Covergence of Extreme Big Data and HPC

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

HPC2014 Italy
Cetraro, Italy
20140707

2013: TSUBAME2.5 No.1 in Japan* in Single Precision FP, 17 Petaflops

(*but not in Linpack)

東京工業大学
Tokyo Institute of Technology

GSIC

北海道大学
東北大学
TOHOKU UNIVERSITY
東京大学
THE UNIVERSITY OF TOKYO
大阪大學
国立大学法人
名古屋大学
九州大学
KYUSHU UNIVERSITY
京都大学

All University Centers COMBINED 9 Petaflops SFP

~=

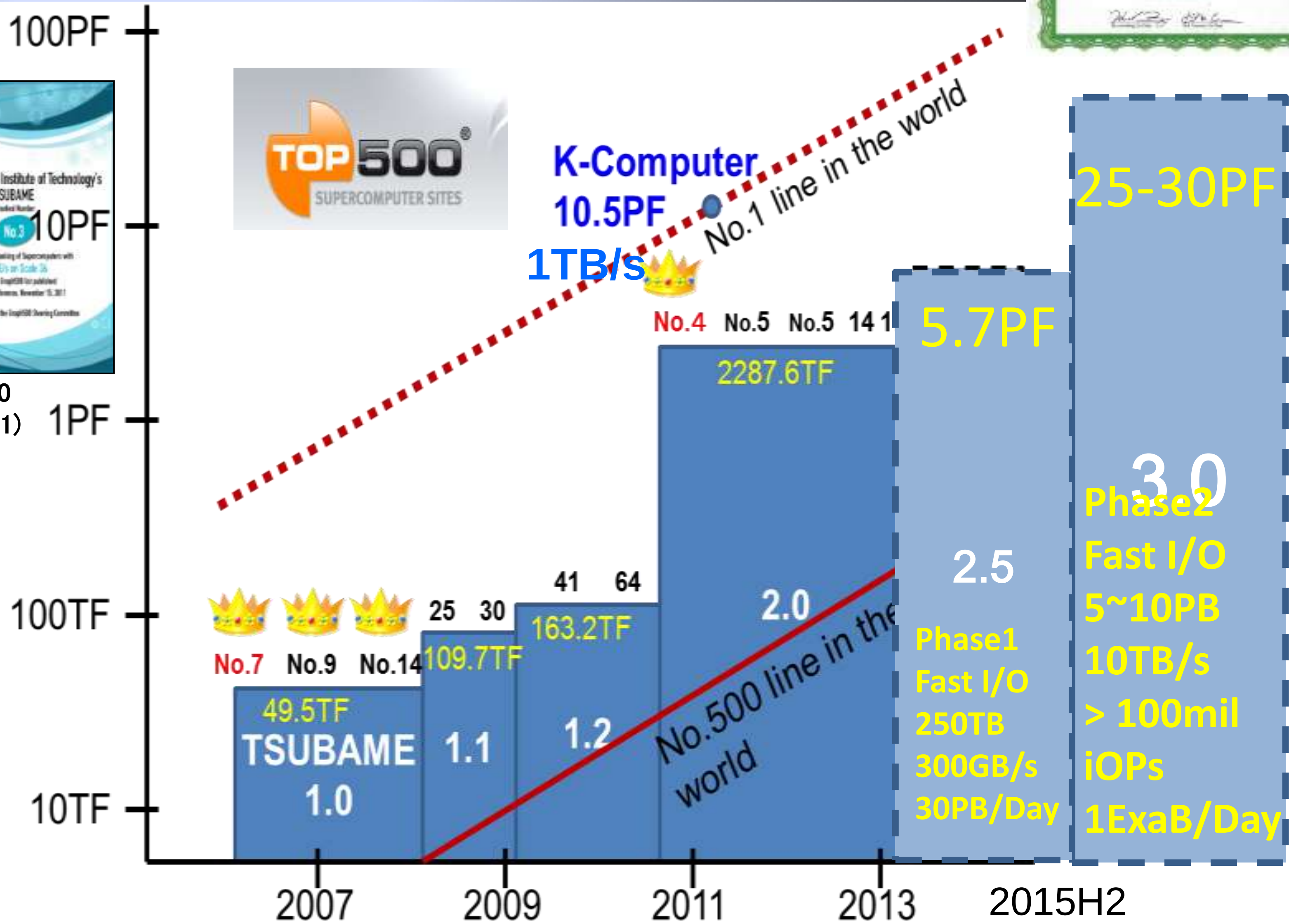
Total
17.1 Petaflops SFP
5.76 Petaflops DFP

K Computer
11.4 Petaflops SFP/DFP

TSUBAME Evolution
Towards Exascale and Extreme Big Data

Graph 500
No. 3 (2011)

HPC wire
Awards



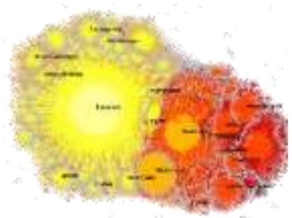
0. Extreme Big Data Backgroud

“Is Big Data really that Big?”

Extreme Big Data Example in Social NW rates and volumes are immense

Slide courtecy David A. Bader @ Georgia Tech

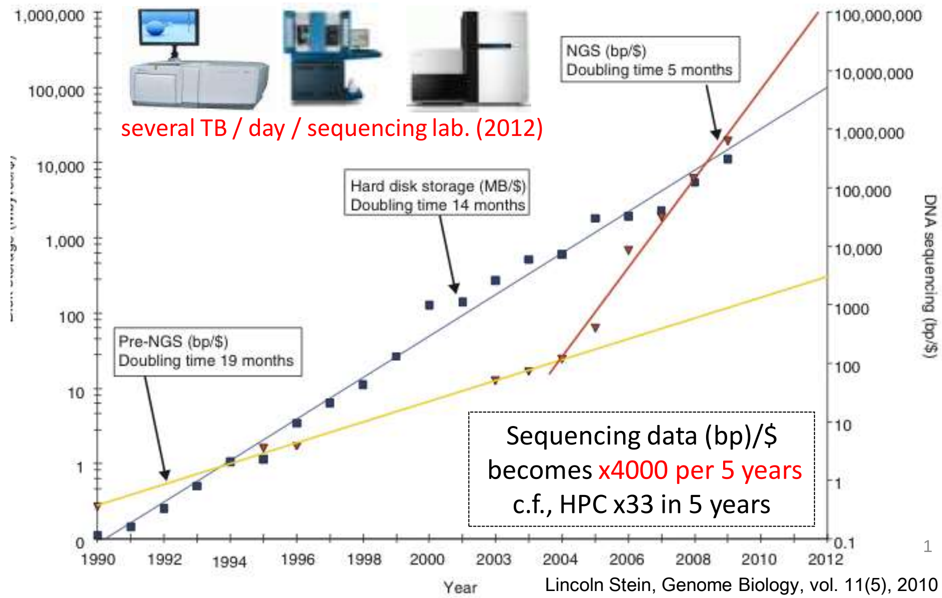
- Facebook:
  - ~1 billion users
  - average 130 friends
  - 30 billion pieces of content shared / month
- Twitter:
  - 500 million active users
  - 340 million tweets / day
- Internet – 100s of exabytes / year
  - 300 million new websites per year
  - 48 hours of video to You Tube per minute
  - 30,000 YouTube videos played per second



Extreme Big Data in Genomics

ɔact of new generation sequencers

[Slide Courtesy Yutaka Akiyama @ Tokyo Tech.]



Lincoln Stein, Genome Biology, vol. 11(5), 2010

Continuous Billion-Scale Social Simulation with Real-Time Streaming Data (Toyotaro Suzumura/IBM-Tokyo Tech)

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

Future "Extreme Big Data"

- **NOT mining Tbytes Silo Data**
- **Peta~Zetabytes of Data**
- **Ultra High-BW Data Stream**
- **Highly Unstructured, Irregular**
- **Complex correlations between data from multiple sources**
- **Extreme Capacity, Bandwidth, Compute All Required**

We will have tons of unknown genes

Metagenome analysis

[Slide Courtesy Yutaka Akiyama @ Tokyo Tech.]

- Directly sequencing uncultured microbiomes obtained from target environment and analyzing the sequence data
  - Finding novel genes from unculturable microorganism
  - Elucidating composition of species/genes of environments

Examples of microbiome

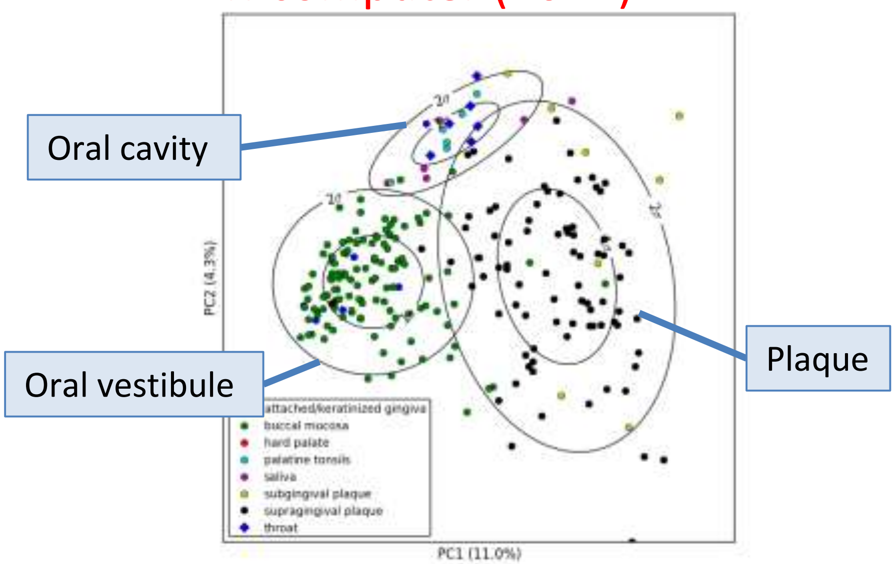
Results from Akiyama group@Tokyo Tech

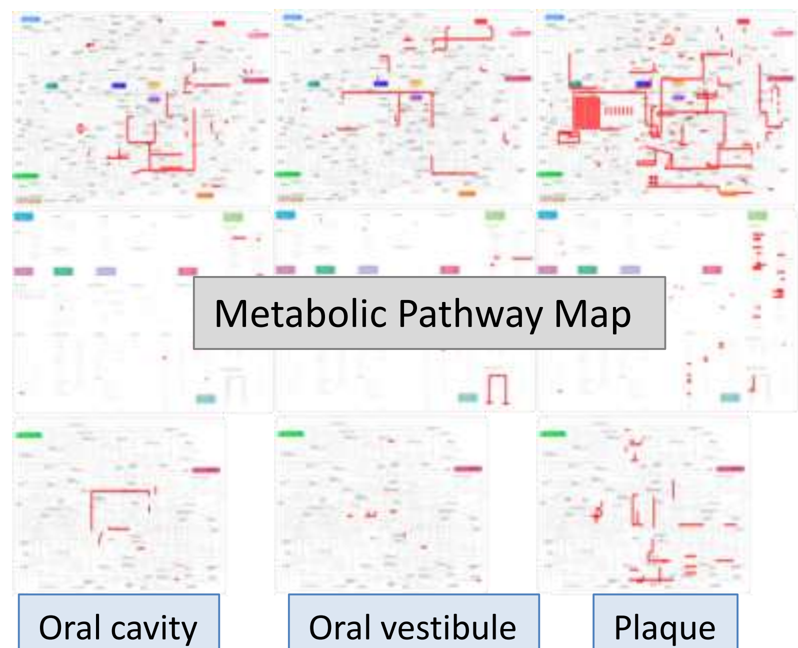
Ultra high-sensitive “big data” metagenome sequence analysis of human oral microbiome

- 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)





Extremely Large Graphs

- **The extremely large-scale graphs that have recently emerged in various application fields**
  - US Road network : 58 **million** edges
  - Twitter fellow-ship : 1.47 **billion** edges
  - Neuronal network : 100 **trillion** edges
- **Fast and scalable graph processing** by using **HPC**

Neuronal network @ EU Human Brain Project

89 billion nodes & 100 trillion edges

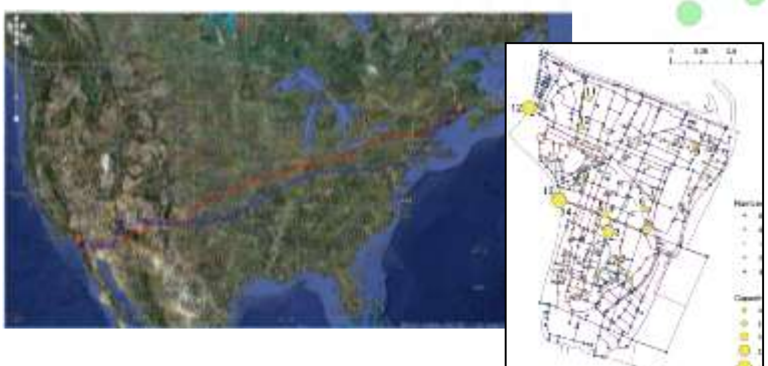
Social network

Twitter

61.6 million nodes & 1.47 billion edges

US road network

24 million nodes & 58 million edges



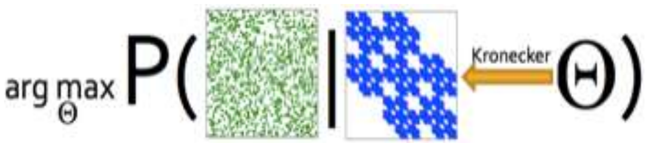
Cyber-security

15 billion log entries / day

Image: Illustration by Mirko Ilic

Graph500 “Big Data” Benchmark

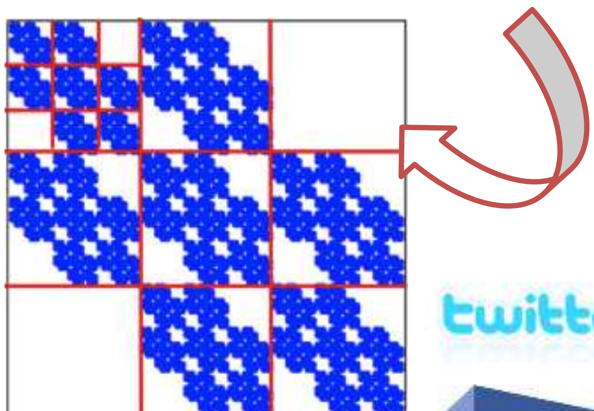
Kronecker graph BSP Problem



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

Graph 500 Takes Aim at a New Kind of HPC

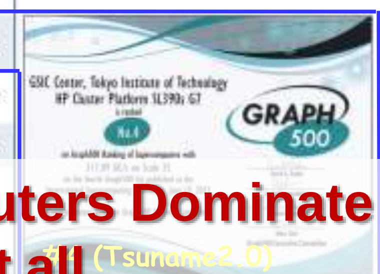
Richard Murphy (Sandia NL => Micron)

“ **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 4th Graph500 List (Jun2012) TSUBAME #4 w/GPUs

Toyotaro Suzumura, Koji Ueno, Tokyo Institute of Technology

Rank	Installation Site	Machine	Number of nodes	Number of cores	Problem scale	GTEPS
1	DOE/SC/Argonne National Laboratory	Mira/BlueGene/Q	32768	524288	38	3541.00
1	LLNL	Sequoia/Blue Gene/Q	32768	524288	38	3541.00
2	DARPA Trial Subset, IBM Development Engineering	Power 775, POWER7 8C 3.836 GHz	1024	32768	35	508.05
3	Information Technology Center, The University of Tokyo	Oakleaf-FX (Fujitsu PRIMEHPC FX 10)	4800	76800	38	358.10
4	GSIC Center, Tokyo Institute of Technology	TSUBAME	1366	16392	35	317.09
5	Brookhaven National Laboratory	BLUE GENE/Q	1024	16384	34	294.29
6	DOE/SC/Argonne National Laboratory	Vesta/BlueGene/Q	1024	16384	34	292.36
[illegible]	[illegible] Labs	Pleiades [illegible] "sandybridge"	[illegible]	[illegible]	[illegible]	[illegible]
8	NERSC/LBNL	XE6	[illegible]	[illegible]	[illegible]	[illegible]
9	NNSA and IBM [illegible]	NNSA/SC Blue Gene/Q [illegible]	[illegible]	65536	[illegible]	[illegible]

