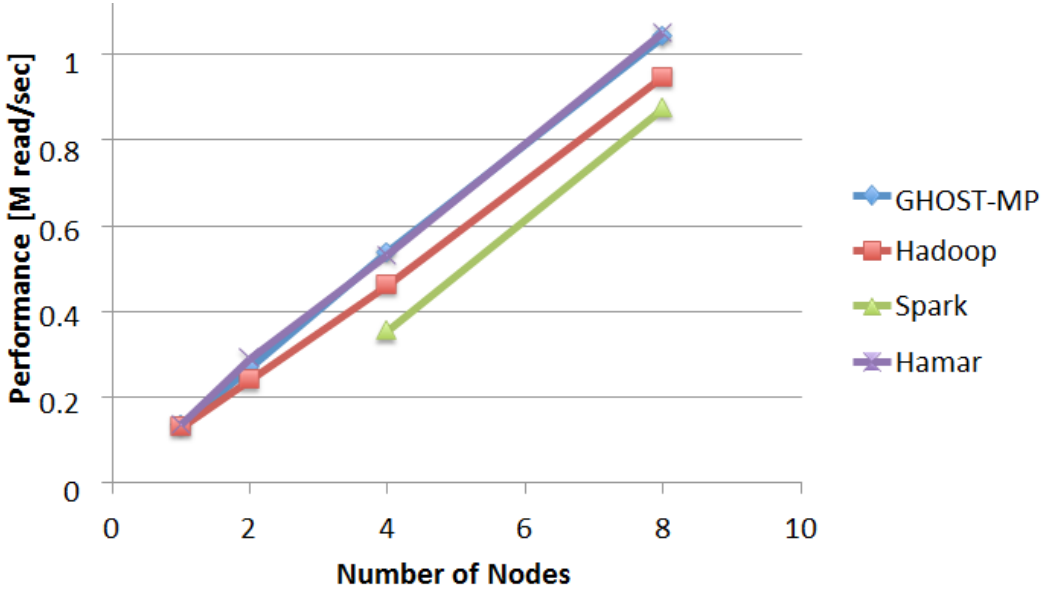


Future GHOST-MP structure



MapReduce implementations of GHOST-MP using Hadoop, Spark, and Hamar

[Zhang et al., HPCS2015]



[3] Zhang C, Shirahata K, Suzuki S, Akiyama Y, Matsuoka S, HPCS2015 (2015).

GPU-based Distributed Sorting [Shamoto, IEEE BigData 2014, IEEE Trans. Big Data 2015]

EBD Algorithm Kernels

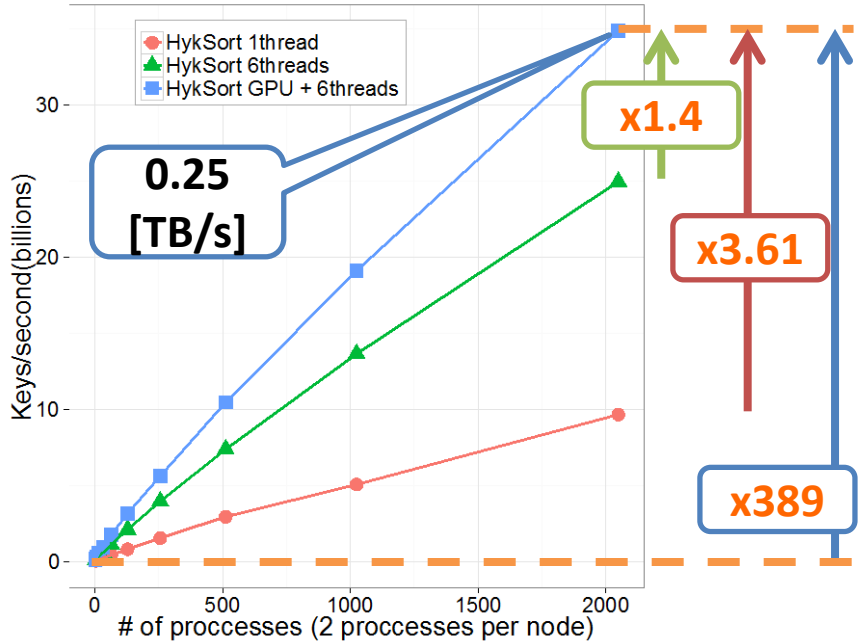
- Sorting: Kernel algorithm for various EBD processing
- Fast sorting methods
  - Distributed Sorting: Sorting for distributed system
    - Splitter-based parallel sort
    - Radix sort
    - Merge sort
  - Sorting on heterogeneous architectures
    - Many sorting algorithms are accelerated by many cores and high memory bandwidth.

- Sorting for large-scale heterogeneous systems remains unclear

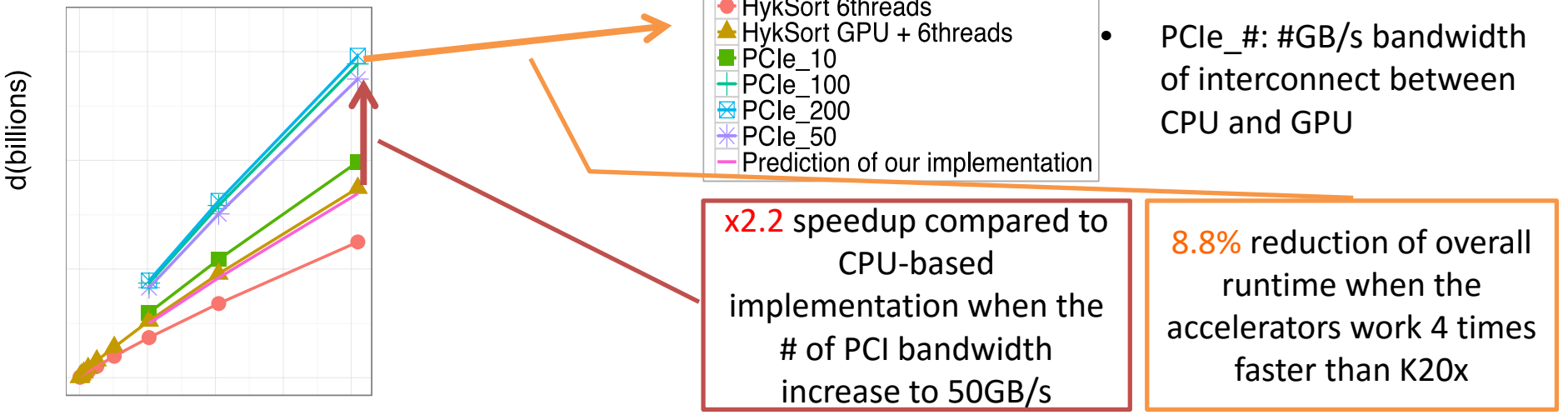
- We develop and evaluate bandwidth and latency reducing GPU-based HykSort on TSUBAME2.5 via latency hiding
  - Now preparing to release the sorting library

GPU implementation of splitter-based sorting (HykSort)

- Weak scaling performance (Grand Challenge on TSUBAME2.5)
  - 1 ~ 1024 nodes (2 ~ 2048 GPUs)
  - 2 processes per node
  - Each node has 2GB 64bit integer
- C.f. Yahoo/Hadoop Terasort: 0.02[TB/s]
  - Including I/O



Performance prediction



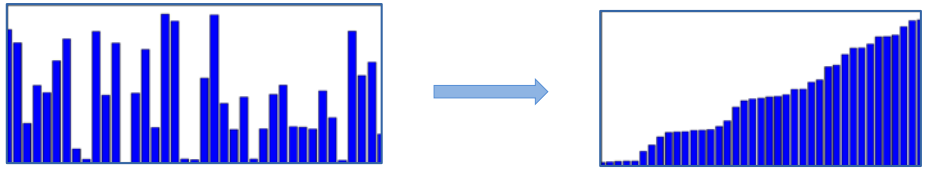
- PCIe_#: #GB/s bandwidth of interconnect between CPU and GPU

x2.2 speedup compared to CPU-based implementation when the # of PCI bandwidth increase to 50GB/s

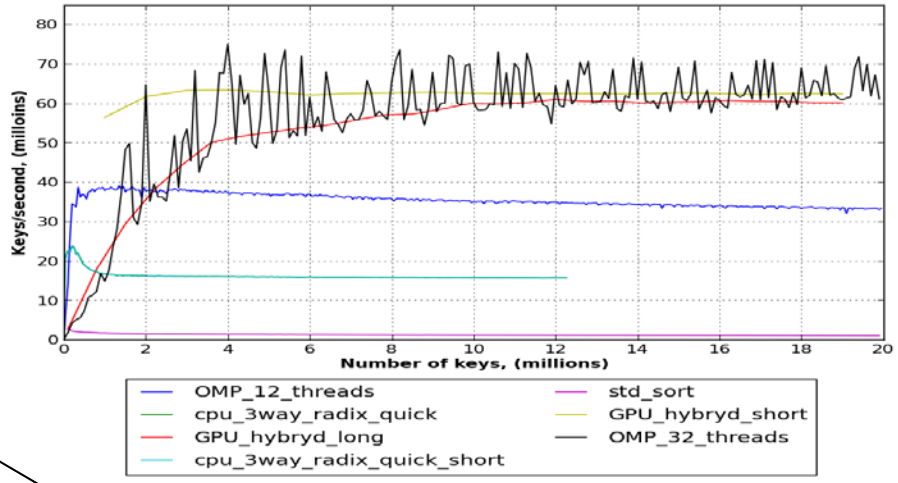
8.8% reduction of overall runtime when the accelerators work 4 times faster than K20x

EBD Algorithm Kernels

Efficient Parallel Sorting Algorithm for Variable-Length Keys



- Comparison-based sorts inefficient for long/variable-length keys (like strings)
- Better way: examining individual characters (based on MSD Radix sort algorithm)
- Hybrid parallelization scheme: combining data-parallel and task-parallel stages



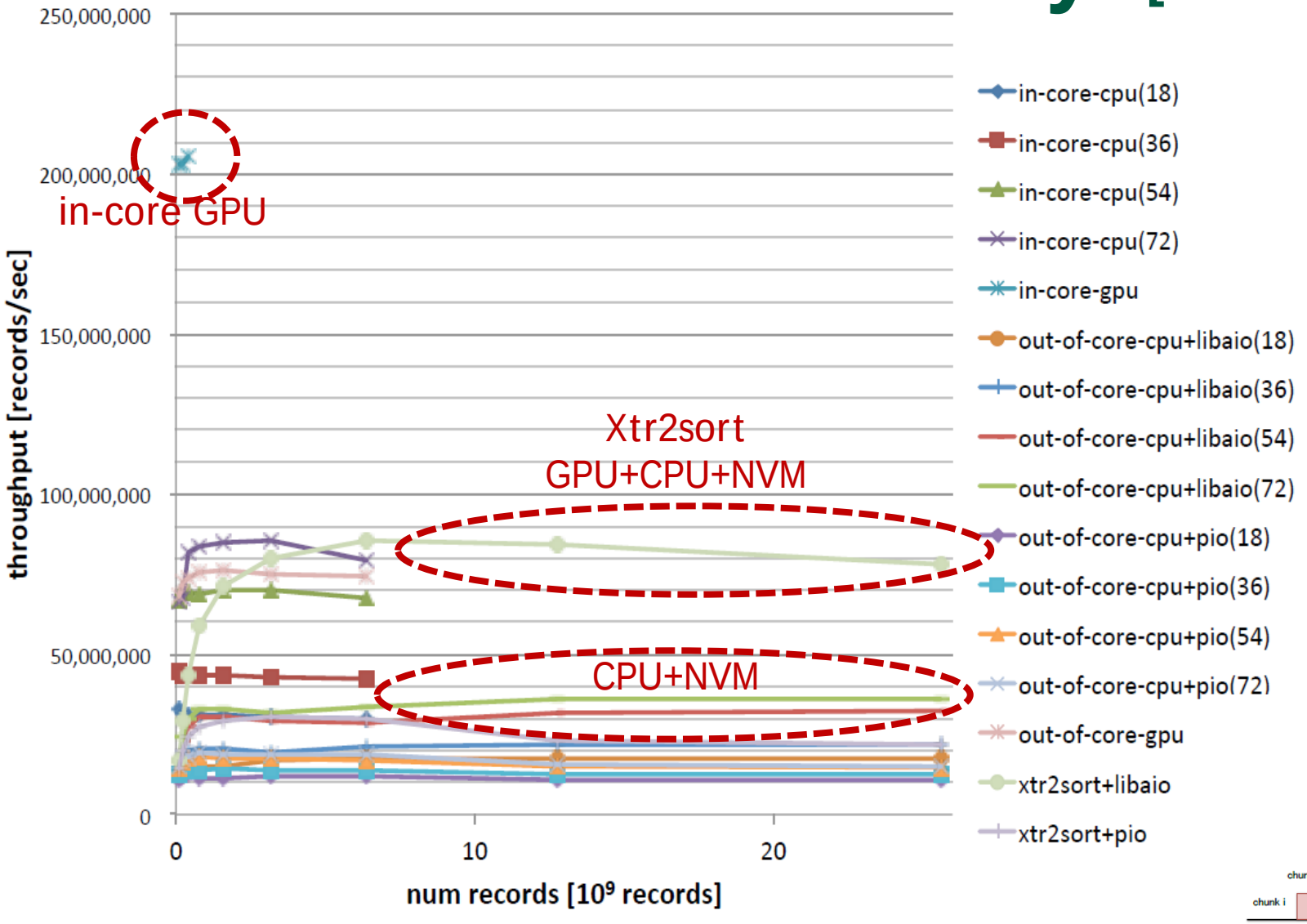
apple
apricot
banana
kiwi

70 M keys/second sorting throughput on 100bytes strings

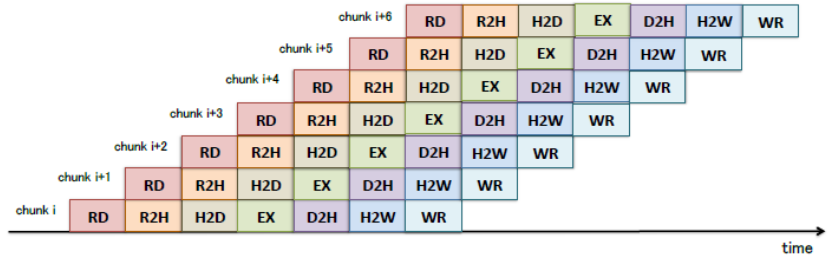
Aleksandr Drozd, Miquel Pericàs, Satoshi Matsuoka. Efficient String Sorting on Multi- and Many-Core Architectures. *in Proceedings of IEEE 3rd International Congress on Big Data. Anchorage, USA, August 2014*

Aleksandr Drozd, Miquel Pericàs, Satoshi Matsuoka. MSD Radix String Sort on GPU: Longer Keys, Shorter Alphabets *in proceedings of 第142回ハイパフォーマンスコンピューティング合同研究発表会 (HOKKE-21)*

GPU + NVM + PCIe SSD Sorting our new Xtr2sort library [H.Sato et.al. SC15 Poster]



Single Node Xeon
- 2 socket 36 cores
- 128GB DDR4
- K40 GPU (12GB)
- SSD PCIe card (2.4TB)



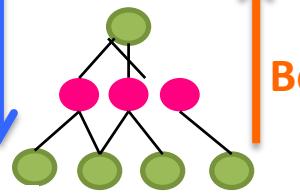
Large Scale Graph Processing Using NVM [Iwabuchi, IEEE BigData2014]

1. Hybrid-BFS (Beamer'11)

EBD Algorithm Kernels

Switching two approaches

Top-down ↓ ↑ Bottom-up



$$n_{frontier} < \frac{n_{all}}{\beta} \qquad n_{frontier} > \frac{n_{all}}{\alpha}$$

# of frontiers:$n_{frontier}$, # of all vertices:n_{all}, Parameter : α, β

2. Proposal

DRAM

Holds highly accessed data

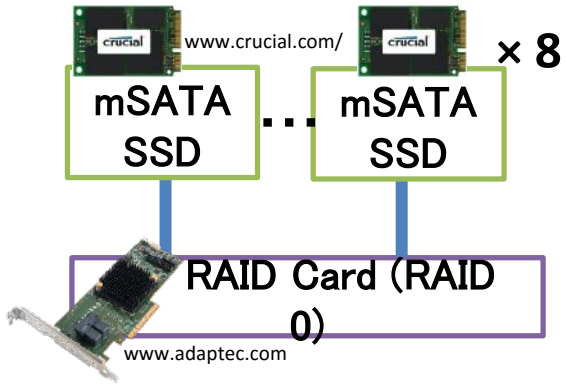
NVM

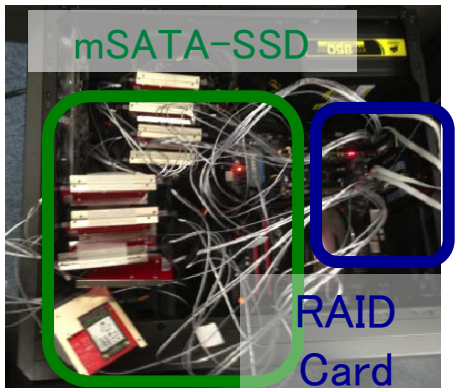
Holds full size of Graph

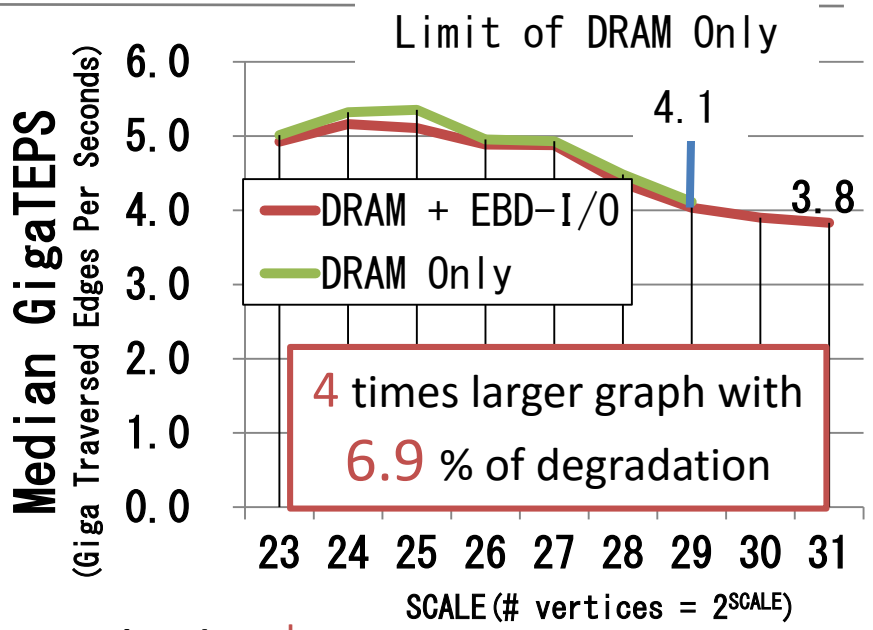
Load highly accessed graph data before BFS

3. Experiment

CPU	Intel Xeon E5-2690 × 2
DRAM	256 GB
NVM	EBD-I/O 2TB × 2





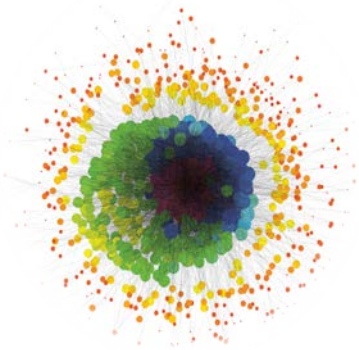


Ranked 3rd in **Green Graph500 (June 2014)**

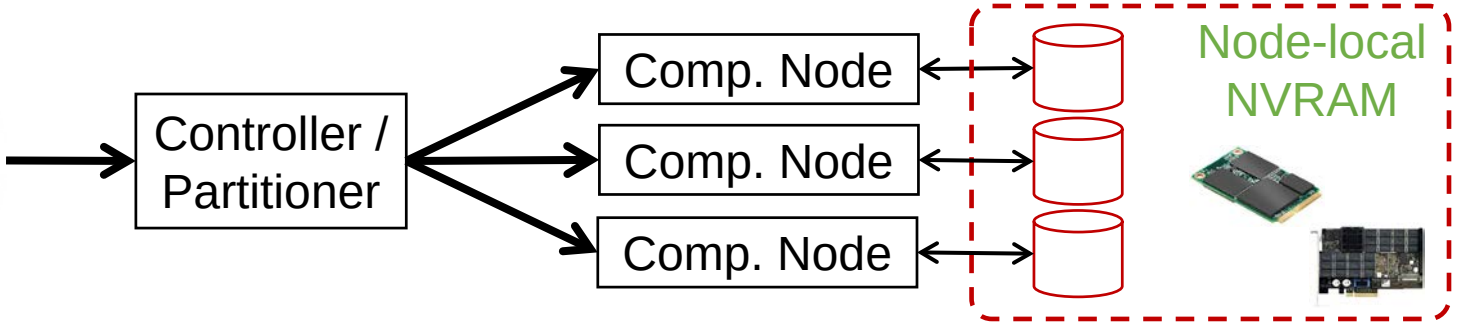
Dynamic Graph Data Structure Using Local-NVRAM

[Iwabuchi et al. SC15 Poster]

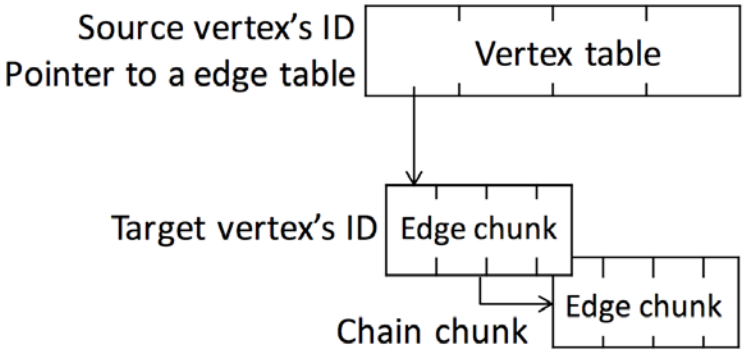
Dynamic edges and vertices insert and delete



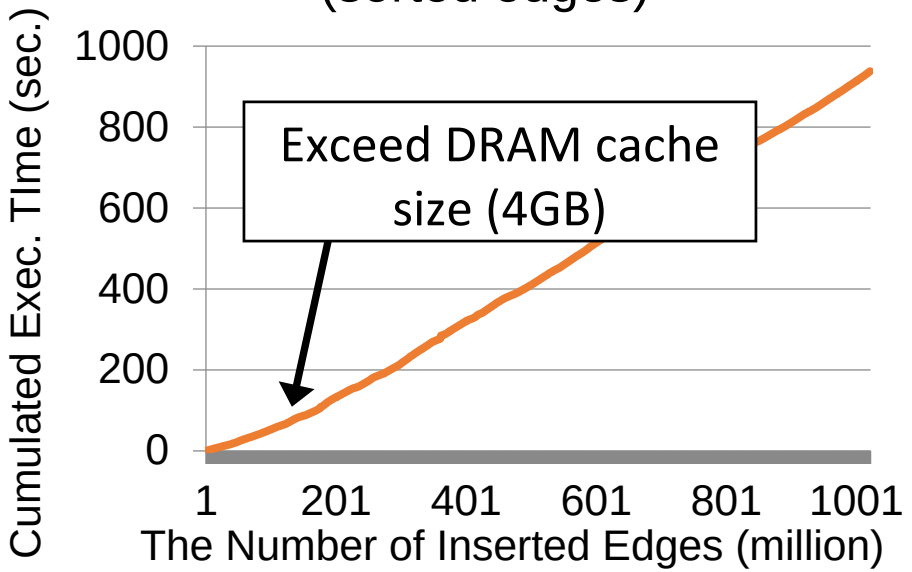
The dynamic graph data structure
(This work)



Increase page-level locality of data stored in NVRAM by carefully choosing a hash function



Edge Insert Performance
(sorted edges)



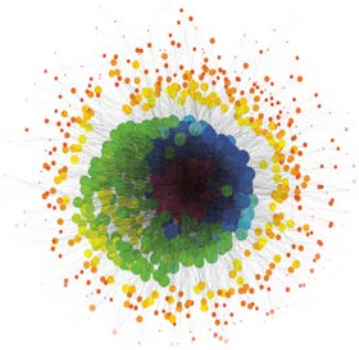
TOKYO TECH

Lawrence Livermore National Laboratory

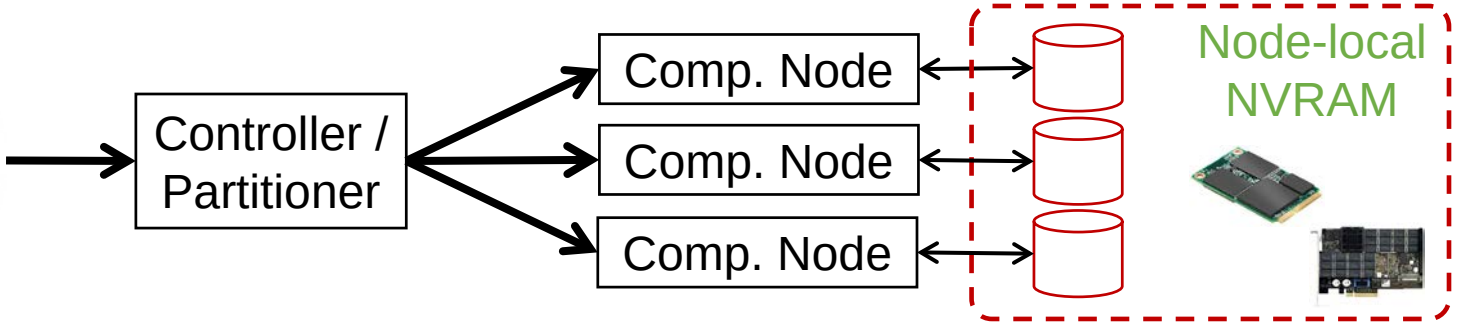
Dynamic Graph Data Structure Using Local-NVRAM

[Iwabuchi et al. SC15 Poster]

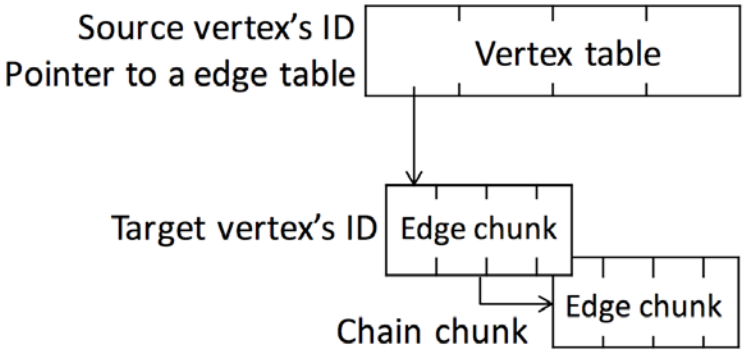
Dynamic edges and vertices insert and delete



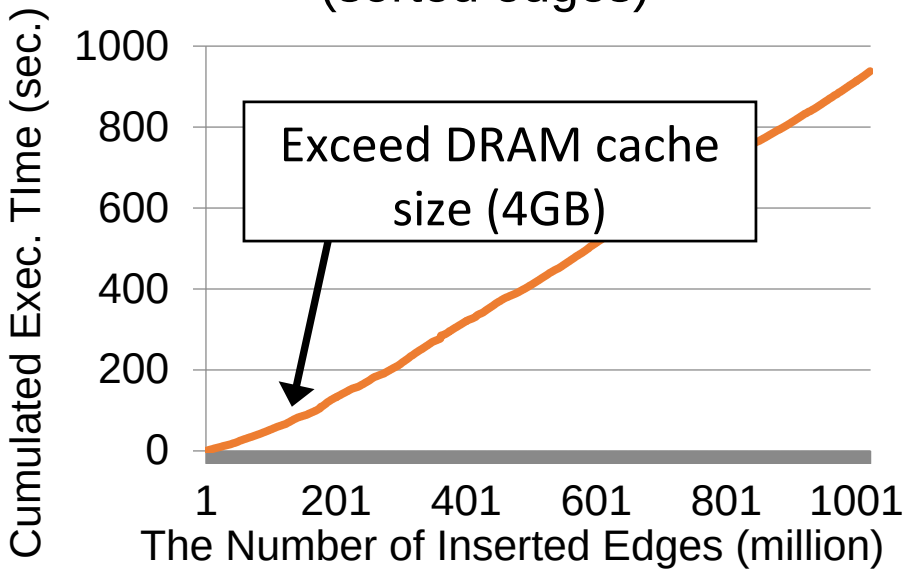
The dynamic graph data structure
(This work)



Increase page-level locality of data stored in NVRAM by carefully choosing a hash function