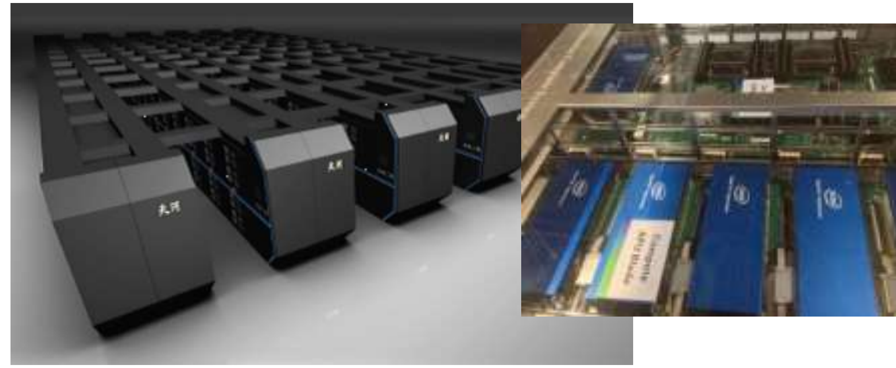


#4 (Tsuname2.0)

Reality: Top500 Supercomputers Dominate
No Cloud IDCs at all
TSUBAME2.0 #3(Nov.2011) #4(Jun.2012)

Top Supercomputers vs. Global IDC

K Computer (#1 2011-12) Riken-AICS
Fujitsu Sparc VIII-fx Venus CPU
88,000 nodes, 800,000CPU cores
~11 Petaflops (10^{16})
1.4 Petabyte memory, 13 MW Power
864 racks、3000m^2



Tianhe2 (#1 2013) China Gwanjou
48,000 KNC Xeon Phi + 36,000 Ivy Bridge Xeon
18,000 nodes, >3 Million CPU cores
54 Petaflops (10^{16})
0.8 Petabyte memory, 20 MW Power
??? racks、???m^2

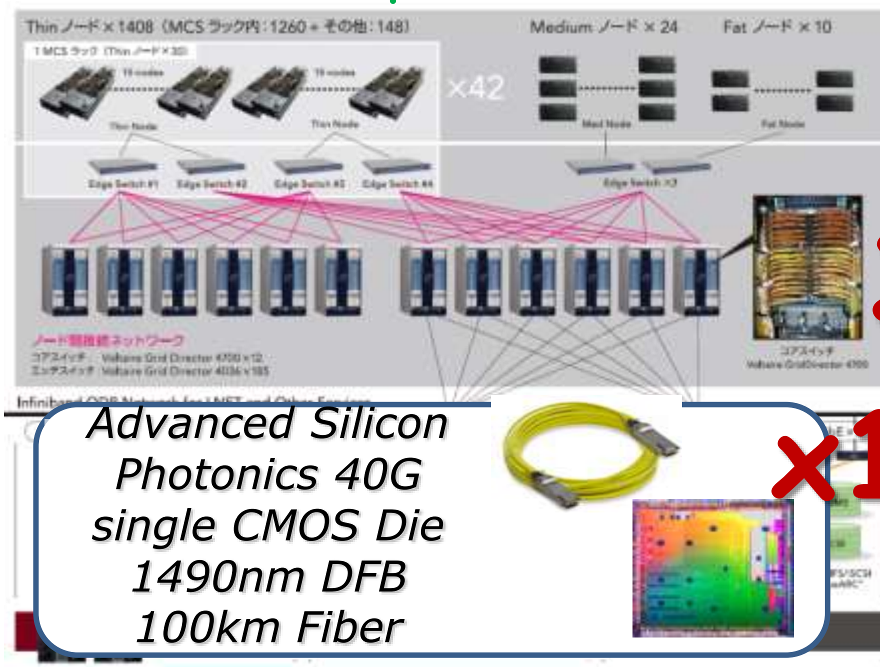
C.f. Amazon ~= 500,000 Nodes, ~6 million Cores??

#1 2012 IBM BlueGene/Q "Sequoia"
Lawrence Livermore National Lab
IBM PowerPC System-On-Chip
98,000 nodes, 1.57million Cores
~20 Petaflops
1.6 Petabytes, 8MW, 96 racks

DARPA study
2020 Exaflop (10^{18})
100 million~
1 Billion Cores

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

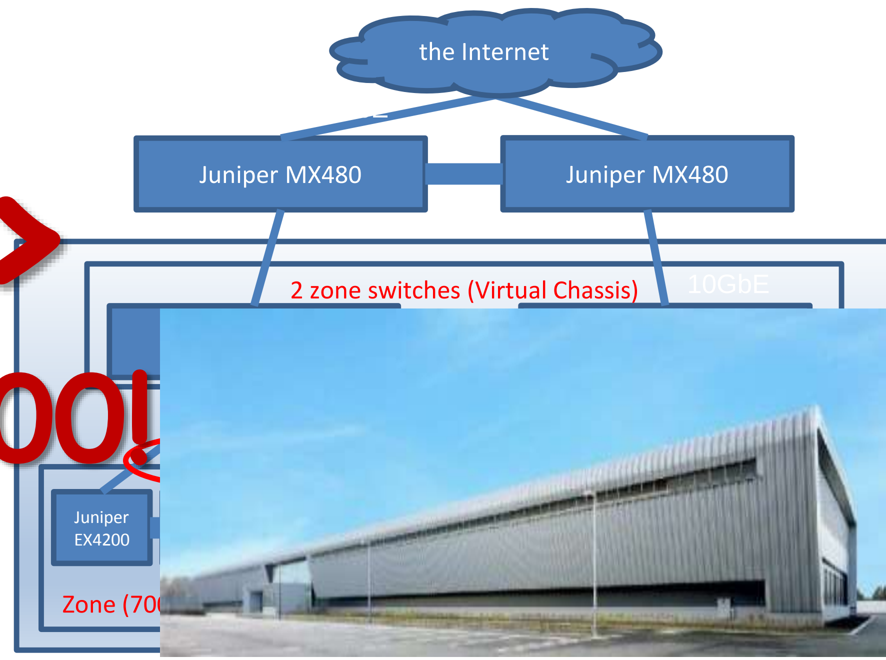
A Major Northern Japanese Cloud Datacenter (2013)



Advanced Silicon Photonics 40G single CMOS Die 1490nm DFB 100km Fiber

> >

×1000!



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

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

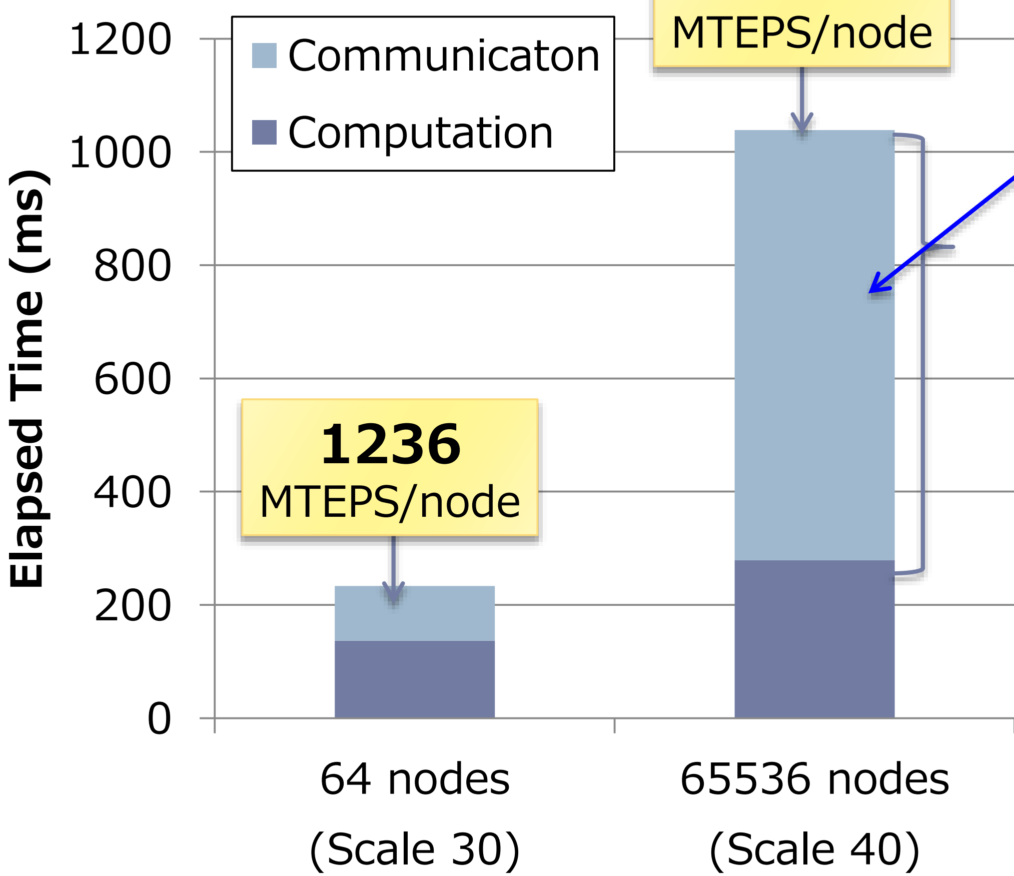
But what does “220Tbps” mean?

Global IP Traffic, 2011-2016 (Source Cicso)							
	2011	2012	2013	2014	2015	2016	CAGR 2011-2016
By Type (PB per Month / Average Bitrate in Tbps)							
Fixed Internet	23,288	32,990	40,587	50,888	64,349	81,347	28%
	71.9	101.8	125.3	157.1	198.6	251.1	
Managed IP	6,849	9,199	11,846	13,925	16,085	18,131	21%
	21.1	28.4	36.6	43.0	49.6	56.0	
Mobile data	597	1,252	2,379	4,215	6,896	10,804	78%
	1.8	3.9	7.3	13.0	21.3	33.3	
Total IP traffic	30,734	43,441	54,812	69,028	87,331	110,282	29%
	94.9	134.1	169.2	213.0	269.5	340.4	

TSUBAME2.0 Network has TWICE the capacity of the Global Internet, being used by 2.1 Billion users

Breakdown of BFS execution on K computer

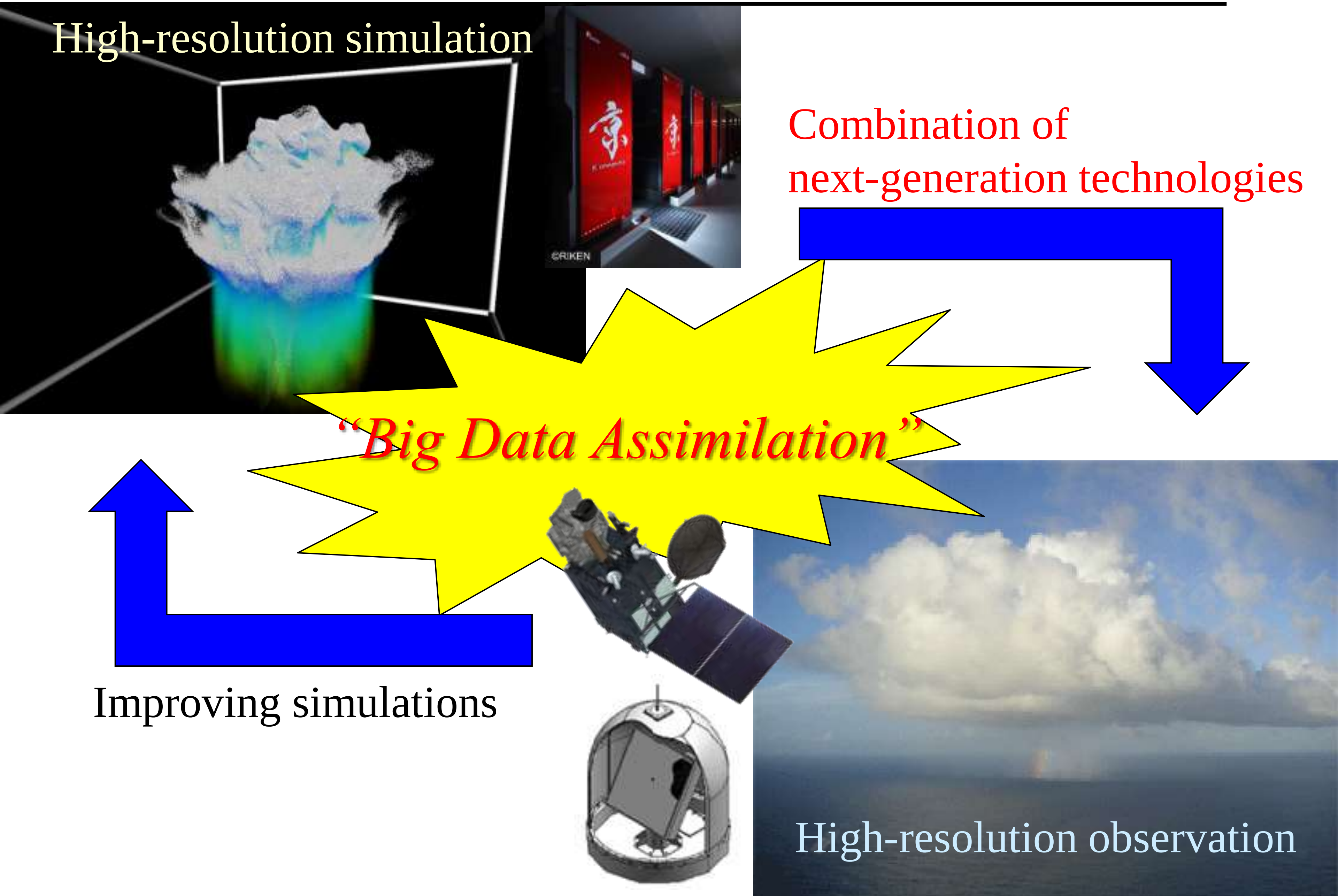
- Now, it is a communication intensive benchmark!!!



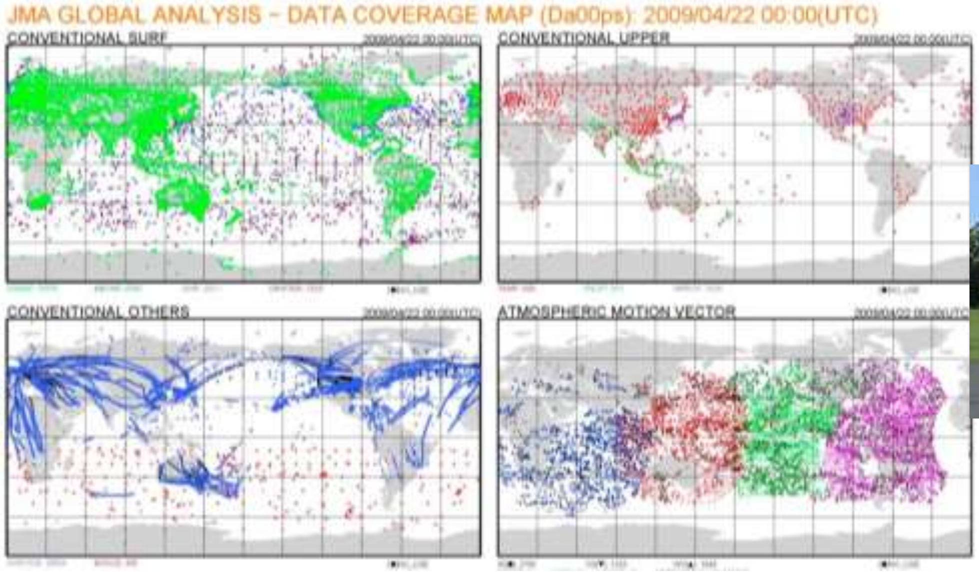
73% of the total execution time is spent on the communication waiting.