Edge Insert Performance
(sorted edges)

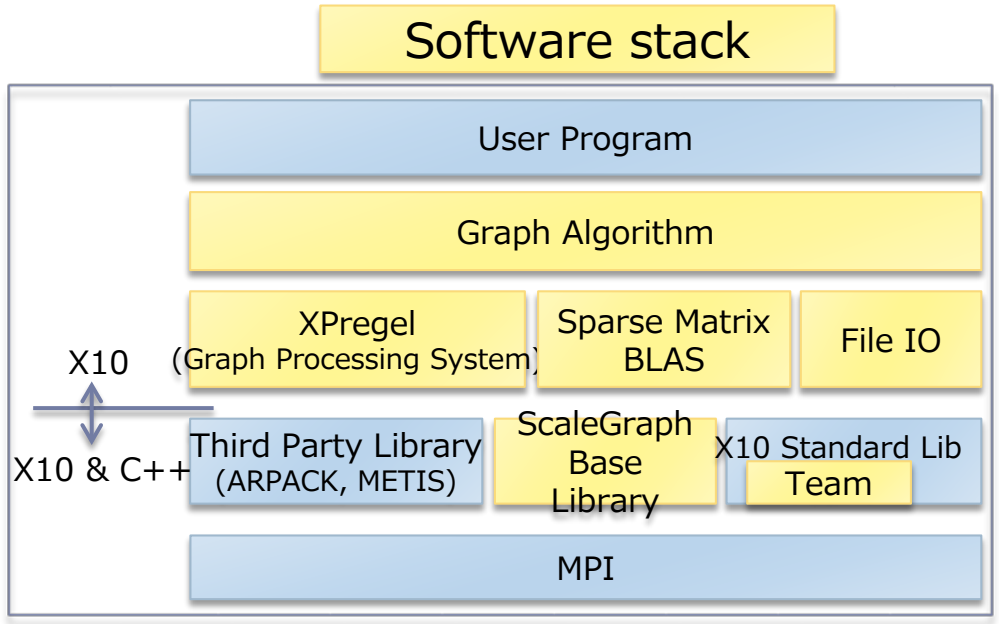


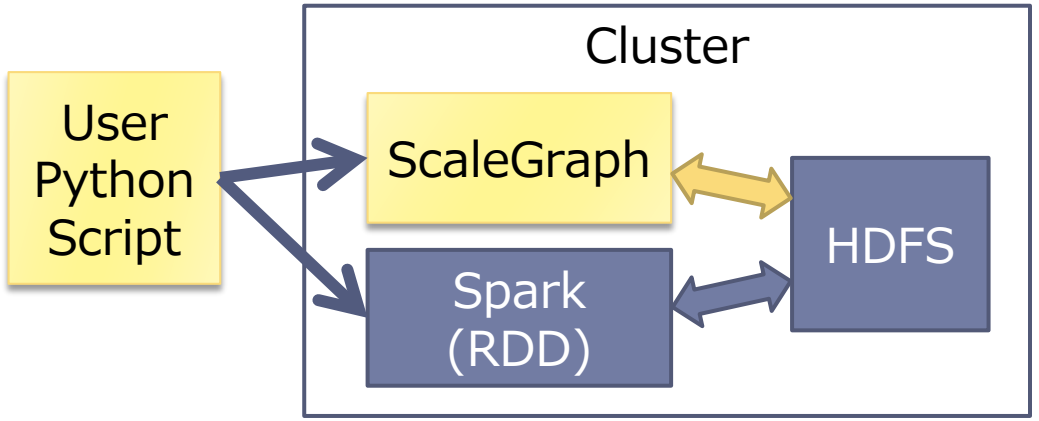
TOKYO TECH

Lawrence Livermore National Laboratory

ScaleGraph Large-scale Graph Processing Framework enhanced w/ User-Friendly Python / Spark Interface

- ScaleGraph [Suzumura]
  - X10-based open source **Highly Scalable Large Scale Graph Analytics Library** beyond the scale of billions of vertices and edges on Distributed Systems
    - **XPregel**: Pregel-based bulk synchronous parallel graph processing framework
    - Built-in graph algorithms (Centrality, Connected Component, Clustering, etc.)
- NEW Development: Python Interface
  - Allow users to use ScaleGraph with Spark* by easy python interface





*Apache Spark: http://spark.apache.org/

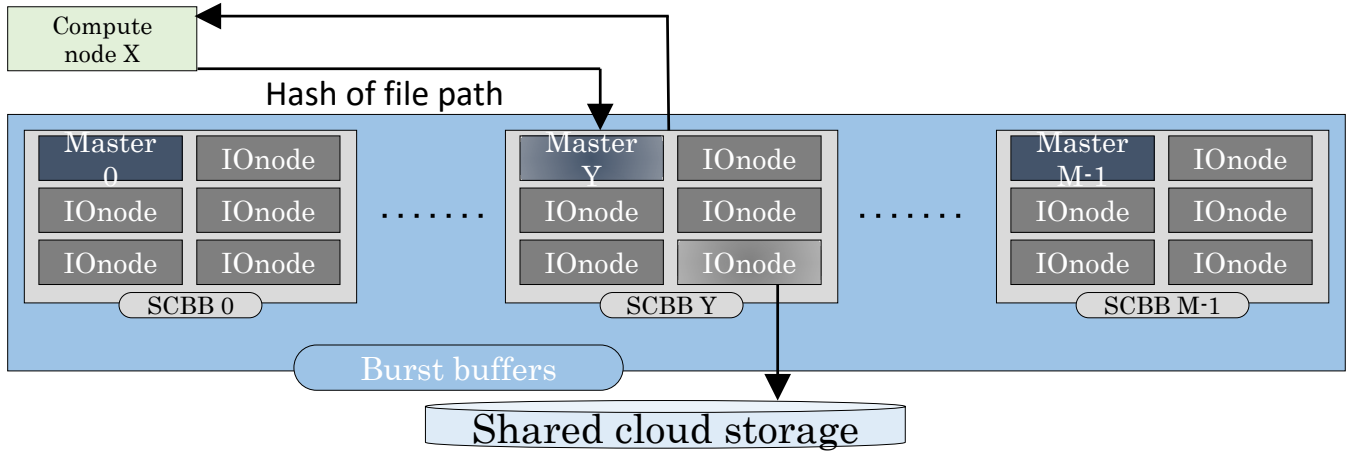
Cloud-based I/O Burst Buffer Architecture

[Xu et al. SC15 Poster]

In collaboration talks w/ AWS

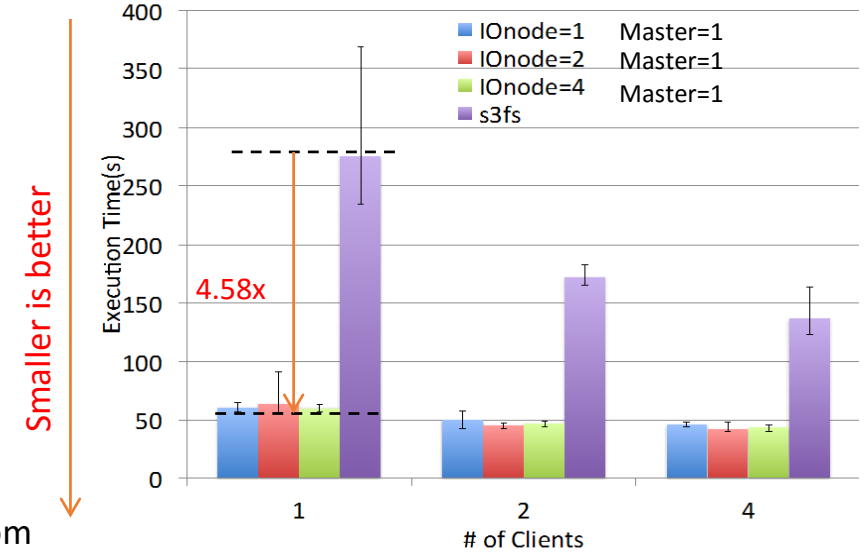
- Cloud-based I/O Burst Buffers
  - Using several compute nodes in a public cloud as I/O buffer nodes
  - Taking advantage of high throughput of LAN inside a public cloud

System Architecture



- The whole system consists of several SCBB (Sub CloudBB)
- Each SCBB consists of a Master and several IOnodes.
  - Masters: controlling all IOnodes in the same SCBBs and handling all I/O requests from Compute Nodes.
  - IOnodes: storing actual data and transferring data with Compute Nodes.

Application Case Study

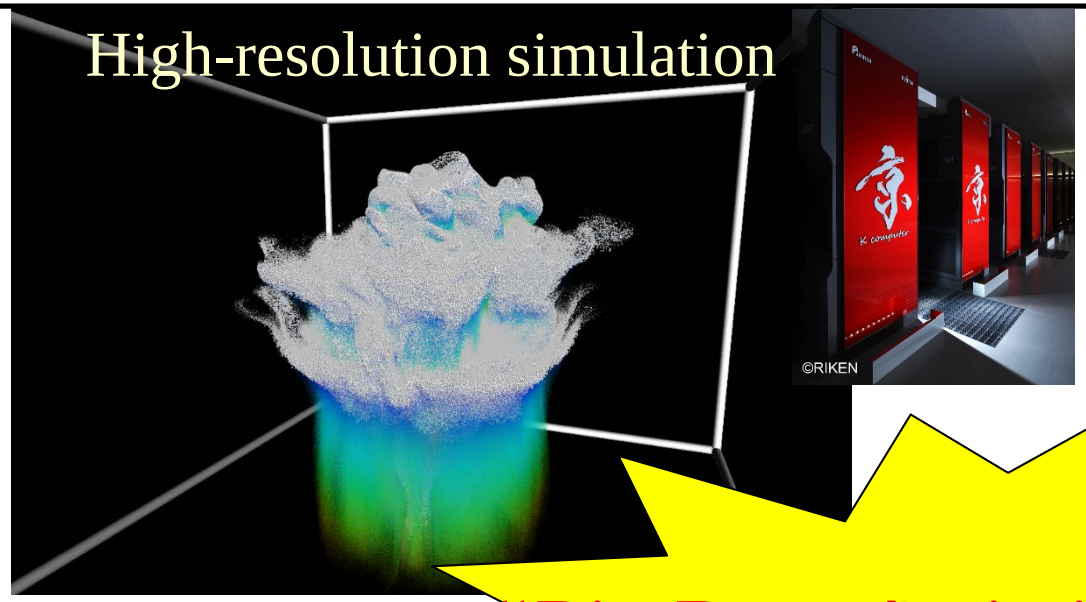


Montage: Improve Performance

EDB Distributed Object Store (co-PI: Osamu Tatebe, U-Tsukuba)

- Background
  - Demand for high IOPS and high BW of object store by EDB applications
  - Innovation of storage device and storage interface
- Objective - design and implementation for EDB Distributed Object Store
  - Design of object store in proposed standard storage interface [GPC 2015]
  - Design of concurrent B+tree index for NVM Key Value
  - Ongoing: Indexed cyclo-join for nearest neighbor search
  - Ongoing: Co-design of execution model with meta-genomics application

“Big Data Assimilation” in Weather

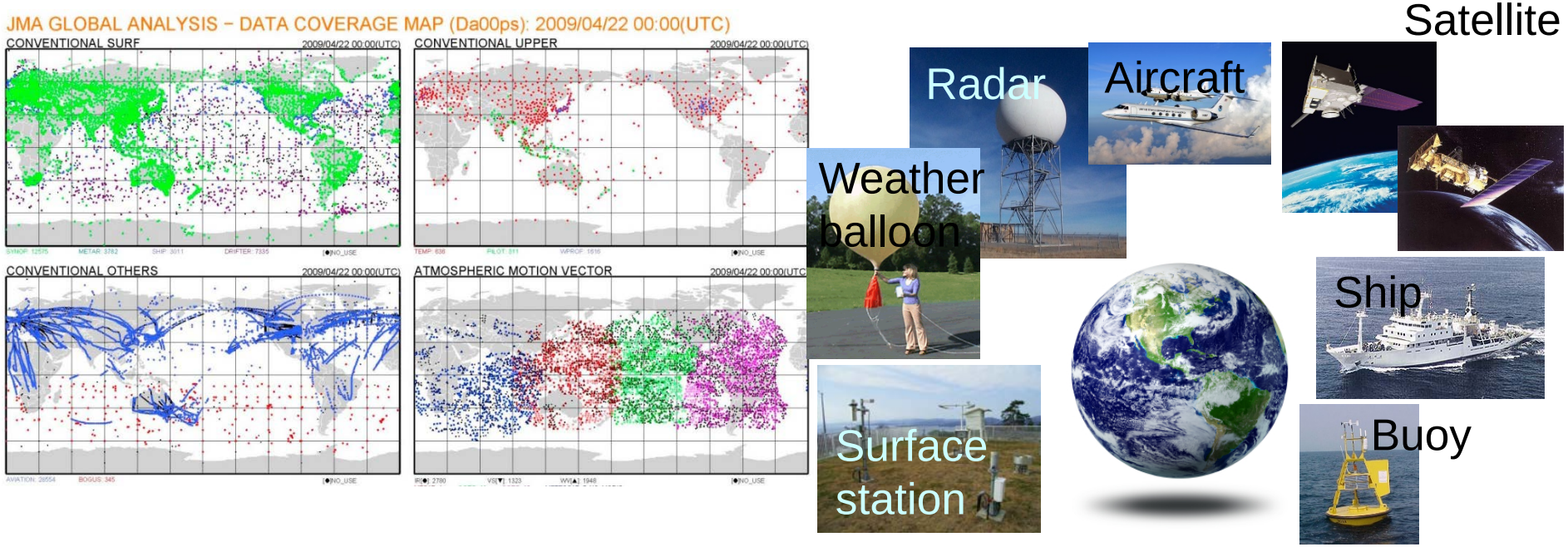


Combination of
next-generation technologies

“Big Data Assimilation”

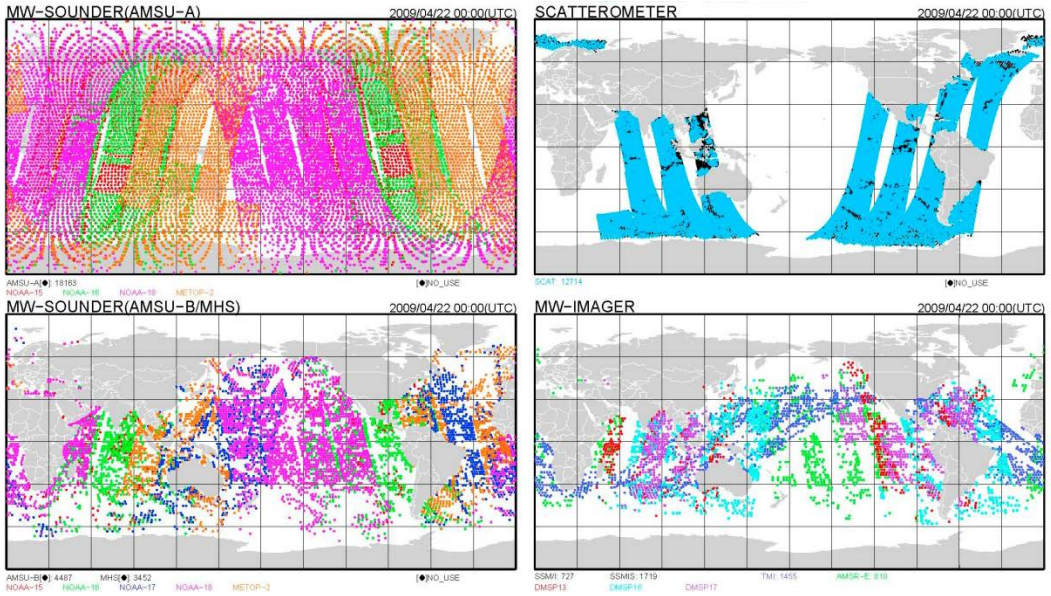
Improving simulations

Collecting Atmospheric Data



Global continuous collection

Variety of sensors, stationary and mobile

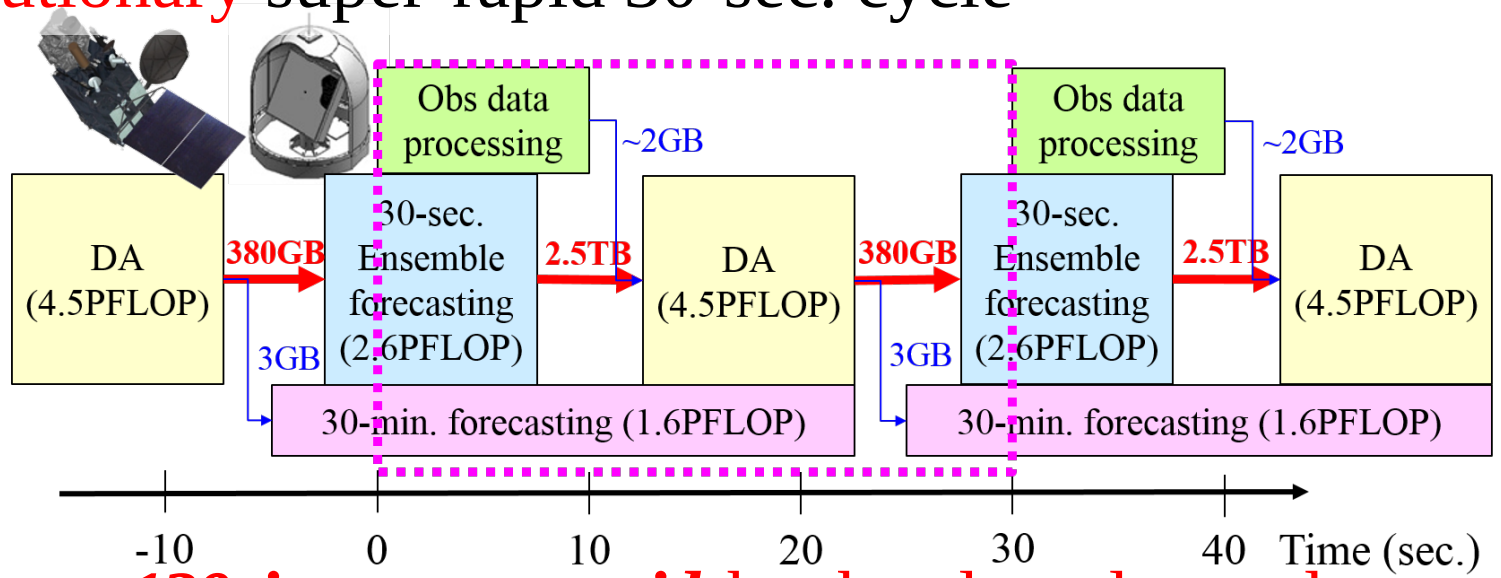


EBD App2: Miyoshi Group (Weather Forecast Application) Requirements – Data Assimilation in Practice

Big Data Assimilation
for severe weather forecast

Goal：Pinpoint (100-m resol.) forecast of severe local weather by updating 30-min forecast every 30 sec!

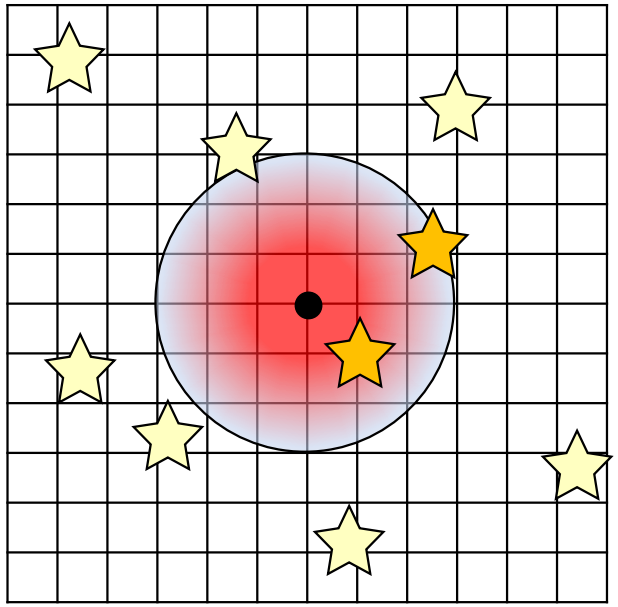
Revolutionary super-rapid 30-sec. cycle



120 times more rapid than hourly update cycles

Another requirement (with Tatebe Group)

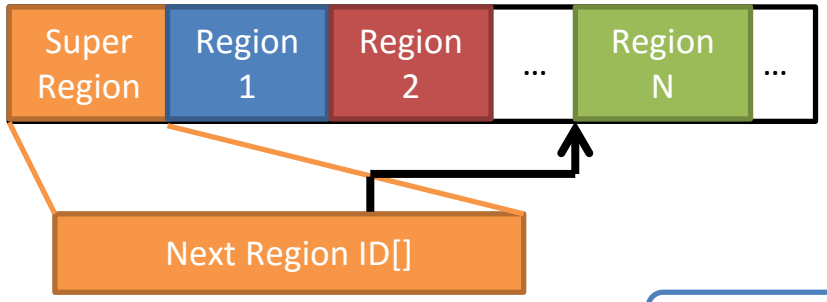
- Geographical Search for LETKF
  - Searching local O(10^3) from global O(10^6) at every grid point
  - The number of grid points = O(10^8)
  - This part takes ~50 % of the total computer time.
- R-tree or kd-tree will be applied to accelerate the search.
  - In collaboration with Tatebe Group

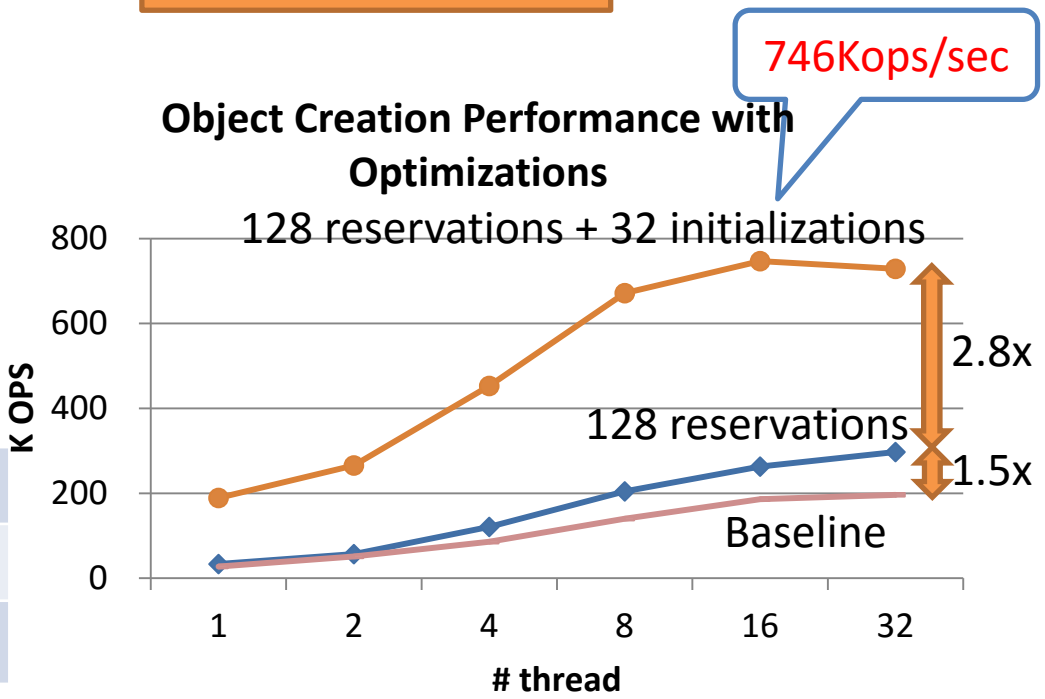


Object Storage Design in OpenNVM [Takatsu et al GPC 2015]

- New interface - Sparse address space, atomic batch operations and persistent trim
- Simple design by fixed-size Region enabled by sparse address space and persistent trim
  - Free'ed by persistent trim and no reuse
  - Enough region size to store one object
- Optimization techniques for object creation
  - Bulk reservation and bulk initialization

XFS	15.6 Kops/s
DirectFS	61.3 Kops/s
Proposal	746 Kops/s



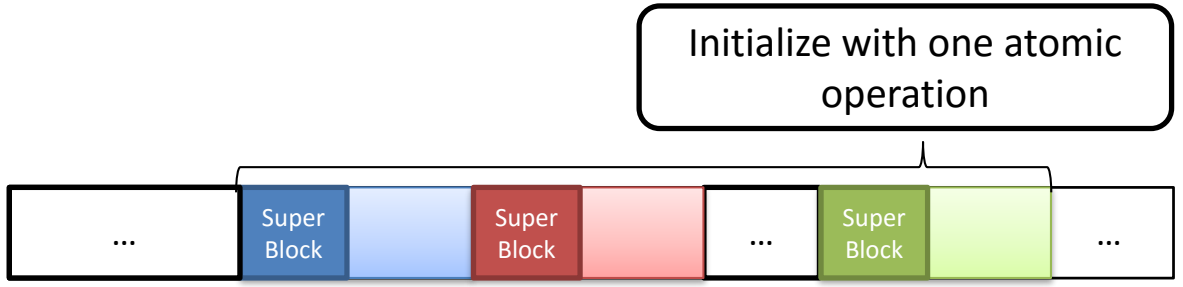


Fuyumasa Takatsu, Kohei Hiraga, and Osamu Tatebe, "Design of object storage using OpenNVM for high-performance distributed file system", the 10th International Conference on Green, Pervasive and Cloud Computing (GPC 2015), May 4, 2015

Optimizations for Object Creation

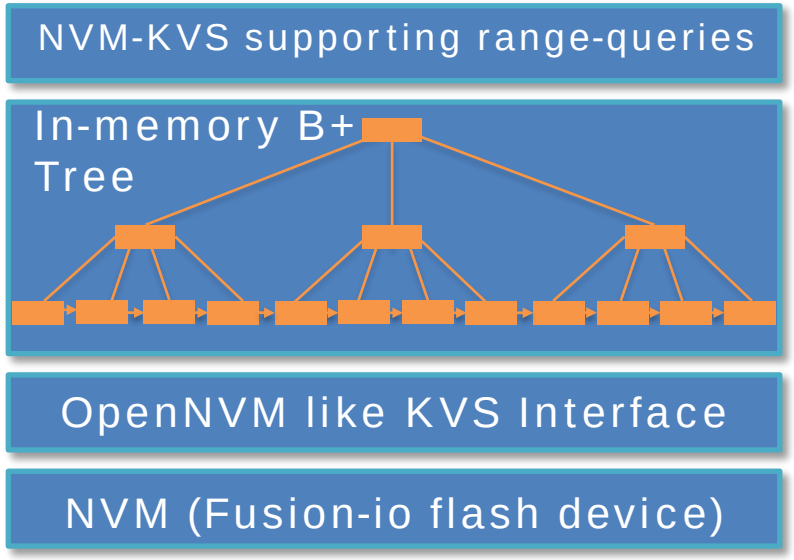
- Bulk Reservation
  - Super region keeps the position of the next region
  - Bulk reservation commits the position every N times
  - When failure happens, only logical sparse space is lost
    - Physical space is not affected

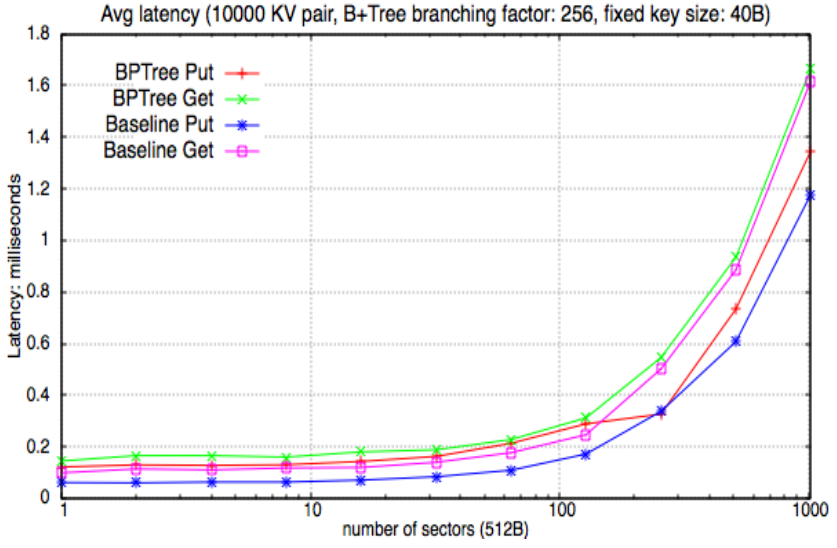
- Bulk Initialization
  - Super block in each region will be initialized when it is used
  - OpenNVM atomic batch write performs better when the number of vectors increases
  - Bulk Initialization initializes N regions with a single batch write operation
  - When failure happens, lose non-used initialized super blocks