*Problem size is weak scaling

“Big Data Assimilation” in Weather

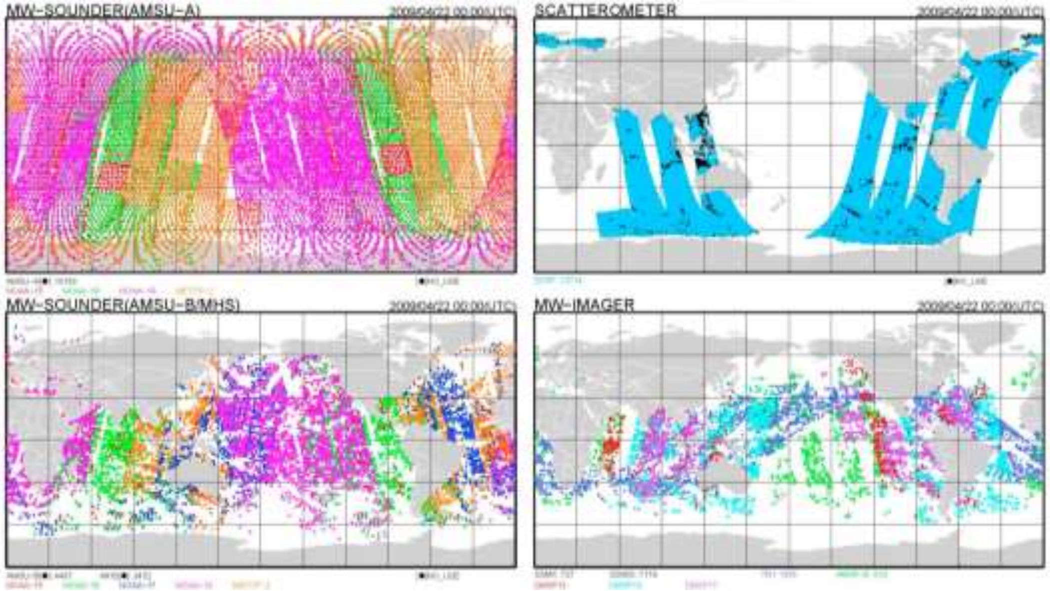


Collecting Atmospheric Data

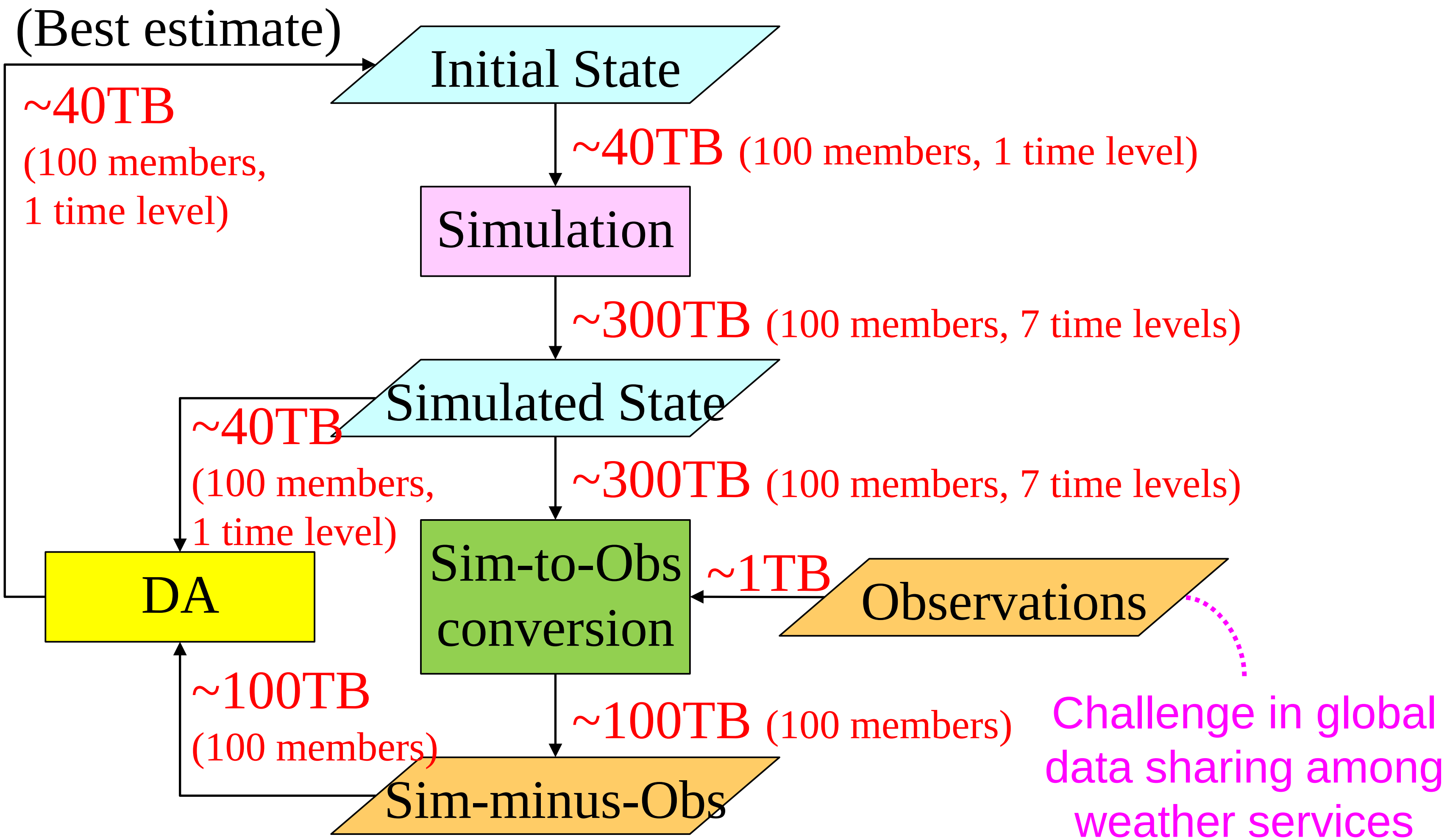


Global continuous collection

Variety of sensors, stationary and mobile



Flow chart with exa-scale data size



I/O intensive!

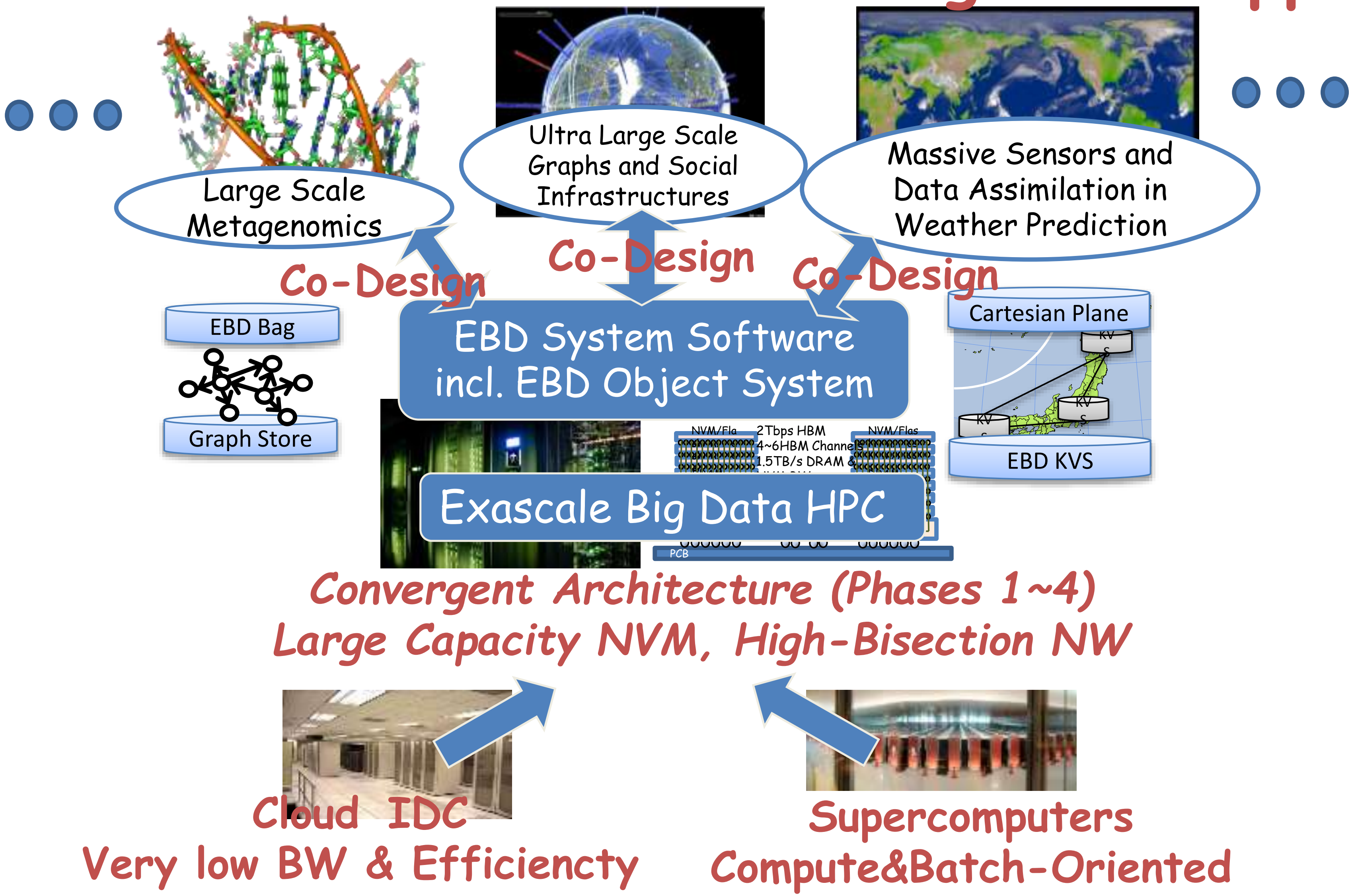
Repetitions of I/O between separate programs

In fact we will not be producing sufficient strorage!

- Worldwide HDD production: 550mil units and declining => ~1 Yottabytes/year
  - Global storage capacity 3~4 Yottabytes?
- Slow capacity CAGR predicted: 15%
- Flash increasing but still 10% of HDD
- C.f. Top500/Exascale CAGR 100%!
- Suppose 5-100 bytes/flop
  - Exascale machine 5~100 Exabytes HDD (&Tape)
    - 500K-10 mil HDDs&Tape, $50mil-$1bil
  - Conclusion: can't store data, need to process them

Extreme Big Data (EBD)
2013-2018 Research Scheme

Future Non-Silo Extreme Big Data Apps



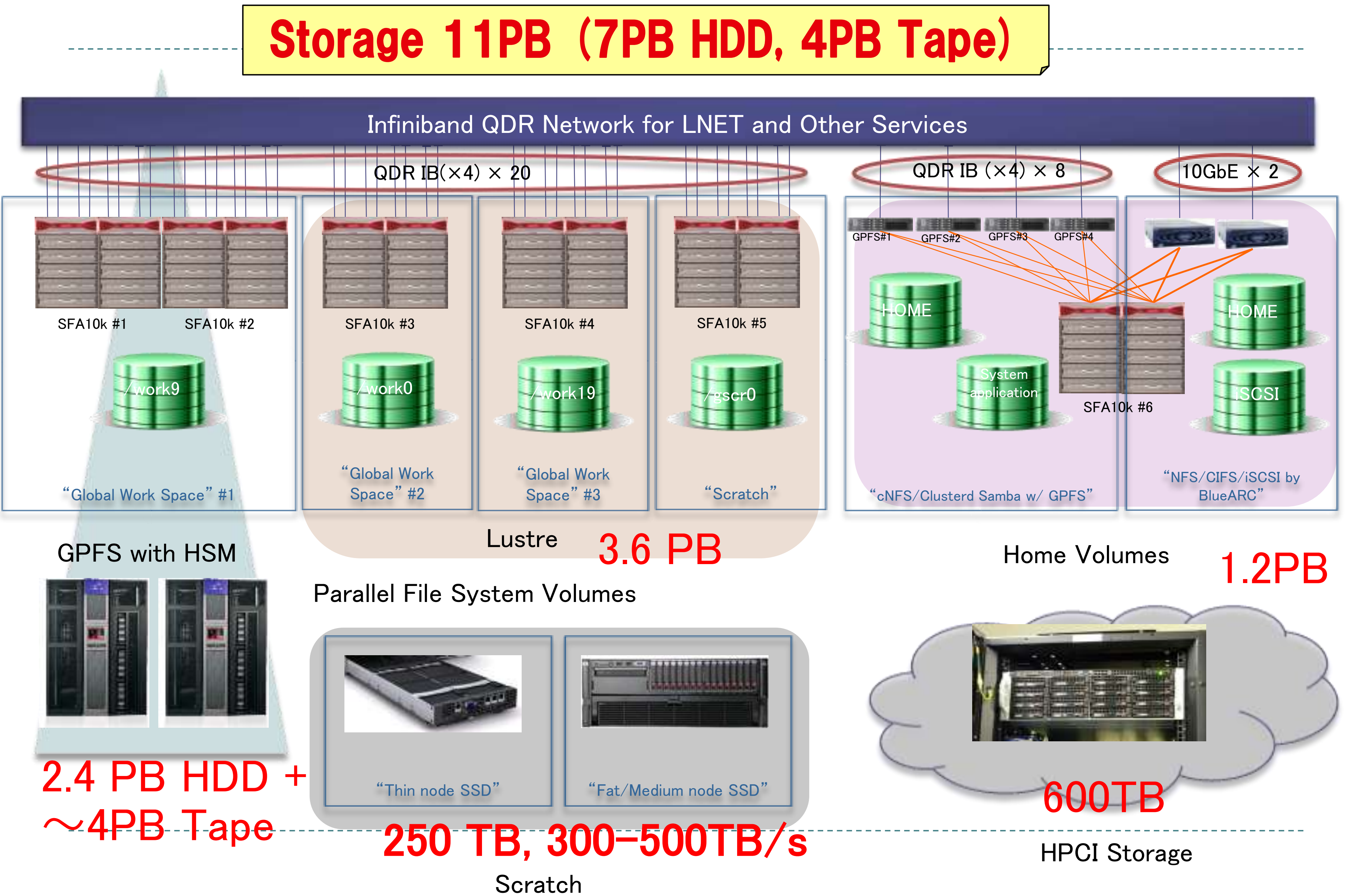
Japanese Big Data-HPC Convergence Projects

- *JST CREST Post Petascale (PD: Akinori Yonezawa)*
  - **Katsuki Fujisawa**(Univ. Kyushu): “Advanced Computing and Optimization Infrastructure for Extremely Large-Scale Graphs on Post Peta-Scale Supercomputers”
  - **Toshio Endo**(Tokyo Tech.) “Software Technology that Deals with Deeper Memory Hierarchy in Post-petascale Era”
  - **Osamu Tatebe** (Univ. Tsukuba): “System Software for Post Petascale Data Intensive Science”
- ***JST CREST “Big Data” (PD: M. Kitsuregawa & Y. Tanaka)***
  - **Takemasa Miyoshi (Riken AICS)**: Innovating "Big Data Assimilation" technology for revolutionizing very-short-range severe weather prediction
- *Other Projects*
  - **S. Matsuoka** (Tokyo Tech.) JSPS Grant-in-Aid S “Billion-way Resiliency”
  - TSUBAME3.0 !

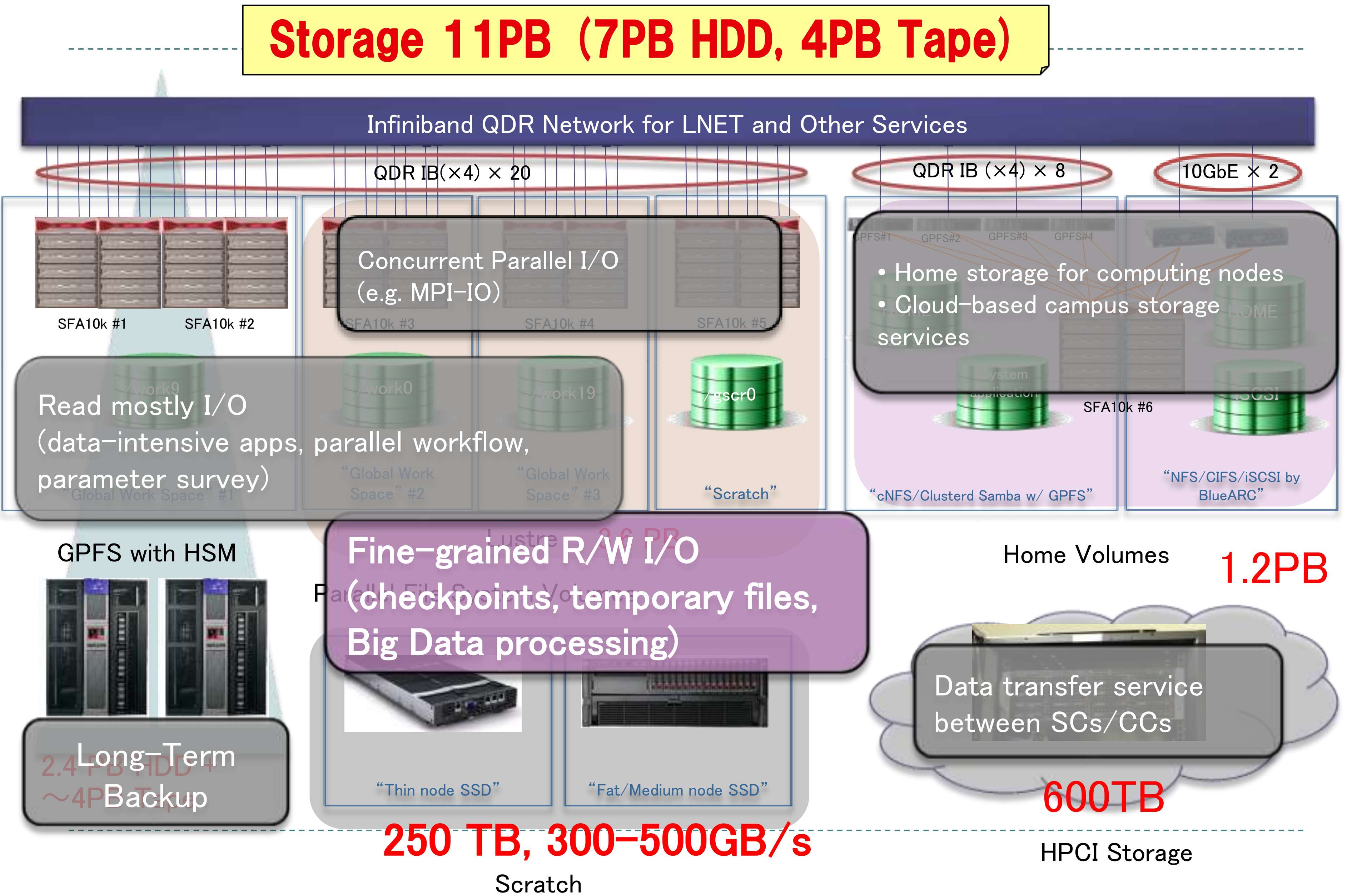
1. Extreme Big Data Machine Architecture

High Bandwidth
High Capacity
Deep Memory Hiearchy via NVMs & Next-Gen Optical Interconnect

TSUBAME2.0/2.5 Storage Overview



TSUBAME2.0/2.5 Storage Overview



TSUBAME-KFC
(Kepler Fluid Cooling)



A TSUBAME3.0 prototype system
with advanced next gen cooling
40 compute nodes are oil-submerged
1200 liters of oil (Exxon PAO ~1 ton)
#1 2013/11& 2014/6 Green 500

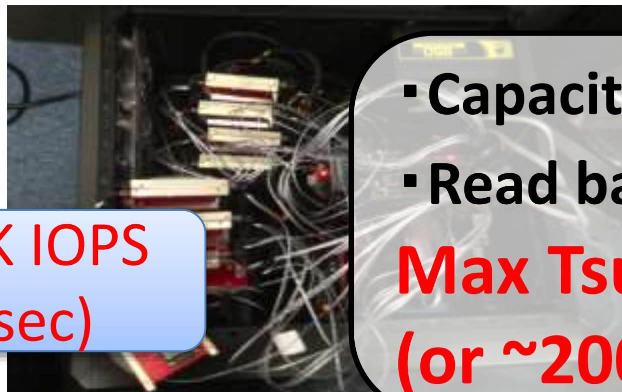
Single Node	5.26 TFLOPS DFP
System (40 nodes)	210.61 TFLOPS DFP 630TFlops SFP
Storage (3SSDs/node)	1.2TBytes SSDs/Node Total 50TBytes ~50GB/s BW

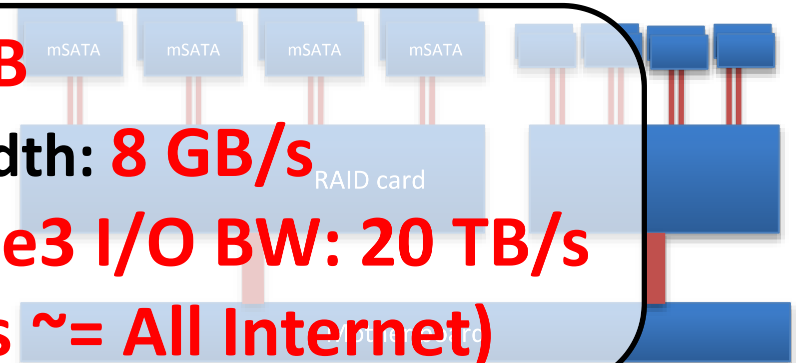

EBD- I/O
(Many-core I/O)

Preliminary I/O Evaluation on GPU and NVRAM

How to design local storage for next-gen supercomputers ?

- Local I/O prototype using 16 mSATA SSDs

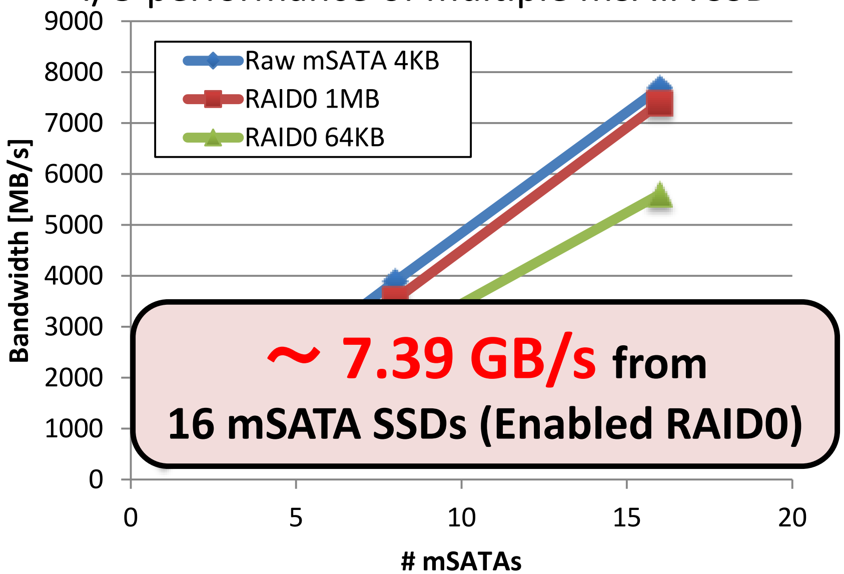




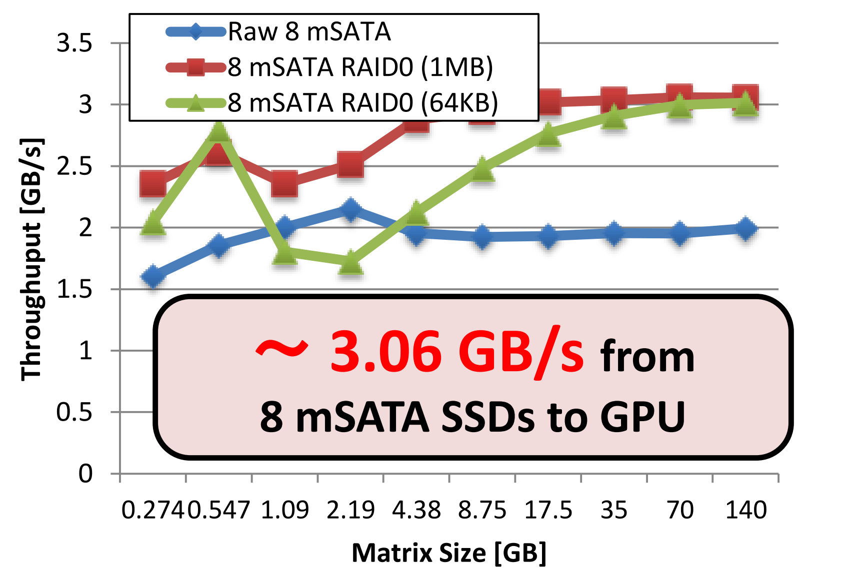
- Capacity: **4TB**
- Read bandwidth: **8 GB/s**

Max Tsubame3 I/O BW: 20 TB/s (or ~200Tbps ~= All Internet)

～320K IOPS (3 μ sec)



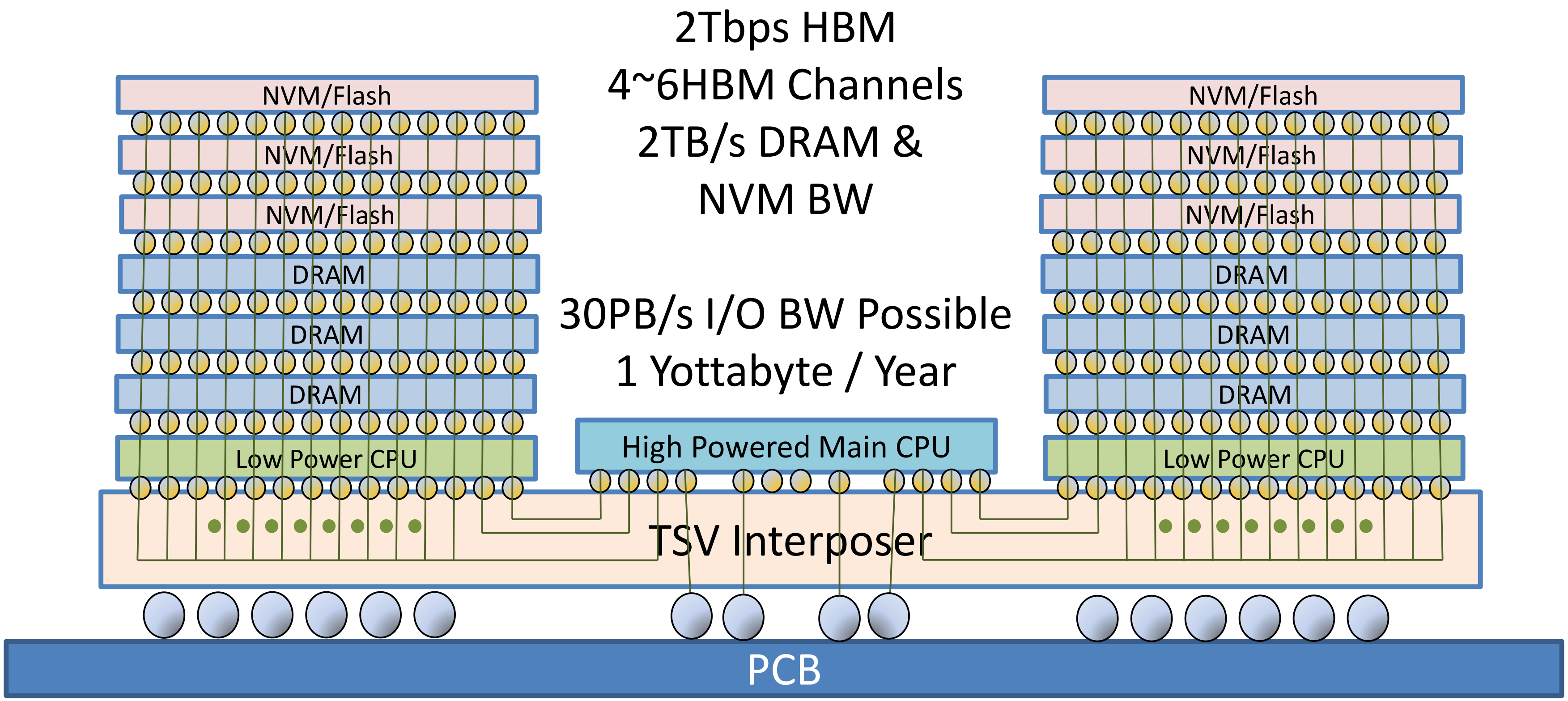
～ 7.39 GB/s from 16 mSATA SSDs (Enabled RAID0)



～ 3.06 GB/s from 8 mSATA SSDs to GPU

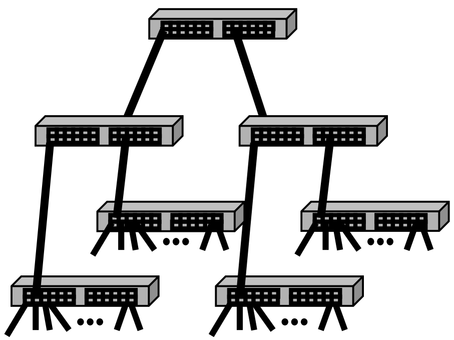
Tsubame 4: 2020- DRAM+NVM+CPU with 3D/2.5D Die Stacking

-The Ultimate Convergence of BD and EC-

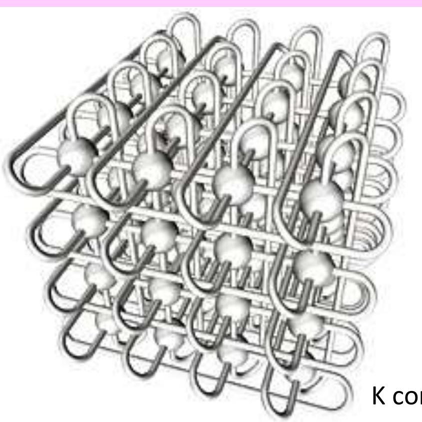


Direct Chip-Chip Interconnect with planar VCSEL optics

EBD Interconnects (NII Group)