Concurrent B+Tree Index for Native NVM-KVS [Jabri]

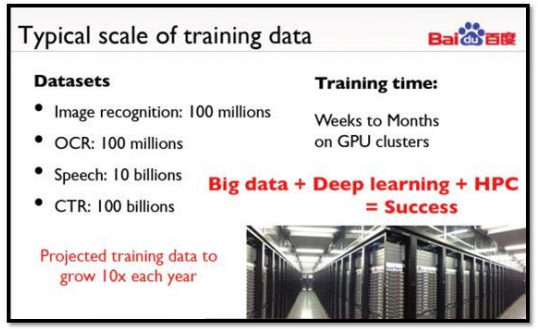
- Enable range-queries support for KVS running natively on NVM like fusionio ioDrive
- Design of Lock-free concurrent B+Tree
  - Lock-free operations – search, insert and delete
    - Dynamic rebalancing of the Tree
    - Nodes to be split or merged are frozen until replaced by new nodes
- Asynchronous interface using future/promise in C++11/14

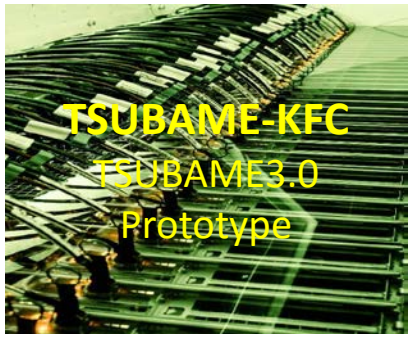




NEW App 2015: EBD Co-Design for Deep Learning Applications

- Deep Learning IS HPC!
  - Training models – mostly dense MatVec
  - Data Access for training target data sets
  - Sharing updated training parameters in neural networks
- Goals
  - Accelerate DL applications in EBD architectures ?
    - Extreme-scale Parallelization, Fast Interconnects, Storage I/O, etc.
  - Performance bottlenecks of multi-node parallel DL algorithms on current HPC systems ?
- Current Status
  - Official Collaboration w/DENSO IT Lab to be signed October
  - Profiling based bottleneck identification and performance optimization of a real DL application on TSUBAME
  - > 100 million images, 1500 GPUs (6 Pflops) 1 week grand challenge run
  - Compete w/Google, MS, Baidu etc. in ILSVRC in ImageNet

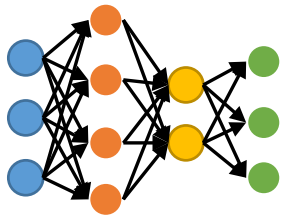


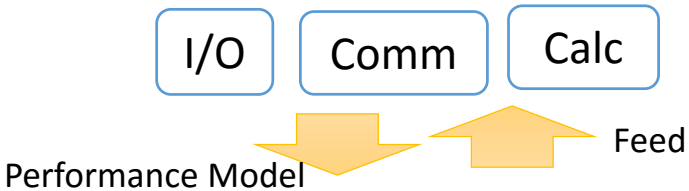


Many companies (ex. Baidu, etc.) employ GPU-based Cluster Architectures, similar to TSUBAME2 & KFC

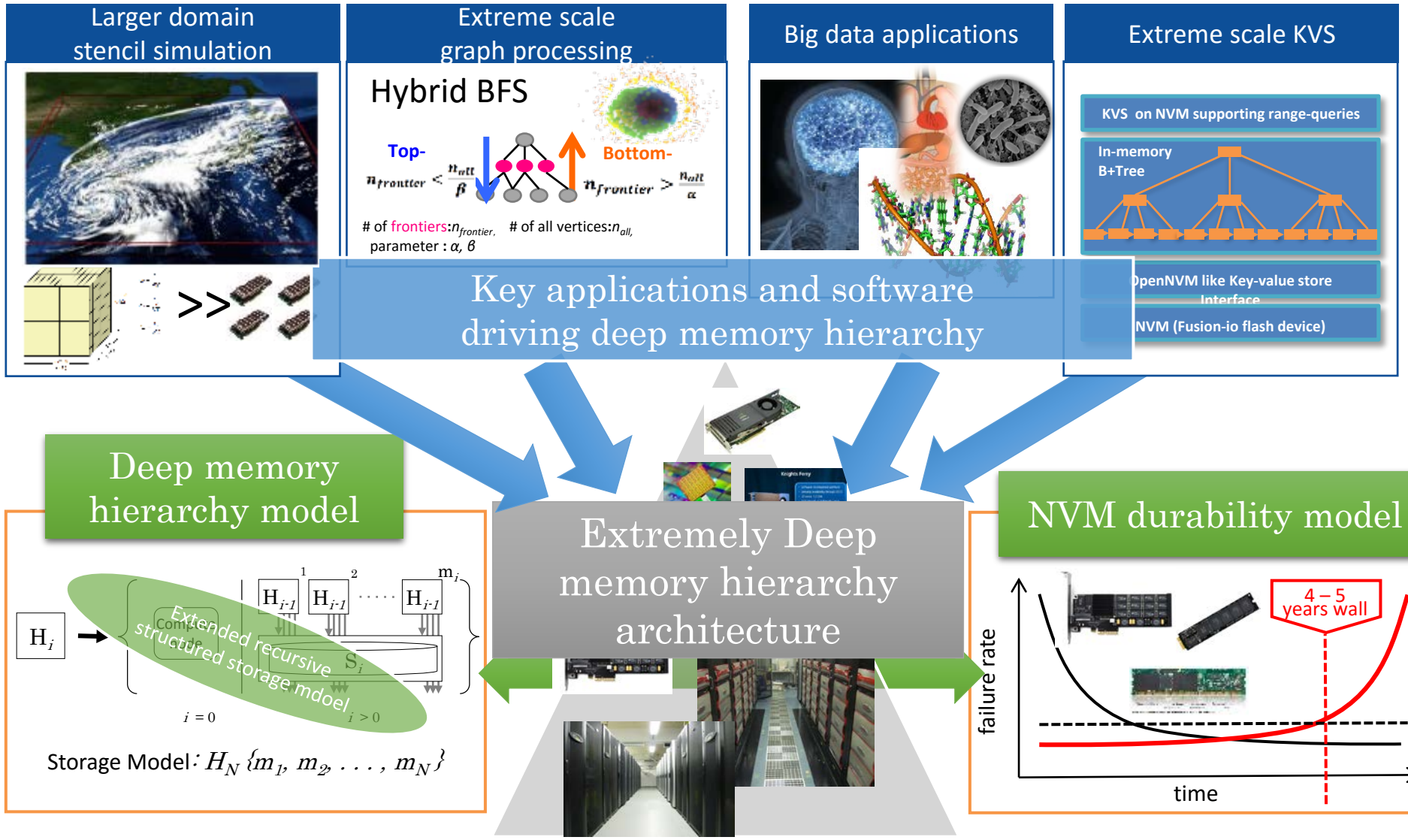
DENSO

Real DL Applications





Convergence of HPC and Big Data – Research Frontiers



EBD Open Source Software Development

- Early research deliverable outcome
- Distributed as OSS & facilitated on TSUBAME2/3 for CREST-BD
- GPU-based Large-scale Distributed Sorting
  - Scales over 1000 nodes
  - STL-like Interfaces
- ScaleGraph: Large-scale Graph Processing Framework enhanced w/ User-Friendly Python / Spark Interface
  - Calling Fast Graph Algorithm Implementations via Python
  - Easy Pregel-style Graph Programing using Python
- Cloud-based I/O Burst Buffers
  - Fuse-based Filesystem for general purpose I/O operations

Towards TSUBAME4 and 5: Moore's Law will end in the 2020's

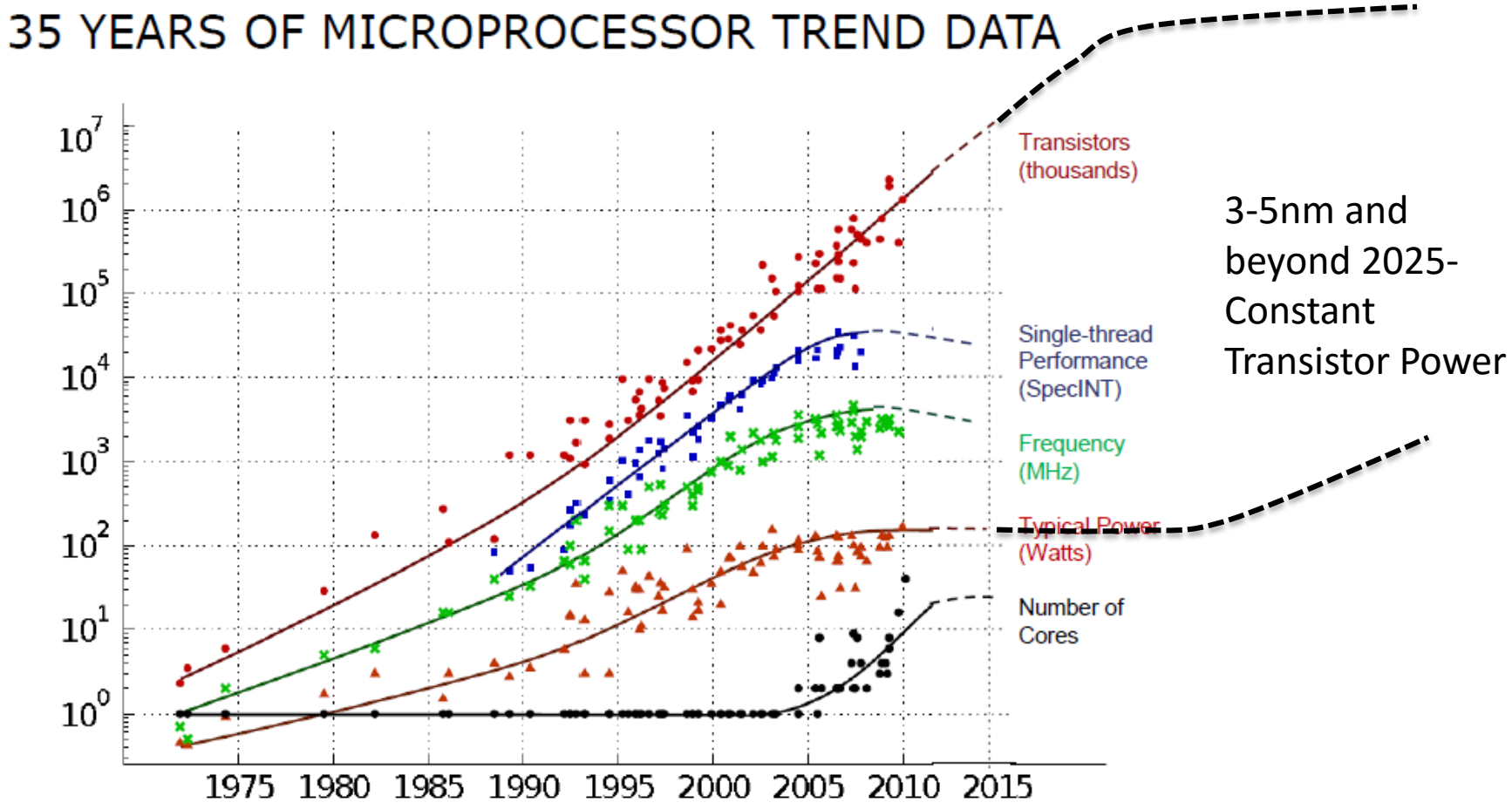
- Much of underlying IT performance growth due to Moore's law
  - "LSI: x2 transistors in 1~1.5 years"
  - Causing qualitative "leaps" in IT and societal innovations
  - The main reason we have supercomputers and Google...
- But this is slowing down & ending, by mid 2020s…!!!
  - End of Lithography shrinks
  - End of Dennard scaling
  - End of Fab Economics

Gordon Moore

The curse of constant transistor power shall soon be upon us

- How do we *sustain* "performance growth" beyond the "end of Moore"?
  - Not just one-time speed bumps
  - Or do we give up and so something else?

The trends will change past End of Moore's Law



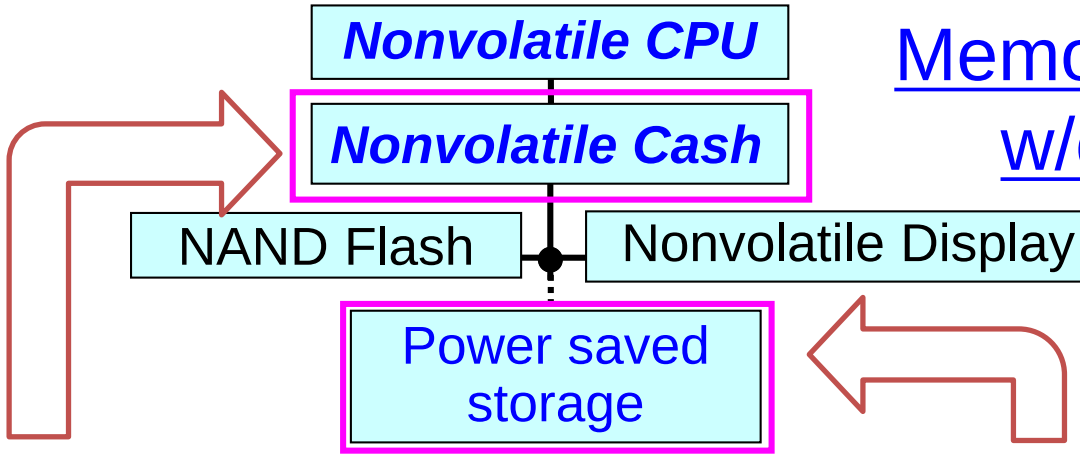
The “curse of constant transistor power” when Moore’s Law ends..