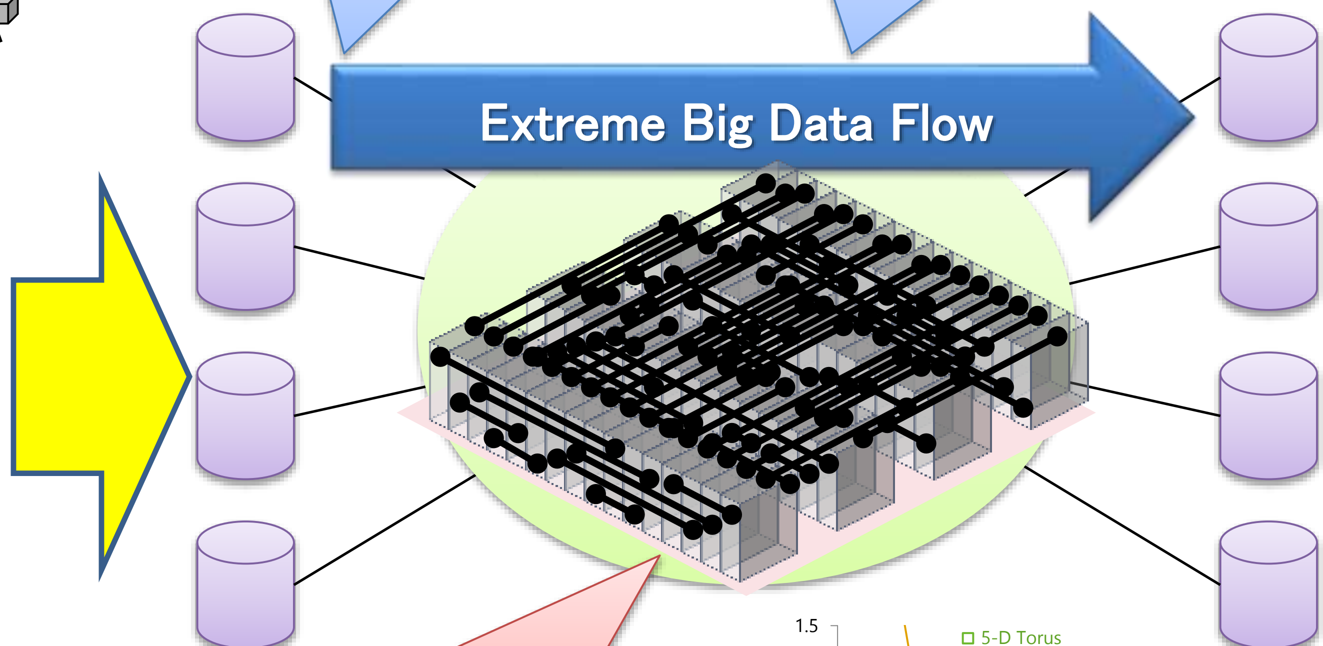
Typical Data Centers
-Poor scalability
- 1GbE + 10GbE
- TCP/IP basis



Supercomputers
- Dedicated to neighboring and uniform access

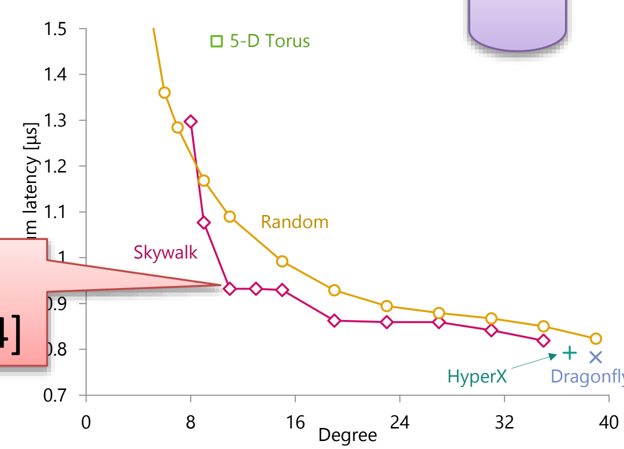
EBD non-uniform access

Low latency write/read ~10 μs for 4KB



Our low-jitter topology (<1us)
w/ random shortcuts

Our topology has better NW latency [IPDPS14]

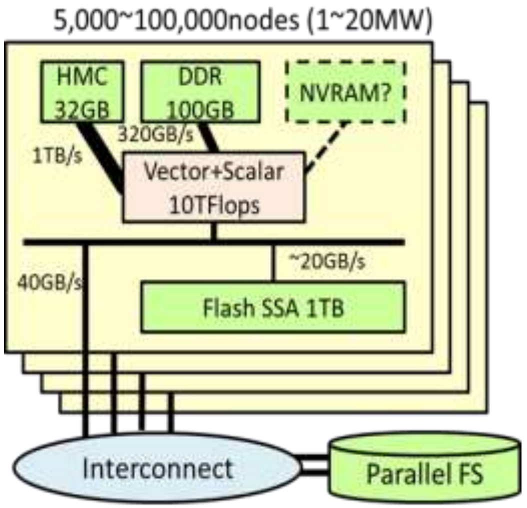


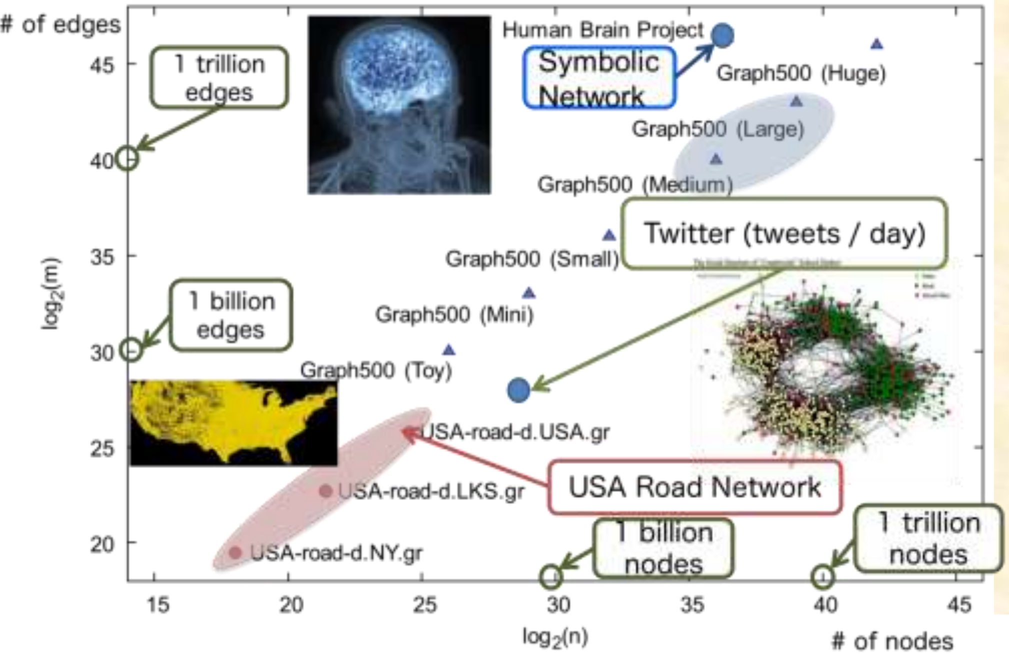
2. Extreme Big Data Algorithms

Graphs, Sorting, Clustering, Spatial Data…

JST CREST: Advanced Computing and Optimization Infrastructure for Extremely Large-Scale Graphs on Post Peta-Scale Supercomputers

- Innovative Algorithms and implementations
  - Optimization, Searching, Clustering, Network flow, etc.
- Extreme Big Graph Data for emerging applications
  - **2^{30} ~ 2^{42} nodes** and **2^{40} ~ 2^{46} edges**
  - **Over 1M threads** are required for real-time analysis
- Many applications on post peta-scale supercomputers
  - Analyzing massive cyber security and social networks
  - Optimizing smart grid networks
  - Health care and medical science
  - Understanding complex life system





- Example: Symbolic Network
  - **Human Brain Project** **http://www.humanbrainproject.eu/**
  - Understanding the human brain is one of the greatest challenges facing 21st century science
  - **89 billion neurons**(nodes)
  - **1 trillion connections**(edges)
  - Over 10^{17} bytes memory(storage) and 10^{18} Flops for brain simulator

The Graph500 – June 2014

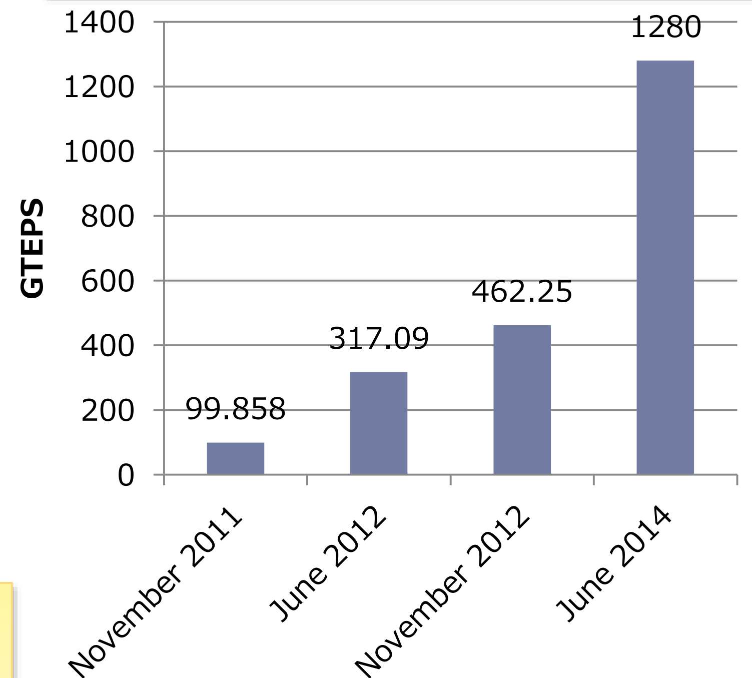
K Computer and TSUBAME 2.0 & 2.5

Graph500 ranking history for TSUBAME2.0 and 2.5

List	Rank	GTEPS	Implementation
November 2011	3	99.858	Top-down only
June 2012	4	317.09	GPU
November 2012	20	462.25	GPU
June 2014	12	1280	**Efficient hybrid**

*Every score is obtained using TSUBAME2.0 1366 nodes or TSUBAME 2.5 1024 nodes

BFS performance on TSUBAME2.0 and 2.5



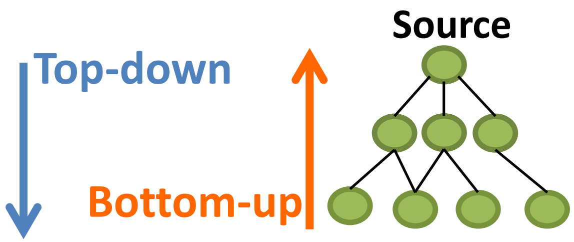
Graph500 ranking history for K Computer

List	Rank	GTEPS	Implementation
November 2013	4	5524.12	Top-down only
June 2014	1	17977.05	**Efficient hybrid**

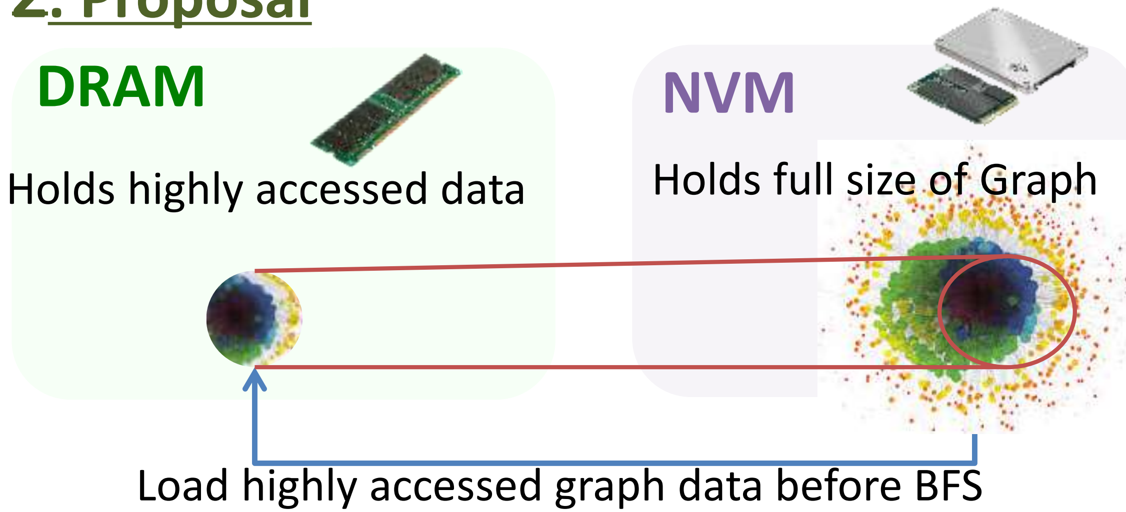
RIKEN Advanced Institute for Computational Science (AICS)'s K computer
No.1
on the Graph500 Ranking of Supercomputers with
17977.1 GE/s on Scale 40
on the 8th Graph500 list published at the International Supercomputing Conference, June 22, 2014.
Congratulations from the Graph500 Executive Committee
GRAPH 500
David A. Bader
Andrew Lumsdaine
Richard Murphy
Marc Snir

Large Scale Graph Processing Using NVM

1. Hybrid-BFS (Beamer'11)

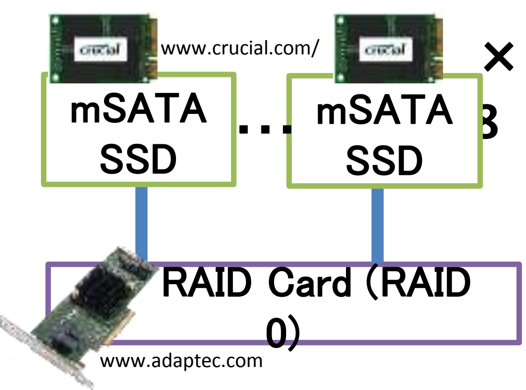


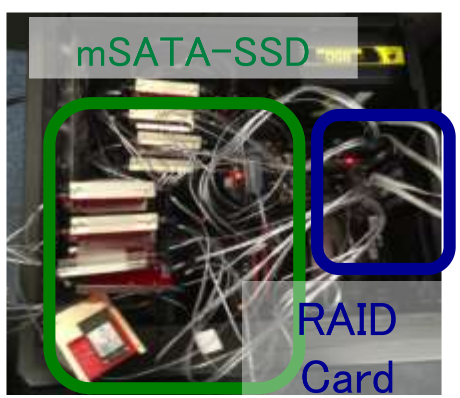
2. Proposal

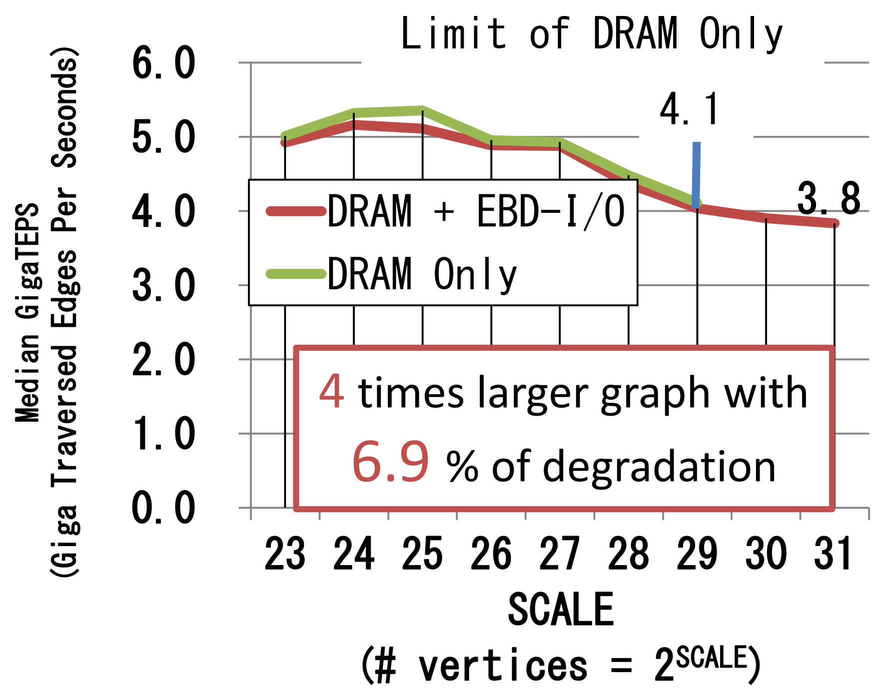


3. Experiment

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







The Green Graph500 list : Nov. 2013

http://green.graph500.org

- Measures power-efficiency using **TEPS/W** ratio
- Results on various systems such as **TSUBAME-KFC Cluster** and **Android mobiles**

TSUBAME-KFC
6.72 MTEPS/W (44.01 GTEPS)

Big Data category

Rank	MTEPS/W	Site	Machine	G500 rank	Scale	GTEPS	Nodes
1	**6.72**	Tokyo Institute of Technology	TSUBAME KFC	47	32	44.01	32
2	**5.41**	Forschungszentrum Julich (FZJ)	JUQUEEN	3	38	5848	16384
3	**4.42**	Argonne National Laboratory	DOE/SC/ANL Mira	2	40	14328	32768
4	**4.35**	Tokyo Institute of Technology	EBD-RH5885v2	96	30	3.67	1
5	**3.55**	Lawrence Livermore National Laboratory	DOE/NNSA/LLNL Sequoia	1	40	15363	65536

SONY Xperia-A-SO-04E
153 MTEPS/W (0.48 GTEPS)

Small Data category

Rank	MTEPS/W	Site	Machine	G500 rank	Scale	GTEPS	Nodes
1	**153.17**	Chuo University	GraphCREST-Xperia-A-SO-04E	143	20	0.478	1
2	**129.63**	Tokyo Institute of Technology	GraphCREST-NEXUS7-2013	141	20	0.534	1
3	**73.57**	University of Tsukuba	kitty6	58	25	17.207	1
4	**64.12**	Chuo University	GraphCREST-Tegra3	150	20	0.154	1
5	**53.82**	Chuo University	GraphCREST-Intel-NUC	124	23	1.082	1

Results : BFS Performance

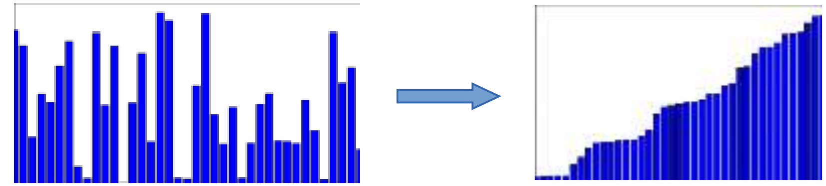
The Graph 500 2014 June
DRAM + NVM model

	MEM-CREST Node #2 (Supermicro 2027GR-TRF)	GraphCrest Node #1	EBD-RH5885v2 (Huawei Tecal RH5885 V2)
DRAM	128 GB	256 GB	1024 GB
NVM	ioDrive2 1.2 TB × 2	EBD-I/O 2TB × 2	• Tecal ES3000 800GBx2,1.2TBx2 • EBD-I/O 4TB × 2
SCALE (Total Data Size)	30 (500GB)	31 (1TB)	33 (4TB)
GTEPS	7.98	13.80	3.11
MTEPS / W	28.88	35.21	3.42

~ x6 better than Nov. 2013 #1 !

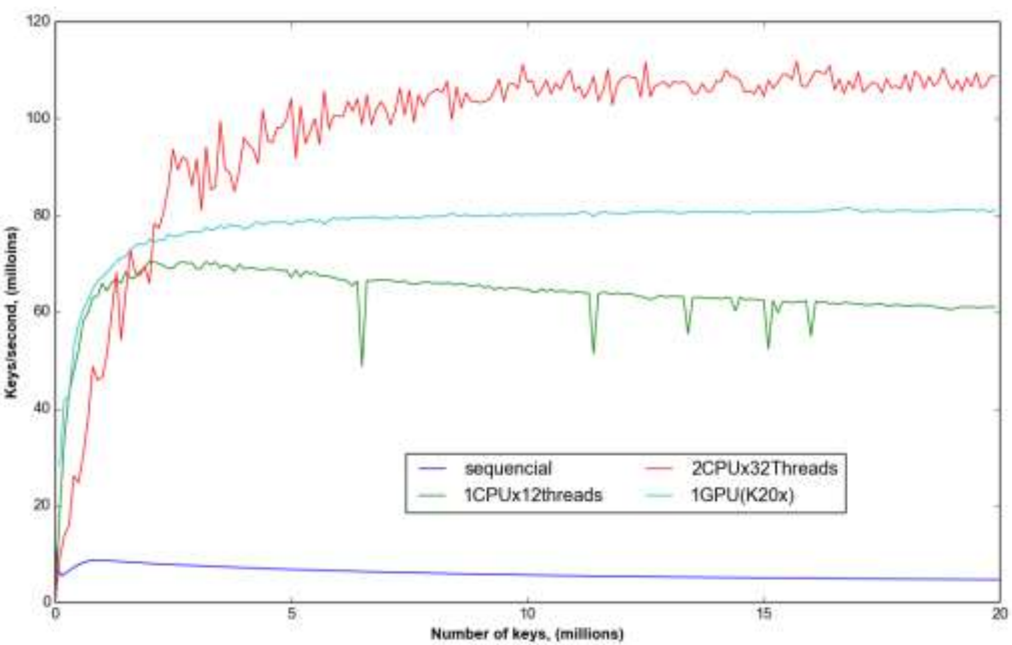
Sorting for EBD

using single node to the utmost capacity



- Sorting long/variable length keys (strings)
- Implementations for GPUs and multi /many-core CPUs
- Hybrid parallelization scheme combining data-parallel and task-parallel stages
- Trimming keys to reduce host-to-device communication overheads
- Up to **100 million string keys per second**

Sorting
One of the fundamental primitives
Extremely well studied
Variety of data types, sizes, hardware architectures and characteristics
leave lots of space for improvement.



MSD radix sort

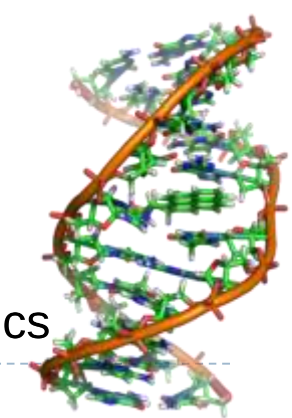
apple
apricot
banana
kiwi

Don't have to examine all characters

Processing textual data (e.g. corpus linguistics)

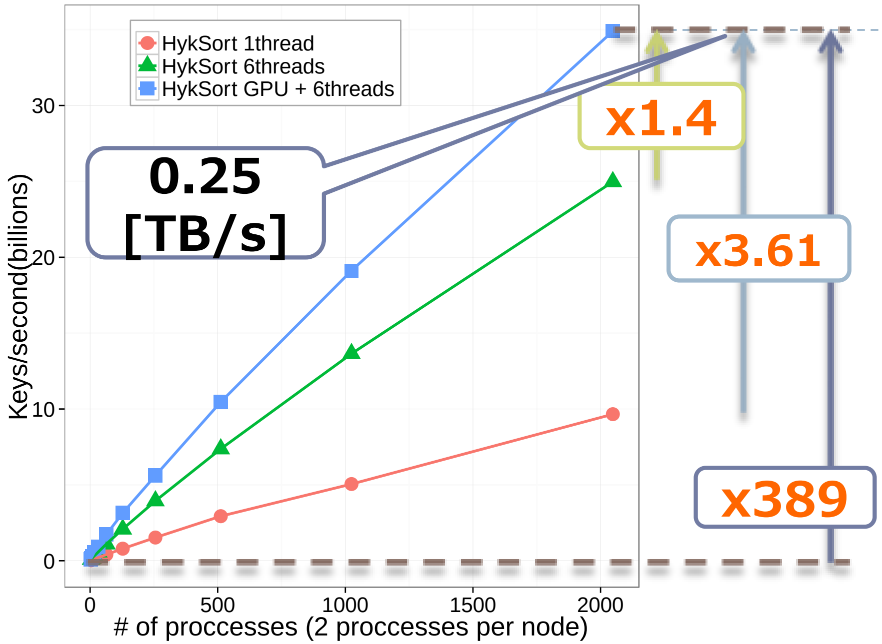
High efficiency on small alphabets

Computational genomics (A,C,G,T)



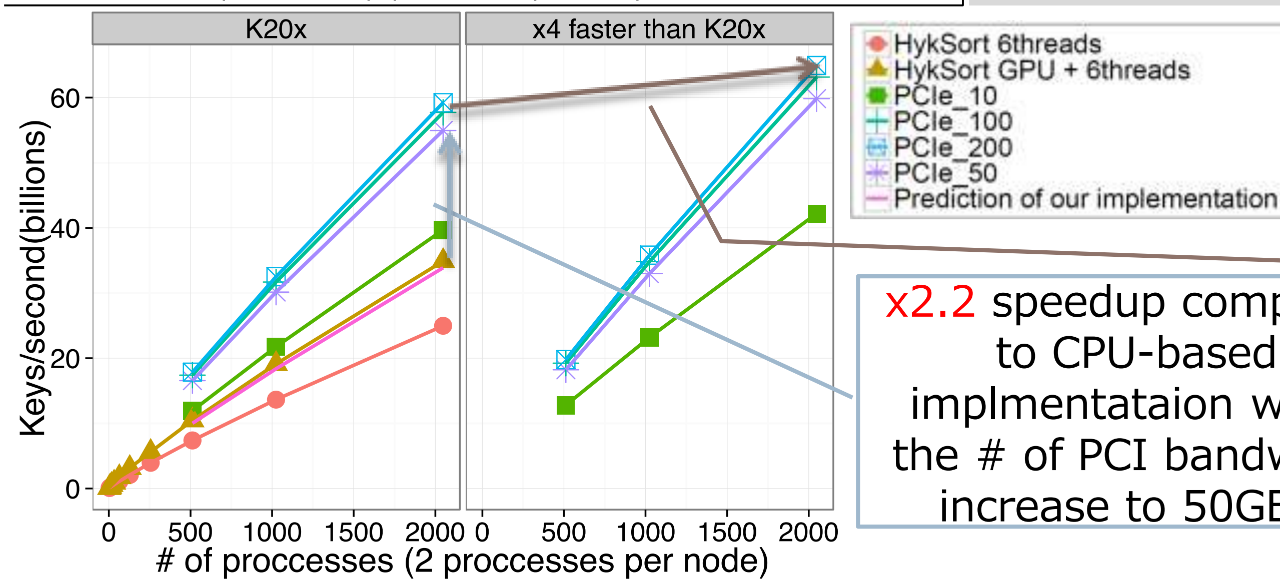
Sorting for EBD

Plugging in GPUs for large-scale sorting



- **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 and each node has 2GB 64bit integer
- 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 implmentataion when the # of PCI bandwidth increase to 50GB/s

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

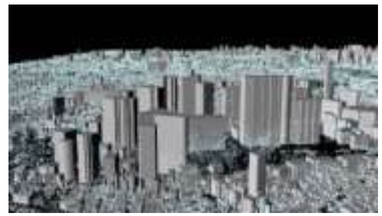
3. Extreme Big Data Programming, DSLs, Libraries, and APIs

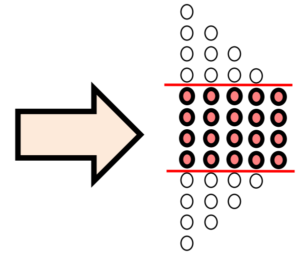
Existing Abstractions made Extreme (MapReduce, Pregel) + New Abstractions for Extreme (Communication Reducing Algorithms)

Software Technology that Deals with Deeper Memory Hierarchy in Post-petascale Era

JST-CREST project, 2012-2018, PI Toshio Endo

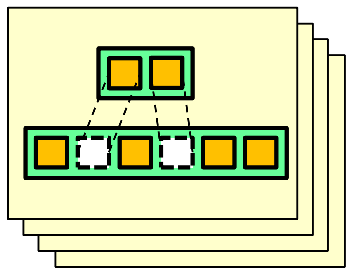
Comm/BW reducing algorithms

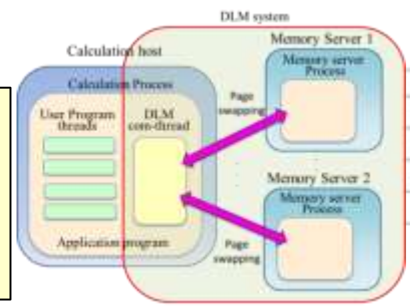




+

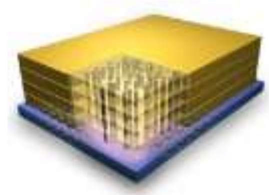
System software for mem hierarchy mgmt





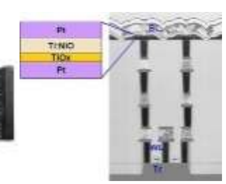
+

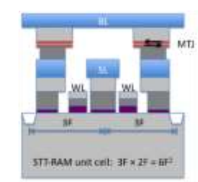
HPC Architecture with hybrid memory devices



HMC, HBM

O(GB/s) Flash

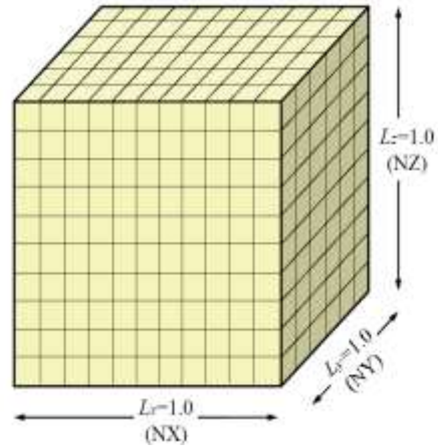




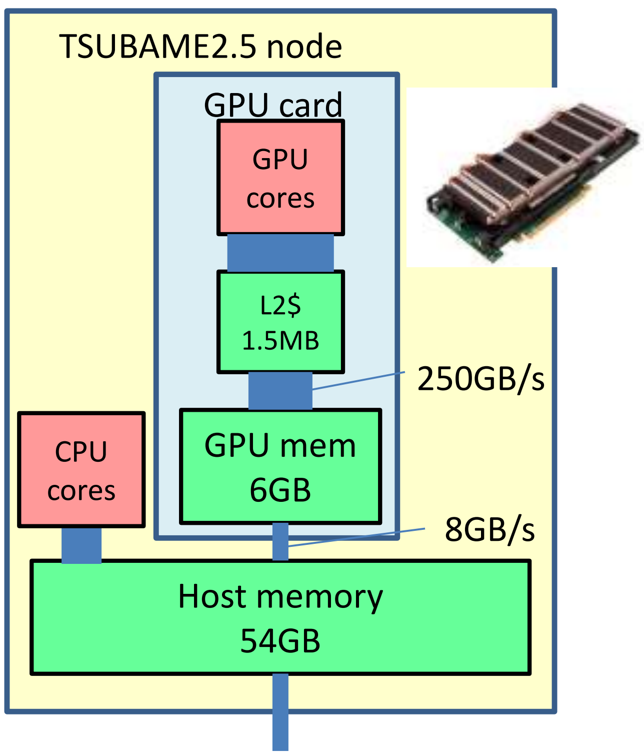
Next-gen NVM

<u>*Target*</u>: Realizing ***extremely Fast&Big simulations*** of {O(100PF/s) or O(10PB/s)} & O(10PB) around 2018

Supporting Larger domains than GPU device memory for Stencil Simulations



>>

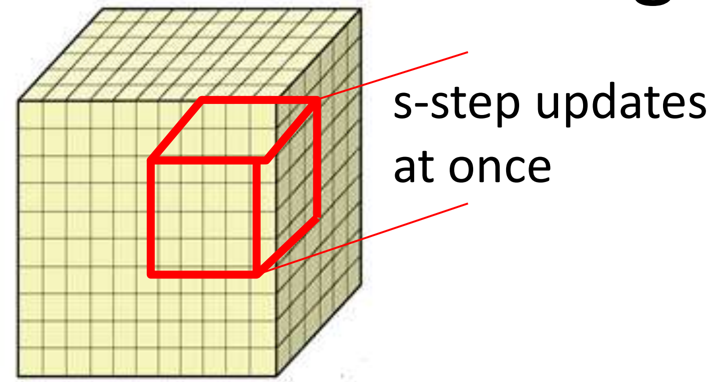


Caution: Simply "swapping out" to larger host memory is disastrously slow
PCIe traffic is too large!