- Systems people have been telling the algorithm people that “FLOPS will be free, bandwidth is important, so devise algorithms under that assumption”
- This will certainly be true until exascale in 2020…
- But when Moore’s Law ends in 2025-2030, constant transistor power (esp. for logic) = FLOPS will no longer be free!
- So algorithms that simply increase arithmetic intensity will no longer scale beyond that point
- Have we been telling lies? NO! but is the consequence of device physics trend and the existing system design

Bold and Controversial Technology Prediction:
Performance growth via *data-centric computing*

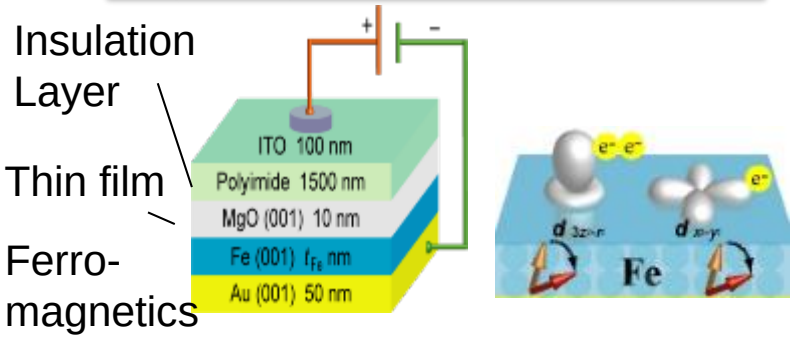
- Because data-related parameters (e.g. capacity and bandwidth) *will still continue to grow beyond 2020s (but not compute)*
- Can grow transistor count of system for compute, but CANNOT use them AT THE SAME TIME => **multiple computing units specialized to type of data**
- **Continued capacity growth** with 3D stacking (esp. direct silicon layering) and low power NVM (e.g. ReRAM)
- Data movement energy initially a problem but will be **capped constant** by dense 3D design and advanced optics from long-haul carrier technologies
- Almost back to the old "vector" days(?), but no free lunch – latency still problem, locality still important, *need* ***general algorithmic acceleration thru data capacity and bandwidth,*** *not FLOPS*
- *What are the implications to sparse numerical algotithms?*

Voltage-controlled Nonvolatile Magnetic RAM



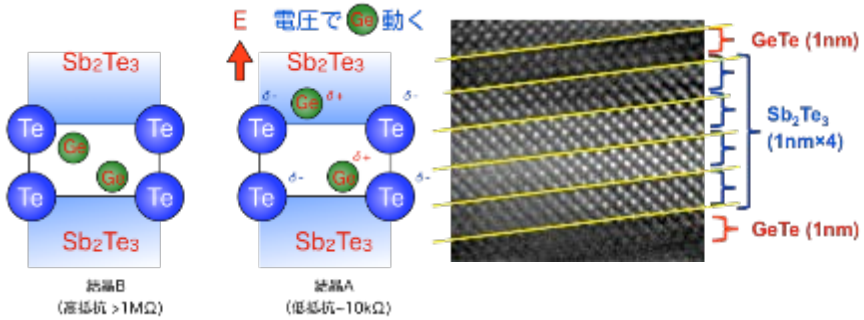
Memory keeping w/o power

Voltage Controlled Spin RAM



- voltage-induced magnetic anisotropy change
- Less than 1/100 rewriting power

Voltage Controlled Topological RAM

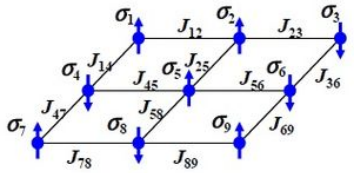


- Resistance change by the Ge displacement
- Loss by entropy: < 1/100

- Hitachi@ISSCC2015
  "An 1800-Times-Higher Power-Efficient 20k-spin Ising Chip for Combinational Optimization Problem with CMOS Annealing"
- Competitive to Quantum Annealing, room temperature, easy to scale
- Could be applicable to deep learning?

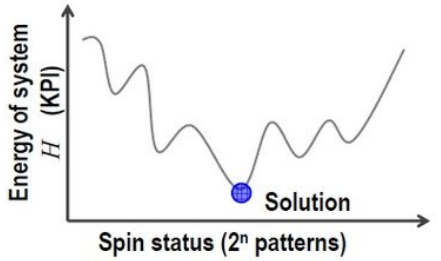
Computing with Ising model

- Ising model: expressing behavior of magnetic spins
- Using Ising model as natural phenomenon to map problems



$$H = -\sum_{\langle i,j \rangle} J_{ij}\sigma_i\sigma_j - \sum_j h_j\sigma_j$$

H: Energy of system
σ_i: Spin status (+1/-1)
J_{ij}: Interaction coefficient

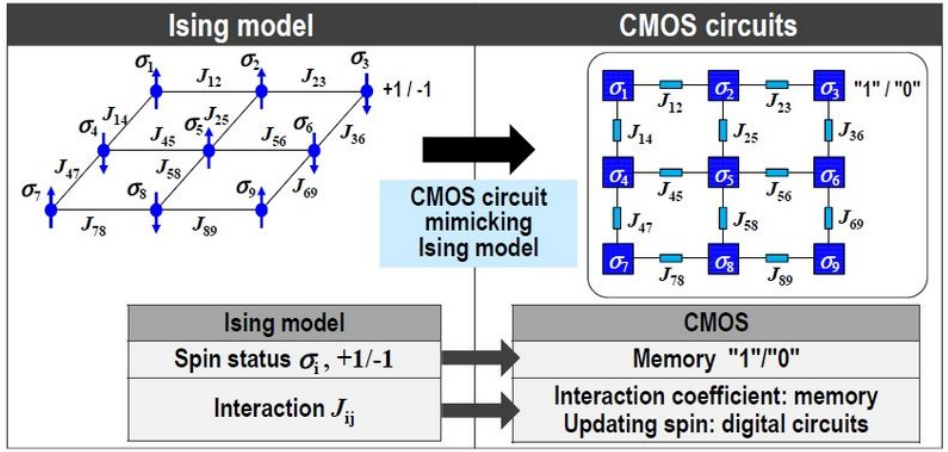


n: number of spins

© 2015 IEEE International Solid-State Circuits Conference

CMOS Ising computing

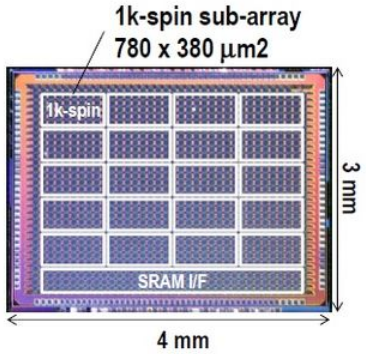
- Mimicking Ising model with CMOS circuits
- Easy to manufacture, easy to use, good scalability



Ising model	CMOS
Spin status σ_i, +1/-1	Memory "1"/"0"
Interaction J_{ij}	Interaction coefficient: memory Updating spin: digital circuits

© 2015 IEEE International Solid-State Circuits Conference

Fabrication results

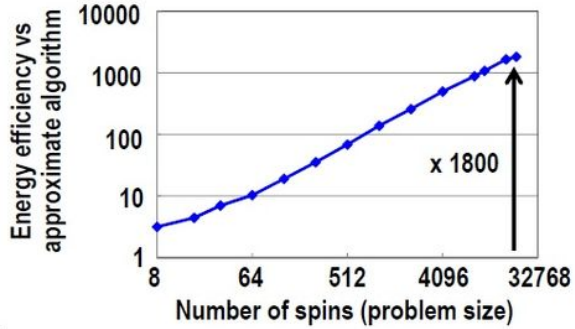


Items	Value
Number of spins	20k (80 x 256)
Process	65 nm
Chip area	4x3=12 mm²
Area of spin	11.27 x 23.94 =270 μm²
Number of SRAM cells	260k bits Spin value: 1 bit Interaction factor: 2 bit x 5=10 bits External magnetic coefficient: 2 bits
Memory IF	100 MHz
Interaction speed	100 MHz
Operating current of core circuits (1.1 V) Do not include IO	Write: 2.0 mA Read: 6.0 mA Interaction: 44.6 mA

© 2015 IEEE International Solid-State Circuits Conference

Measurement results of energy

- 1,800 times higher energy efficiency than conventional approximation algorithm on CPU



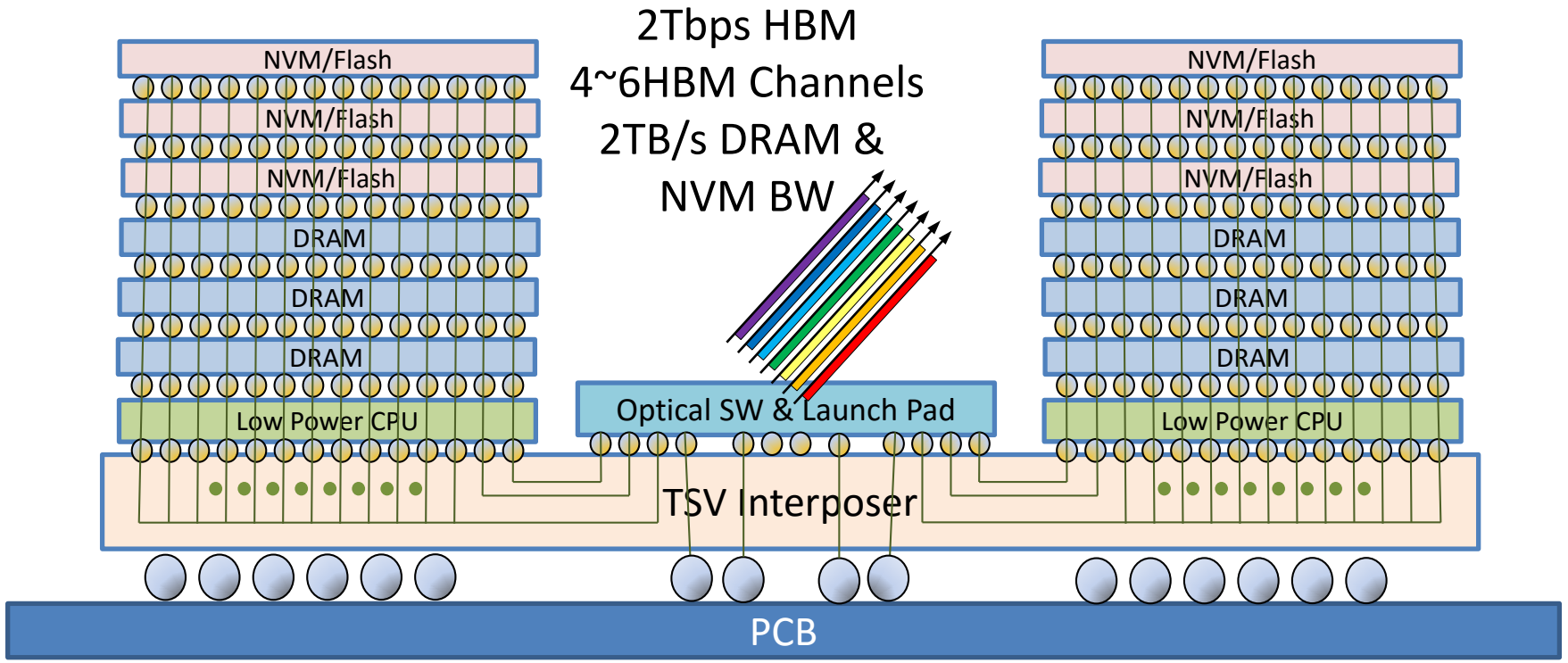
Conditions:
Randomly generated problems, energy for same preciseness solution
Ising chip: VDD=1.1 V, 100-MHz interaction, best solution among 10-times trial is selected.
Approximation algorithm: SG3(*) is operated on Core i5, 1.87 GHz, 10 W/core.

(*): Sera Kahruman et al., "On Greedy Construction Heuristics for the Max-Cut Problem," International Journal on Computational Science and Engineering, Volume 3, Number 3/2007, pp. 211-218, 2007.

© 2015 IEEE International Solid-State Circuits Conference

Tsubame 4: 2020- DRAM+NVM+CPU with 3D/2.5D Die Stacking

-The Ultimate Convergence of BD and EC-



Direct Chip-Chip Interconnect with DWDM optics

Low Power Processor allows Direct 3D Stacking

Optical Network Technology for Future Supercomputers & IDC

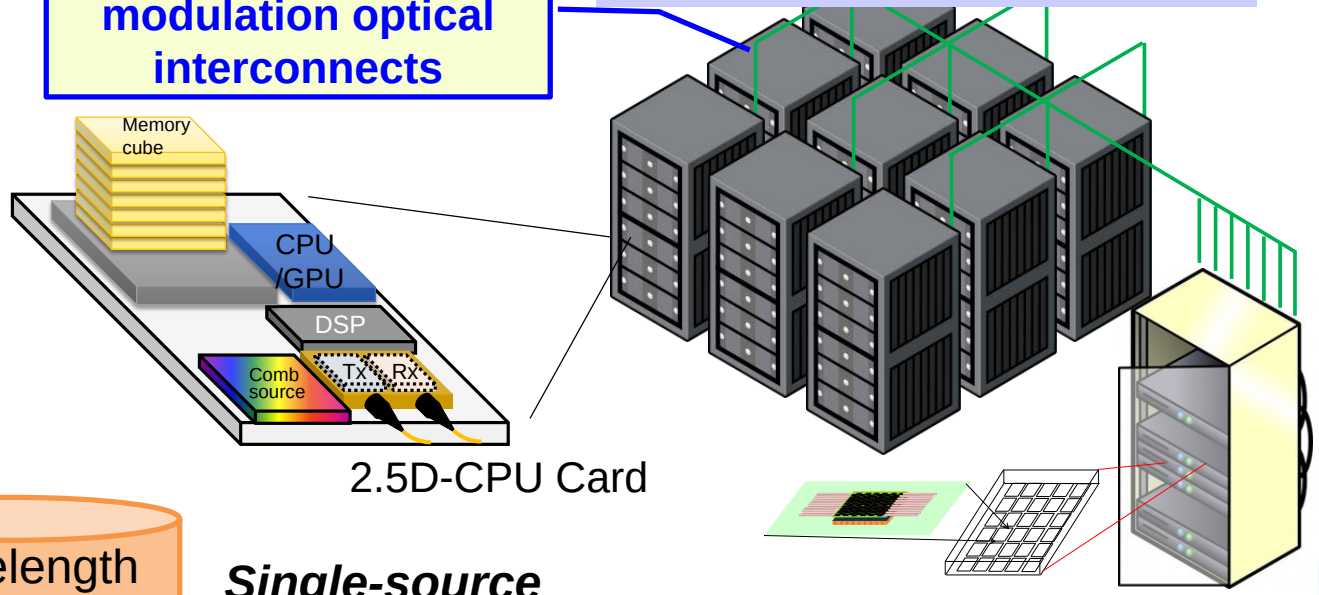
AIST IMPULSE

- Large-scale silicon photonics based cluster switches
- DWDM, multi-level modulation, highly integrated "elastic" optical interconnects
- Ultra-low energy consumption network by making use of optical switches