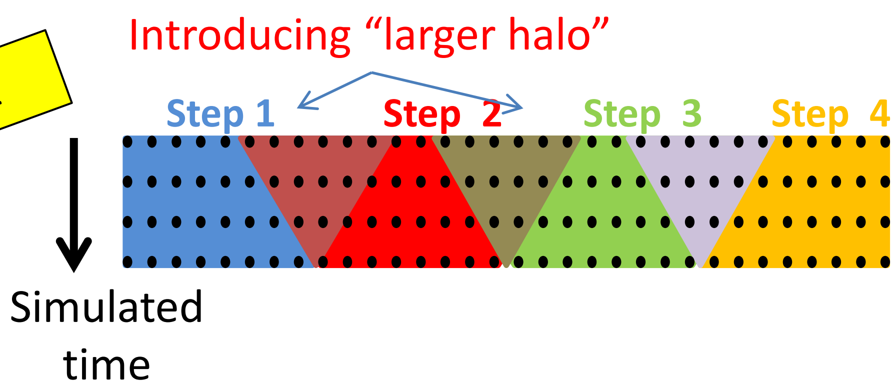
Keys are "Communication Avoiding & Locality Improvement" Algorithms

Temporal Blocking (TB) for Comm. Avoiding

- Performs **multiple updates on a small block**, before proceeding to the next block
  - Originally proposed to improve cache locality [Kowarschik 04] [Datta 08]

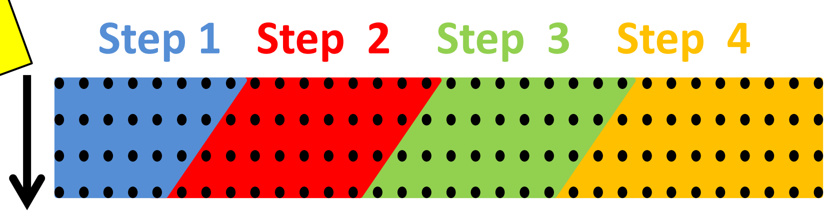


Version 1



Redundant computation is introduced due to data dependency with neighbor

Version 2



Redundancy can be removed when blocks are computed sequentially [Demmels 12]

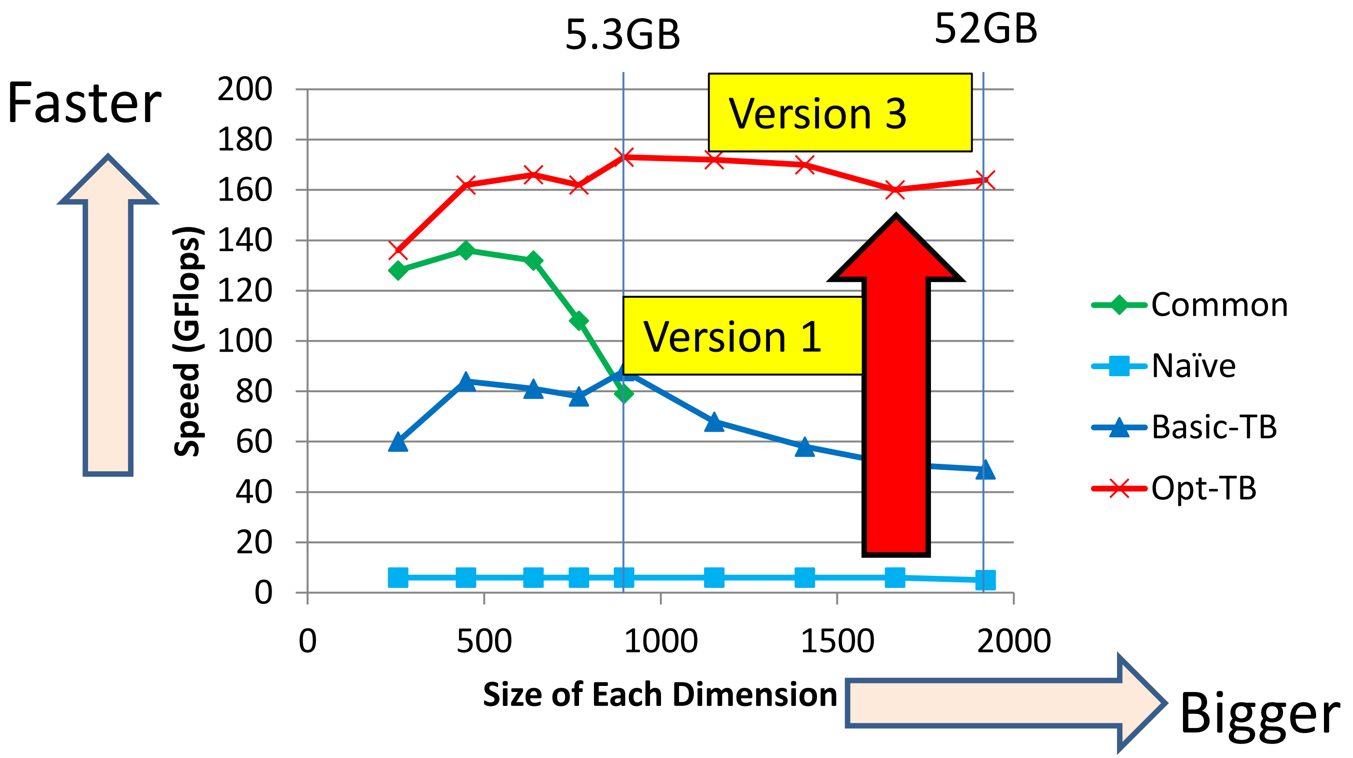
Version 3

Multi-level TB to reduce both

- PCIe traffic
- device memory traffic

Single GPU Performance

3D 7point stencil on a K20X GPU (6GB GPU mem)



- With optimized TB, 10x larger domain size is successfully used with little overhead!!!

→ A step towards extremely fast&big simulations

Problem: Programming Cost

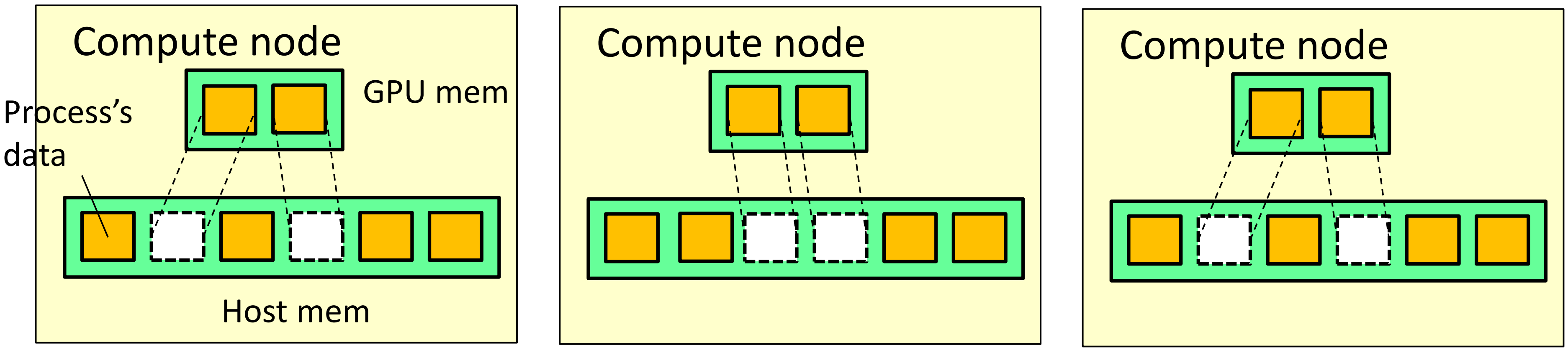
- Communication reducing algorithms efficiently support larger domains
- Programming cost is the issue
  - Complex loop structure, complex border handling

- Reducing programming cost by using system software supporting memory hierarchy
  - HHRT (Hybrid Hierarchical Runtime)
  - Physis DSL, by Maruyama, RIKEN

Memory Hierarchy Management with Runtime Libraries

HHRT (Hybrid hierarchical RT) is for GPU supercomputers and MPI+CUDA user applications

- HHRT provides MPI and CUDA compatible APIs
- # of MPI processes > # of GPUs
  - Several processes share a GPU



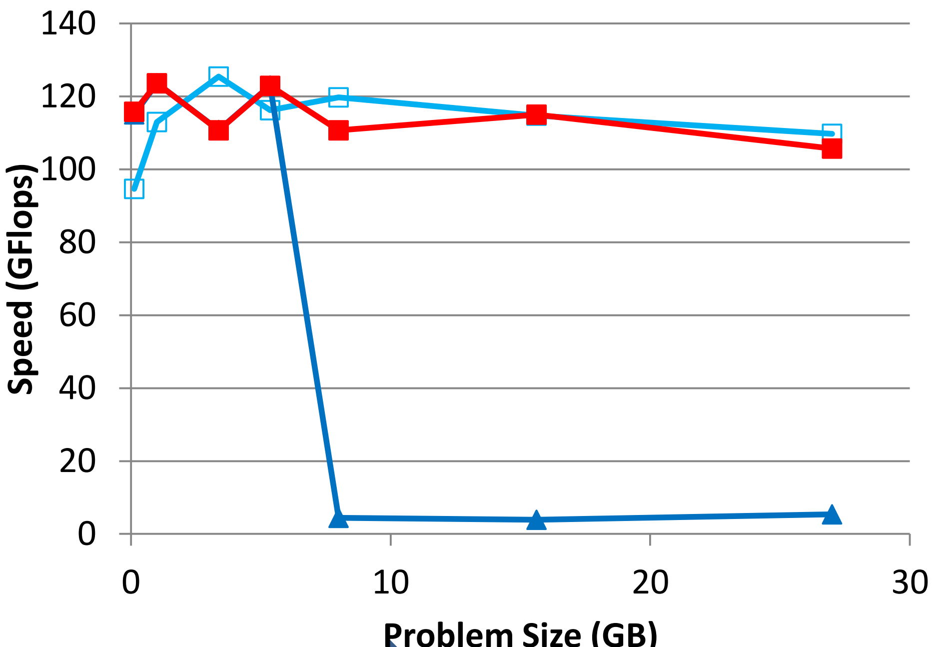
- HHRT supports memory swapping between GPU and host mem at granuarity of processes
- Similar to NVIDIA UVM, but works well with communication reducing algorithms

HHRT Comm. Reducing Results

Faster

Beyond GPU memory efficient execution w/ moderate programming cost

3D 7point stencil on a single K20X GPU

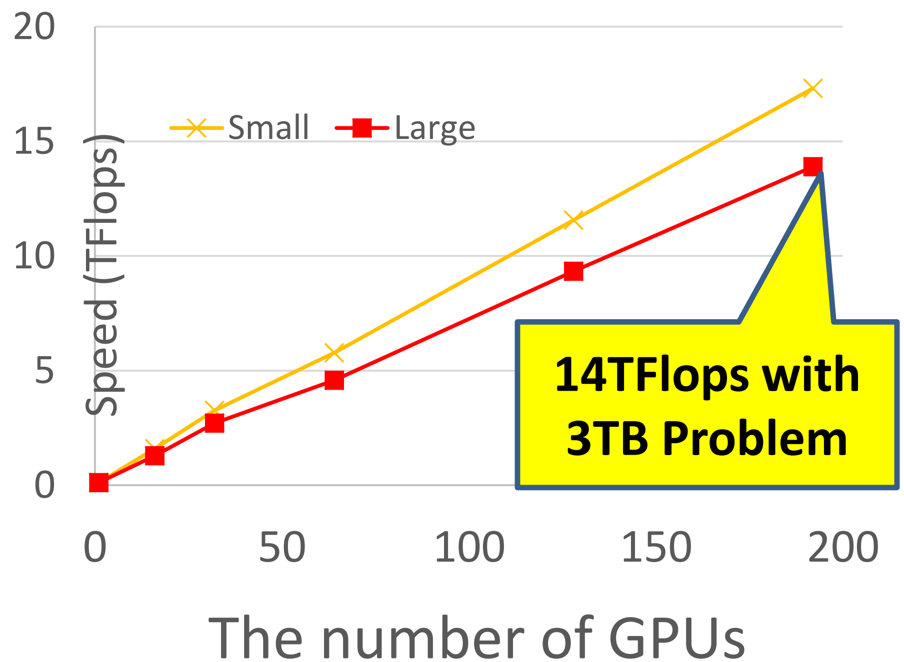


Larger

Weak scalability on TSUBAME2.5

Small: 3.4GB per GPU

Large: “16GB” per GPU (>6GB!)

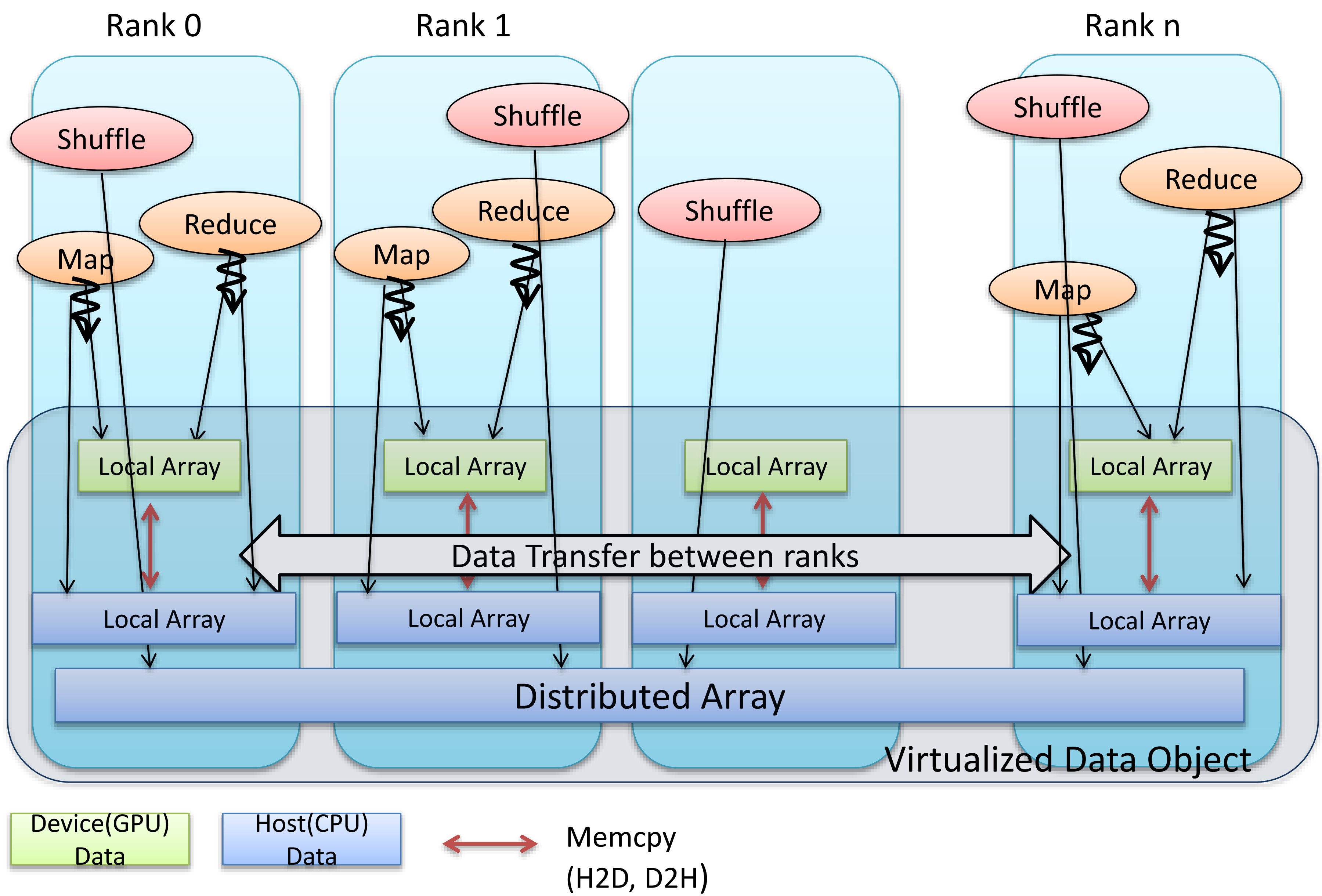


Hamar (Highly Accelerated Map Reduce) [Matsuoka Team- Sato, Shirahata et.al.]