SC14 AIST Booth #2531

DWDM, multi-level modulation optical interconnects

Datacenter server racks



2.5D-CPU Card

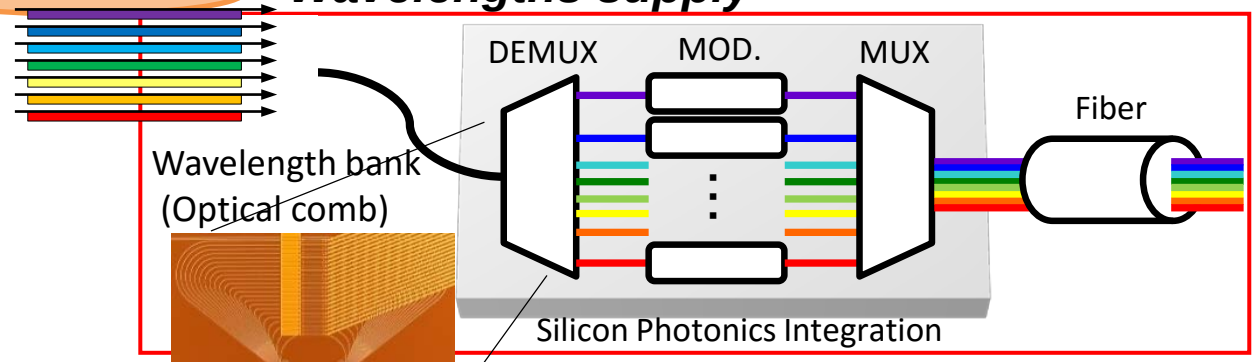
Silicon photonics cluster switches

- Ultra-compact switches based on silicon photonics
- 3D integration by amorphous silicon
- A new server architecture

Current electrical switches:
~1pbps
↓
~500Pbps

Wavelength Bank

Single-source Wavelengths supply



No of λs	Order of mod.	Bit rate
1	1	20 Gbps
4	8	640 Gbps
32	8	5.12 Tbps

Current state-of-the-art Tx
100Gbps
↓
~ 5.12Tbps

http://www.aist.go.jp

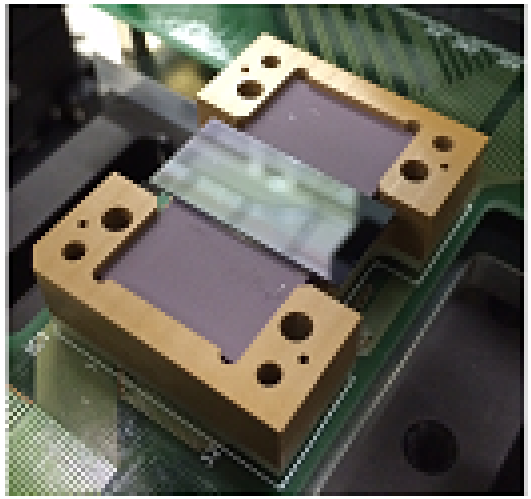
AIST ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

VICTORIES

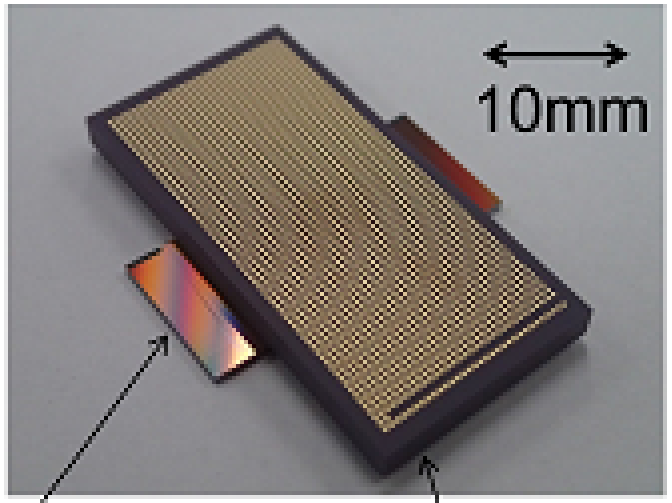
32 x 32 Optical Circuit Switch (Courtesy AIST)

- Ceramic LGA interposer with 0.5-mm pitch
- Flip-chip bonding with Au bumps and non-conductive paste
- LGA socket to contact PCB

LGA socket

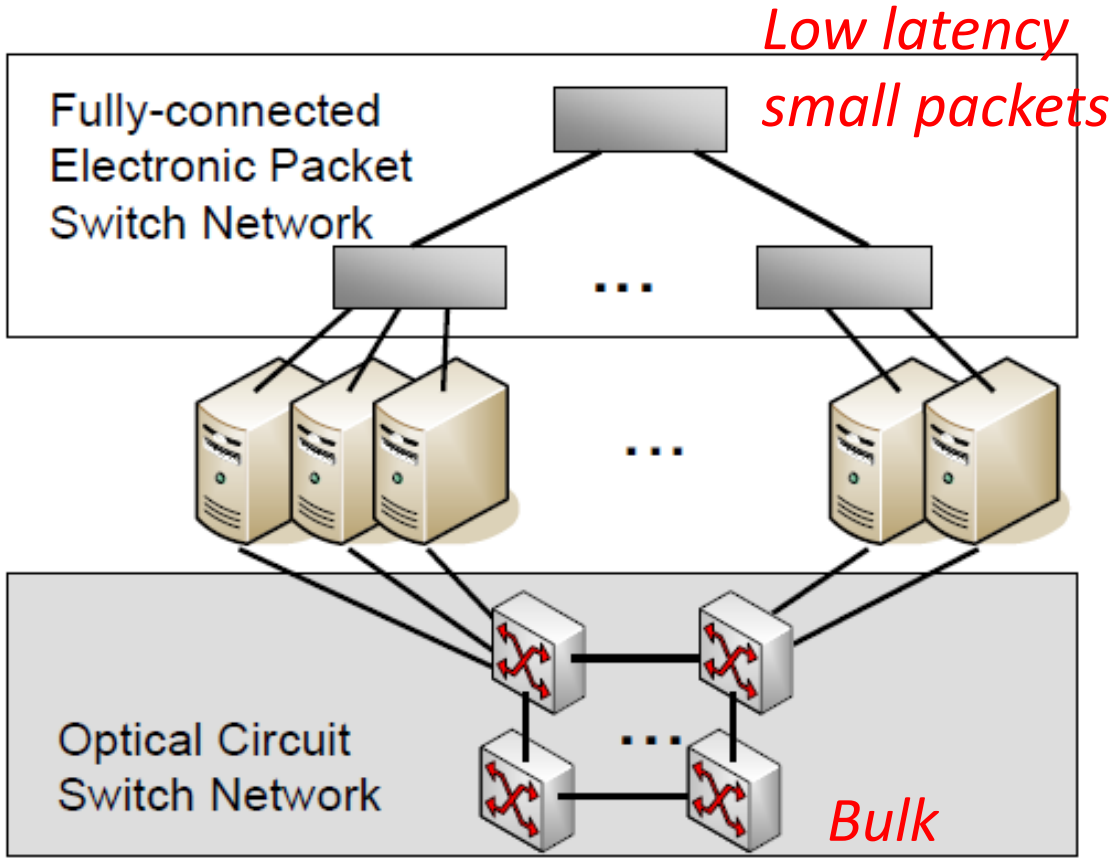


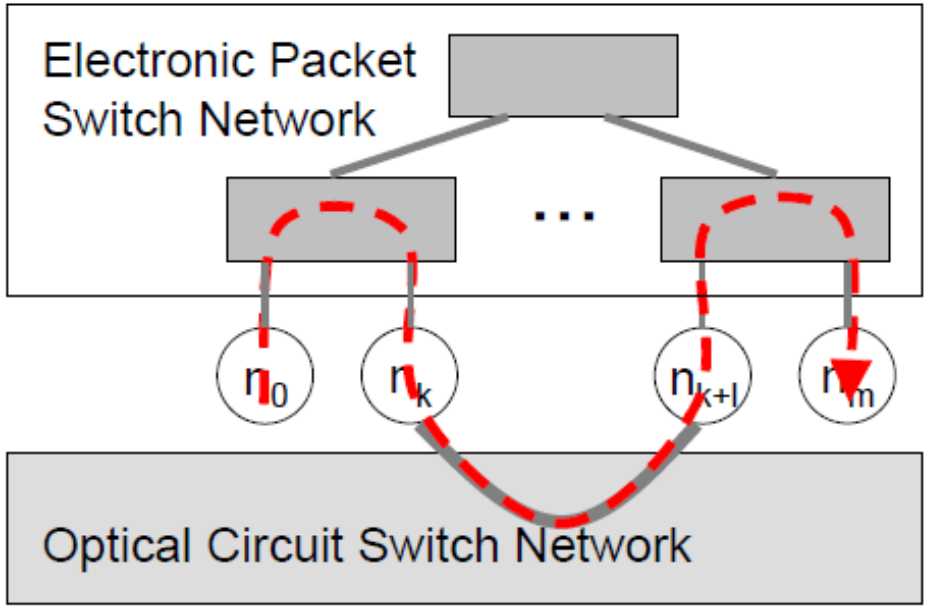
After FC bonding



Hybrid Electro-Optical Network w/shortcuts [Takizawa&Matsuoka LSPP07]

"Locality Aware MPI Communication on a Commodity Opto-Electronic Hybrid Network"

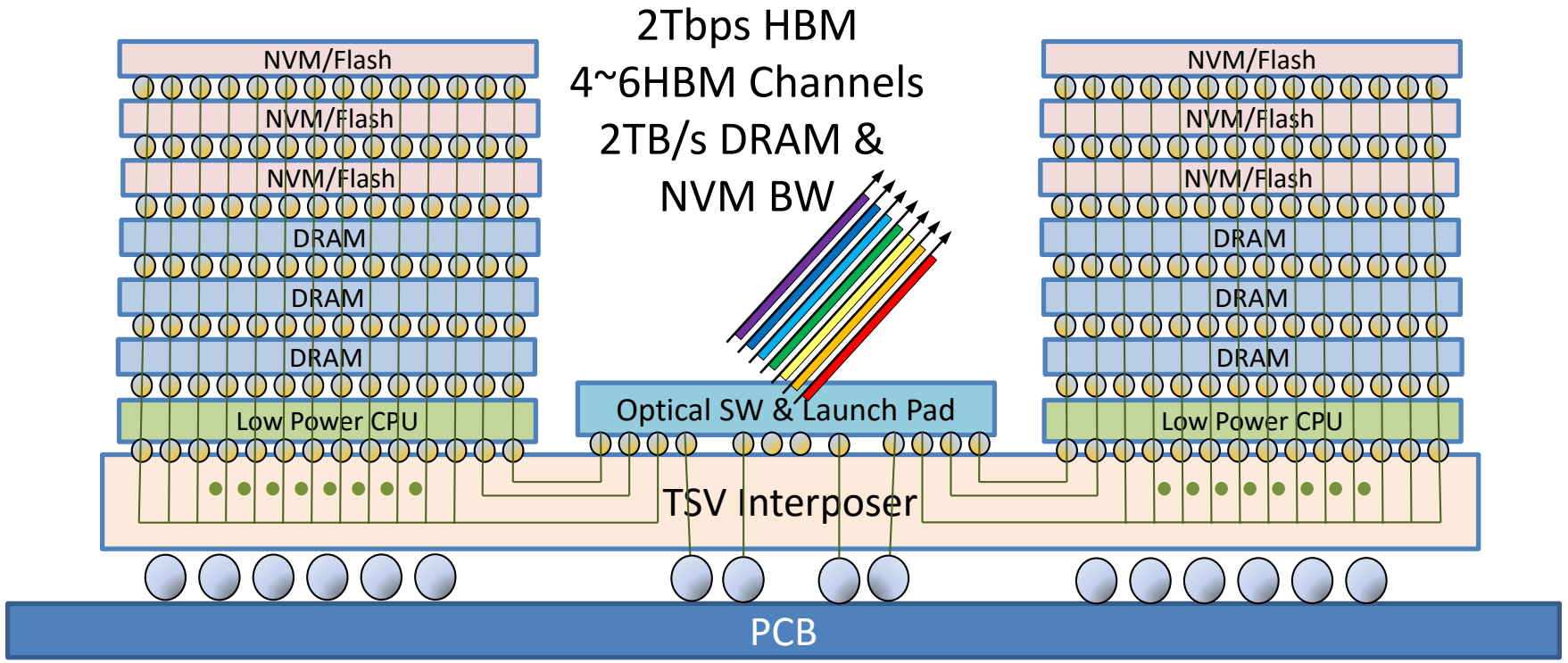




EO-OE Shortcut Forwarding of Messages Across Switches

Tsubame 4: 2020- DRAM+NVM+CPU with 3D/2.5D Die Stacking

-The Ultimate Convergence of BD and EC-

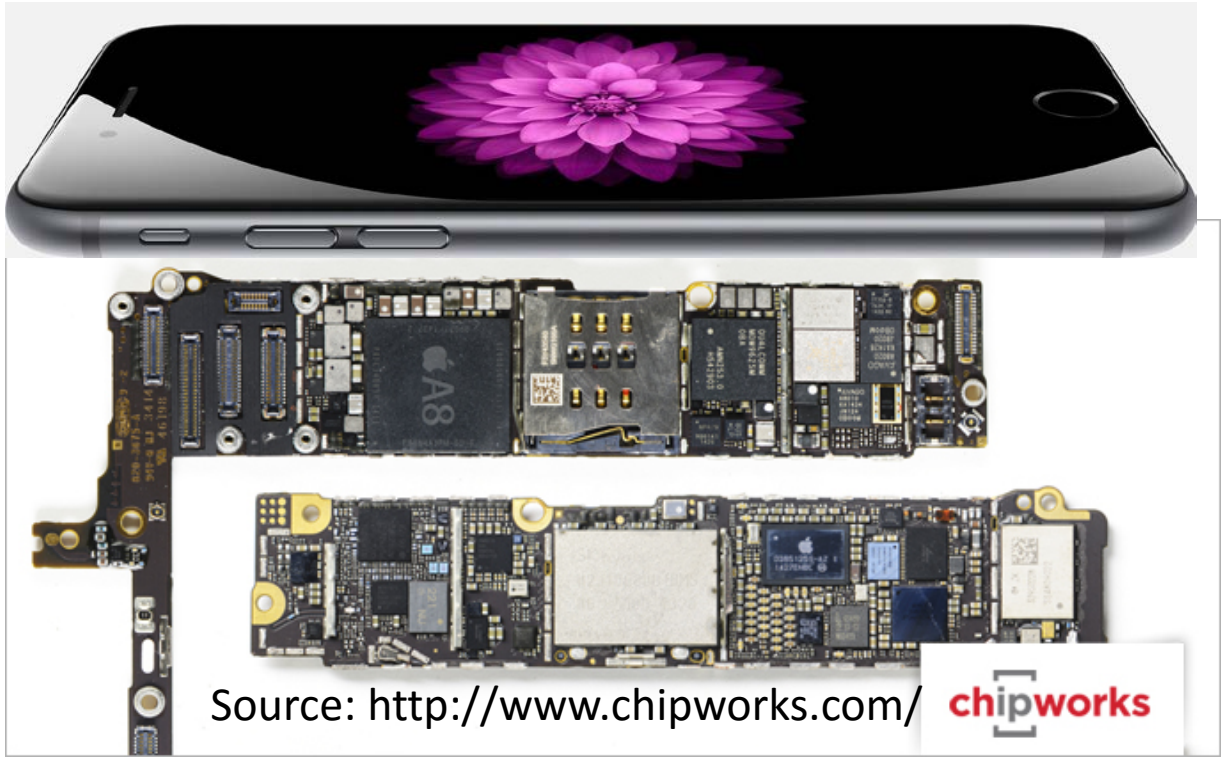


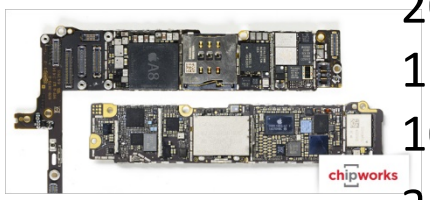
Direct Chip-Chip Interconnect with DWDM optics

Low Power Processor allows Direct 3D Stacking

2013 TSUBAME-KFC 4GPUnode

VS.





2021~22 Convergence Node
7nm SoC technology
1 Teraflop CPU+GPU hetero
200GB/s fast memory
1 Terabyte NVM
100GB/s I/O (to neighbor node and to long-range optics)
5~15Watts
$200 – exploits ecosystem
Runs today's software!

"We should *optimally* design future SCs like Cell Phones or even human brains so that we *never repair* them, but simply work around"

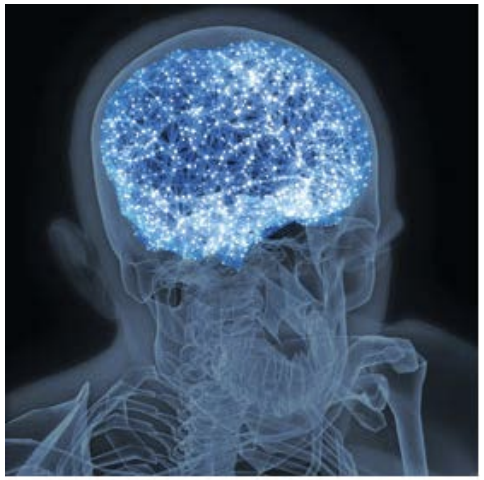


Image: Illustration by Mirko Ilic

SC14 New Orleans, LA
Wed 2:00PM

Fail-in-Place Network Design

[SC14, Domke, Hoefler, Matsuoka]

What if we never repaired network failures?
Will intelligent routing compensate?

All investigated routing algorithms show limitations

- Fat-tree, UpDown, DOR, Torus2QoS
- MinHop, SSSP, DFSSSP, LASH

Topology-aware routings

- High throughput decrease possible with small failure percentage
- Fail to route highly damaged netw.
- Routes not always DL-free (DOR)

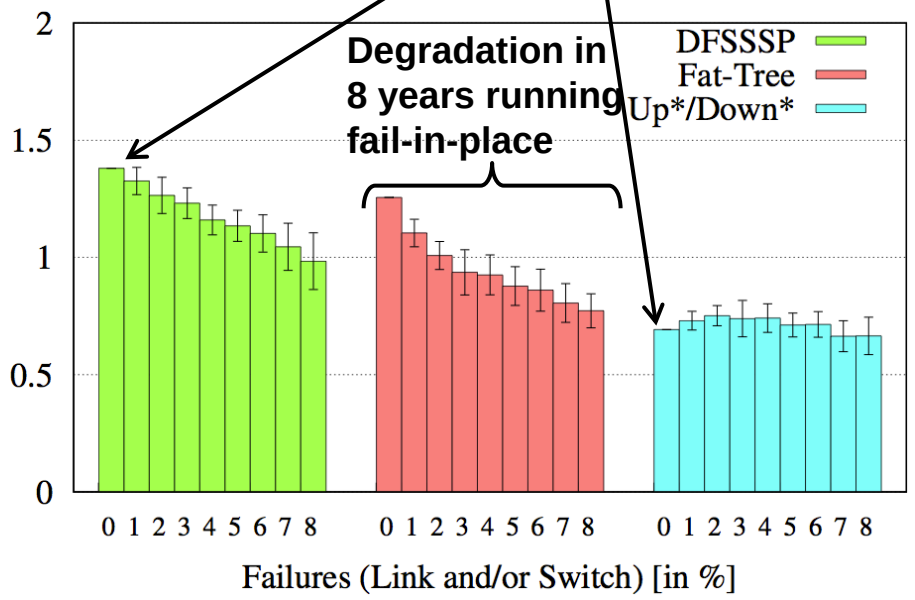
Topology-agnostic routing algorithms

- Ignore deadlocks (MinHop, SSSP)
- Deadlock-avoidance via VLs can be impossible for large scale netw.

TABLE II. INTERCEPT, SLOPE, AND R^2 FOR TSUBAME2.0

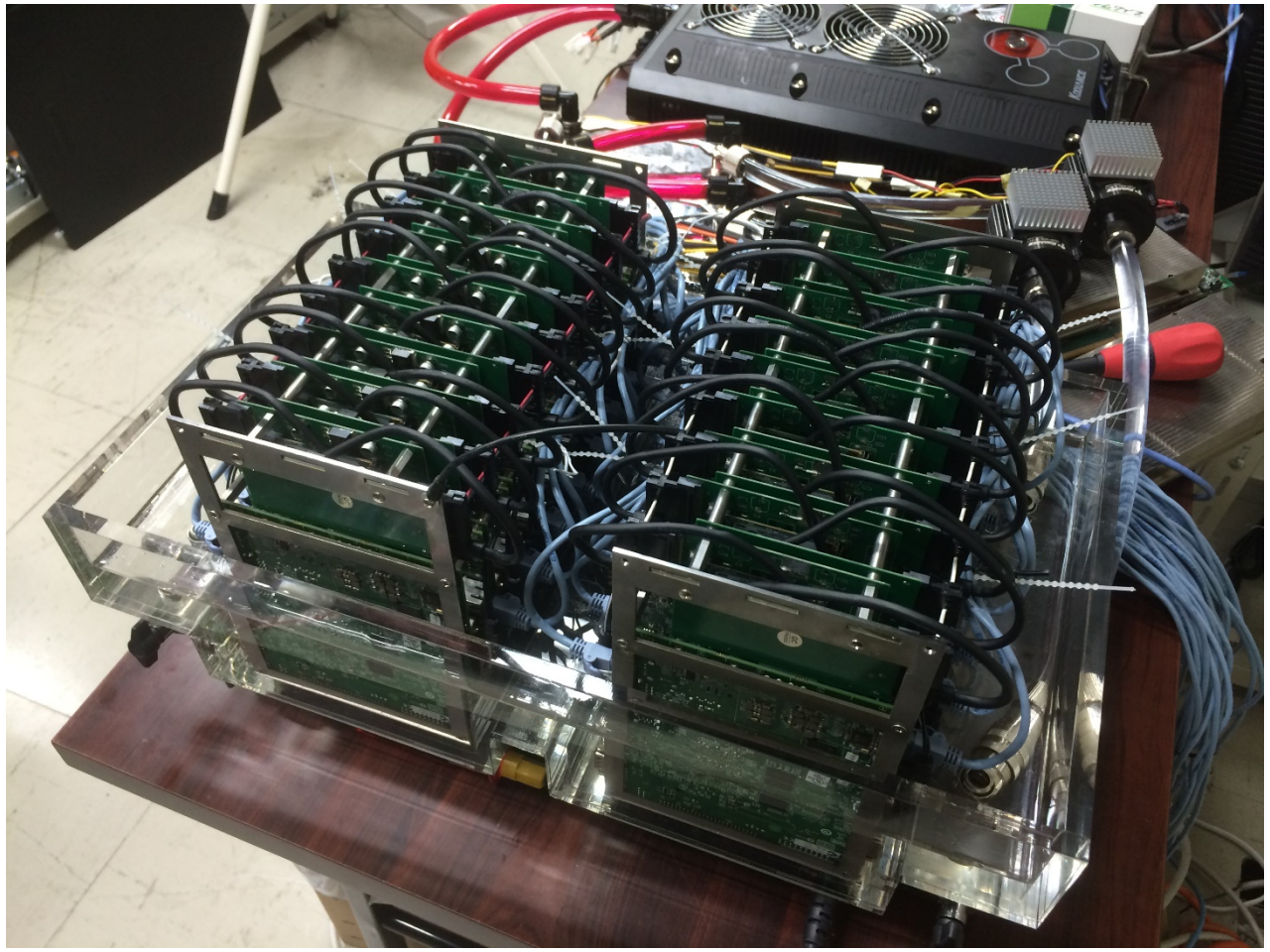
HPC system	Routing	Intercept [in Gbyte/s]	Slope	R^2
TSUBAME2.0	**DFSSSP**	1,393.40	-1.33	0.62
	Fat-Tree	1,187.19	-1.48	0.66
	Up/Down**	717.76	-0.08	0.01

Changing from Up*/Down* (default) to DFSSSP routing on TSUBAME2.5 improves the throughput by 2.1x for the fault- free network and increases TSUBAME's fail-in-place characteristics.



Fail-in-Place Network Design is possible! (but we have to improve the routing)

GoldenBox "Proto1" (NVIDIA K1-based) Demo at SC14 Tokyo Tech. Booth #1857

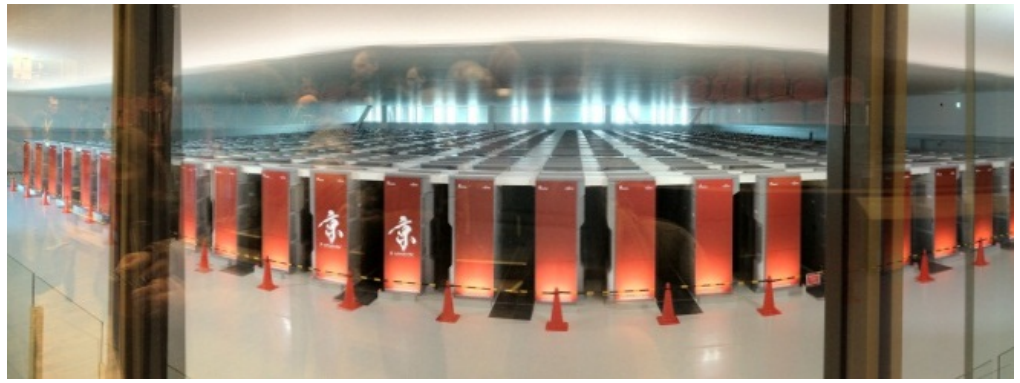


- 36 Node Tegra K1, ~11TFlops SFP
- ~700GB/s BW
- 100-700Watts
- Integrated mSata SSD, ~7GB/s I/O
- Ultra dense, Oil immersive cooling
- Same SW stack as TSUBAME2

2022: x10 Flops, x10 Mem Bandwidth, silicon photonics, x10 NVM, x10 node density, with new device and packaging technologies

TSUBAME4 beyond 2021~2022 K-in-a-Box (Golden Box)
BD/EC Convergent Architecture

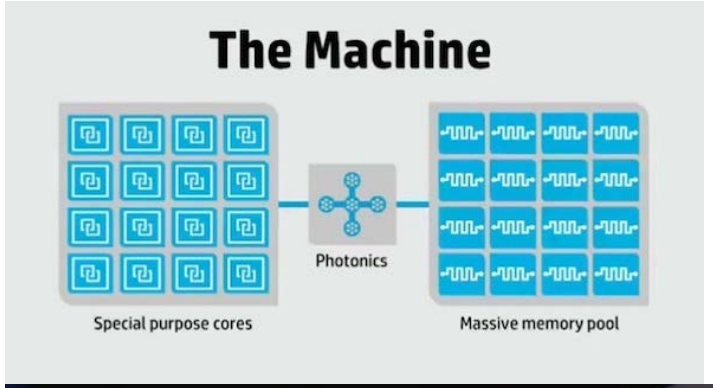
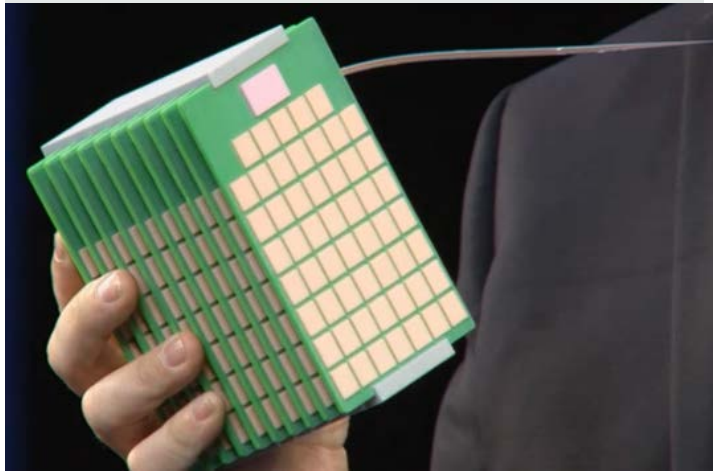
1/500 Size, 1/150 Power, 1/500 Cost, x5 DRAM+ NVM Memory



10 Petaflops, 10 Petabyte Hiearchical Memory (K: 1.5PB),
10K nodes
50GB/s Interconnect (200-300Tbps Bisection BW)
(Conceptually similar to HP “The Machine”)

Datacenter in a Box

Large Datacenter will become “Jurassic”

Similar HPC/IDC Technology Under Research

- HP “The Machine”
  - “Memrister” ReRAM and Silicon Photonics for high density next generation HPC/IDC machine
  - ReRAM as “universal memory” eliminating DRAM, remote memory pool, single memory hierarchy with large capacity
  - A new “Machine” OS to exploit capacity, diffuse memory and secondary storage
  - High risk due to memrister-only design, new OS, little exploitation of locality
- UC Berkeley FireBox
  - NVM and large 100 core large-scale SoC high density implementation, 1000 nodes & 100 petabytes per rack
  - System-wide encryption for IDC usage
  - Details unclear, large-scale SoC unfit for 3-D stacking, low performance for HPC