- 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.

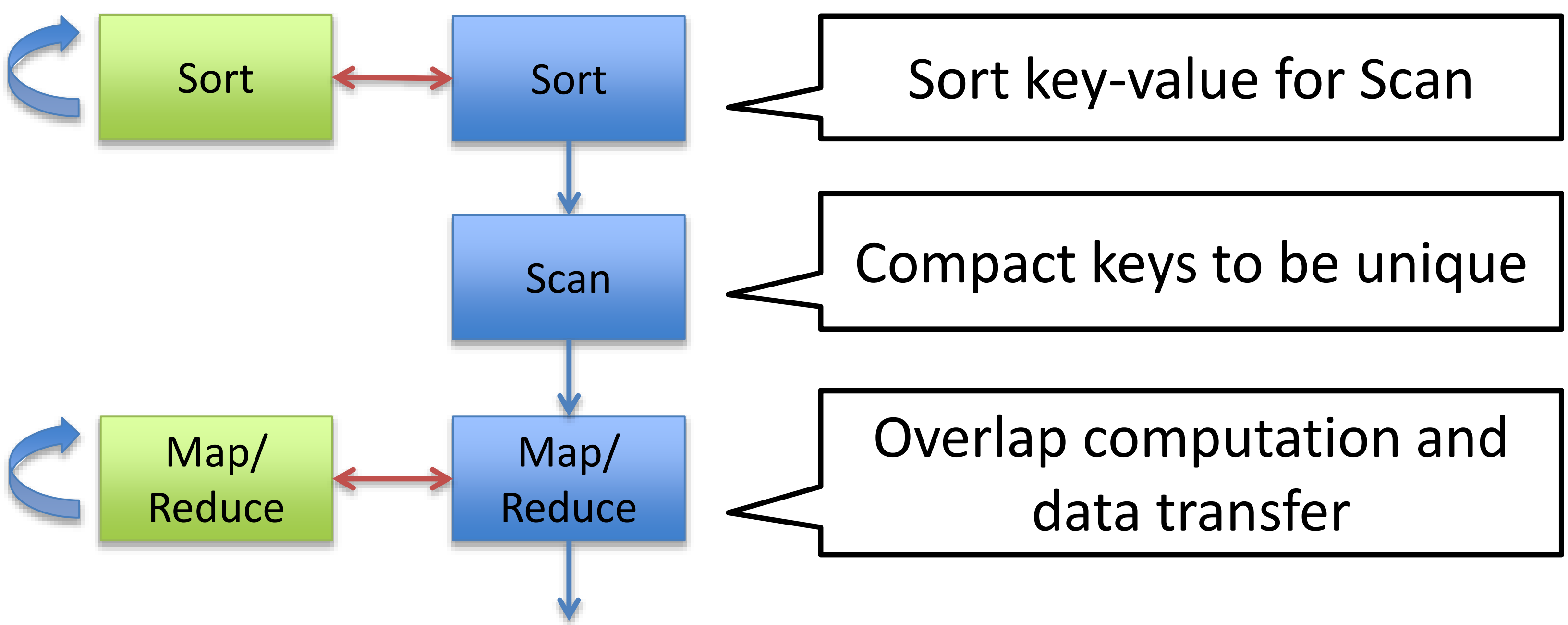
- Features
  - Object-oriented
    - C++-based implementation
    - Easy adaptation to modern commodity many-core accelerator/Flash devices w/ SDKs
      - CUDA, OpenNVM, etc.
  - Weak-scaling over 1000 GPUs
    - TSUBAME2
  - Out-of-core GPU data management
    - Optimized data streaming between device/host memory
    - GPU-based external sorting
  - Optimized data formats for many-core accelerators
    - Similar to JDS format

Hamar Overview



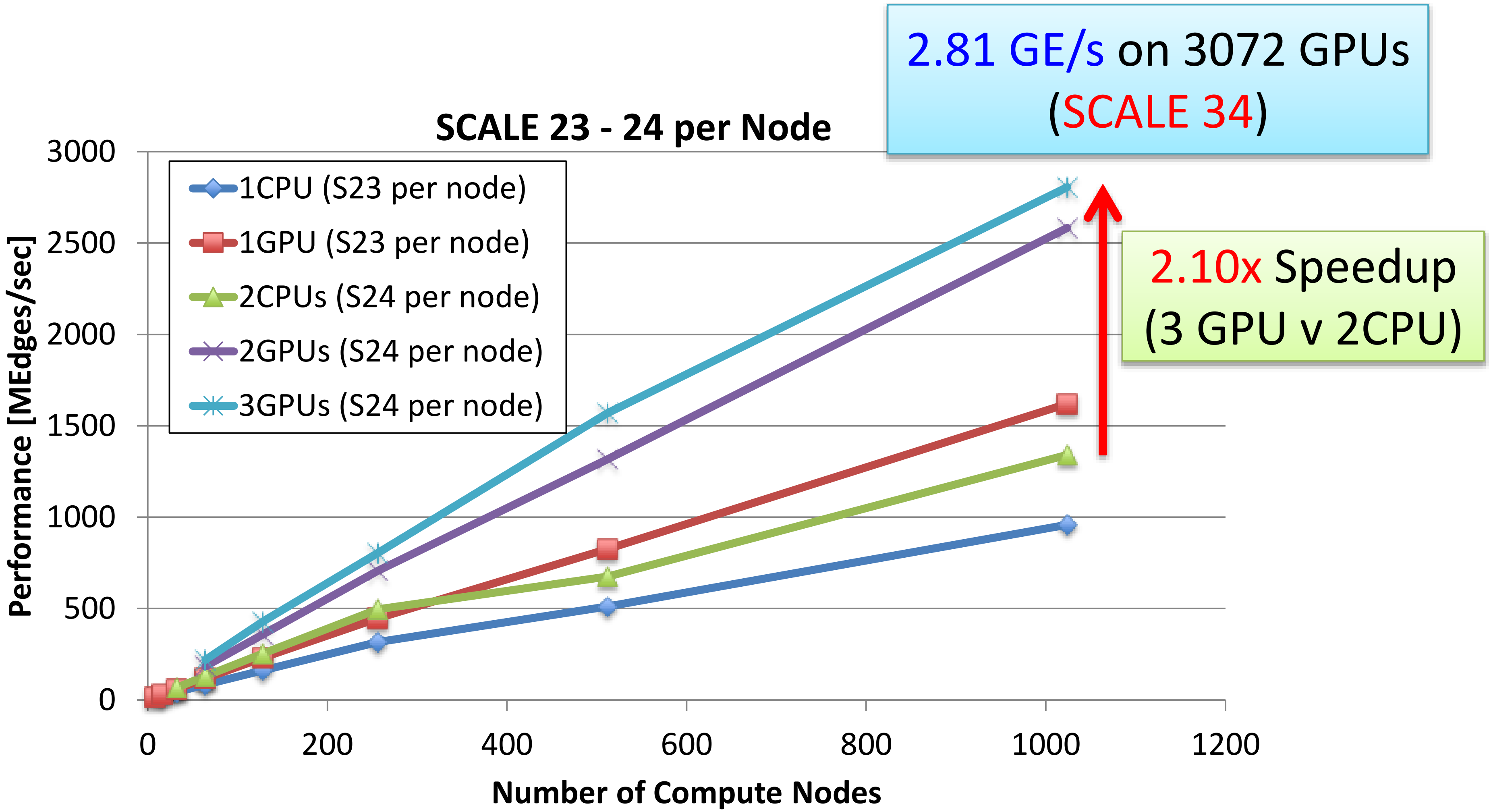
Map/Reduce Implementation

- **Optimizations for GPU accelerators**
  - Assign a warp (32 threads) per key for avoiding warp divergence in Map/Reduce
  - Overlapping computation on GPU and data transfer between CPU and GPU
  - Out-of-core GPU Sorting Algorithm



Weak Scaling Performance

- PageRank application on TSUBAME 2.5
- Data size is larger than GPU memory capacity



Existing Graph Analytics Libraries

- **Single Node**
  - igraph (R package)
  - GraphLab/GraphChi (Carnegie Mellon University and Start-up, C++)
- **Distributed Systems**
  - **MPI-based libraries**
    - PBGL2 (Parallel Boost Graph Library, C++) [Gregor, Oopsla 2005]
    - ParMetis (dedicated for parallel graph partitioning, C+), etc
  - **Hadoop-based libraries**
    - Apache Giraph (Pregel Model, Java)
    - PEGASUS (Generalized Iterative Sparse Matrix Vector Mult CMU), etc
  - **GPS** (Graph Processing System - Pregel Model, Stanford, Java + NIO)
  - **Distributed Graphlab** (CMU)

ScaleGraph Library

- Many existing graph analytics libraries
  - **Single Node**
    - **igraph, GraphLab/GraphChi, …**
  - **Distributed Systems**
    - **Apache Giraph, PBGL2, PEGASUS, GPS, Distributed Graphlab, …**
- However, they are not optimized for the state of the art hardware.
  - High-speed network, Multi-core CPUs, NVRAM
- Create an open source **Highly Scalable Large Scale Graph Analytics Library** beyond the scale of billions of vertices and edges on Distributed Systems
- Grand Challenge: Peta byte scale graph analysis
  - 2^{42} vertices and 2^{46} edges (1.1PB) using 100TB DRAM and 5PB NVRAM.

URL: http://www.scalegraph.org/

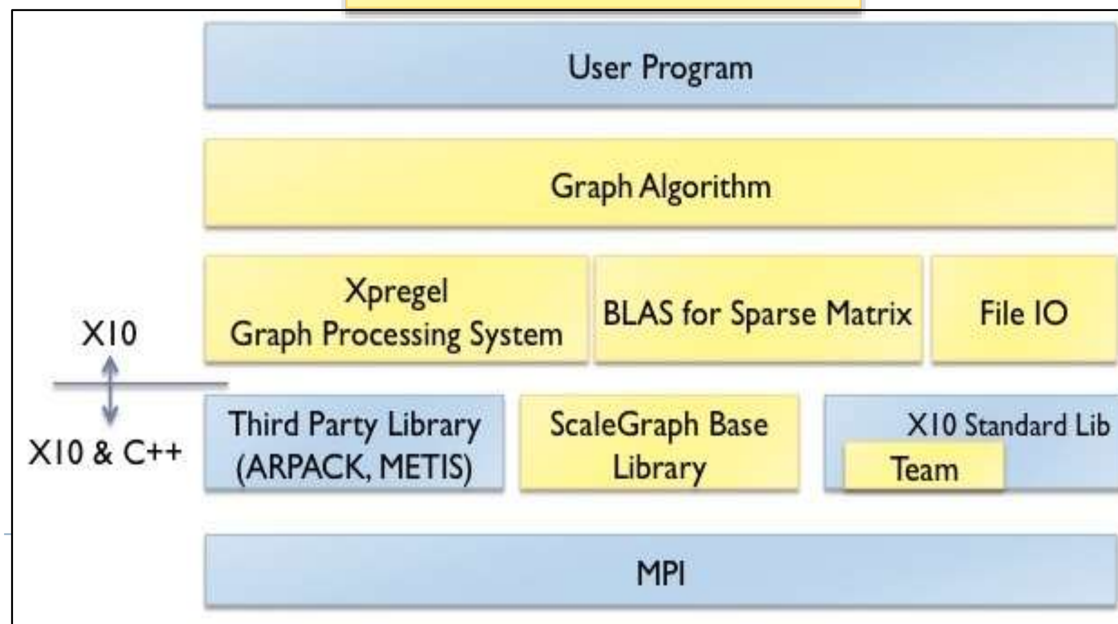
ScaleGraph Architecture Design

- Based on our extended X10
  - X10 is a new parallel distributed programming language.
- Fully utilizing MPI collective communication
- Native support for hybrid (MPI and multi-threading) parallelism
- XPregel: Graph processing framework
  - Optimized message communication and Simple API
- Rich graph algorithms

Supported algorithm

PageRank
Spectral Clustering
Degree Distribution
Betweenness Centrality
Degree of Separation(HyperANF)
Strongly-connected coponent
Maximum Flow
Single Source Shortest Path
BFS ...

Software stack

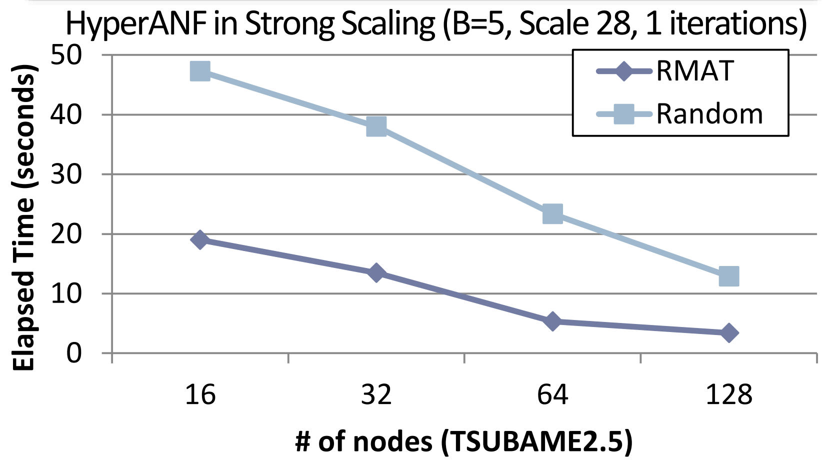


XPregel – X10-based Pregel-like Graph Programming System for convergent architectures

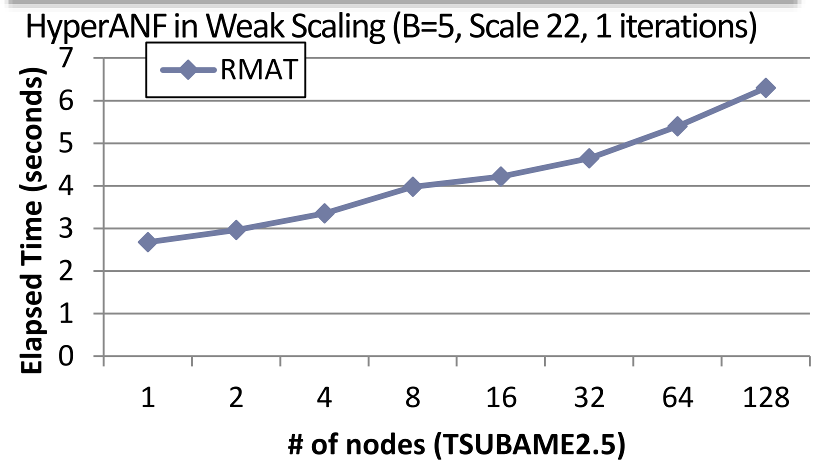
- XPregel optimizations on supercomputers
  1. Utilize MPI collective communication.
  2. Avoid serialization, which enables utilizing fast supercomputer interconnects
  3. Destination of messages computed by a simple bit manipulation thanks to vertex id renumbering.
  4. Optimized message communication when all vertices send the same message to all the neighbor vertices.
  5. Simple API in X10 language.

Performance Evaluation

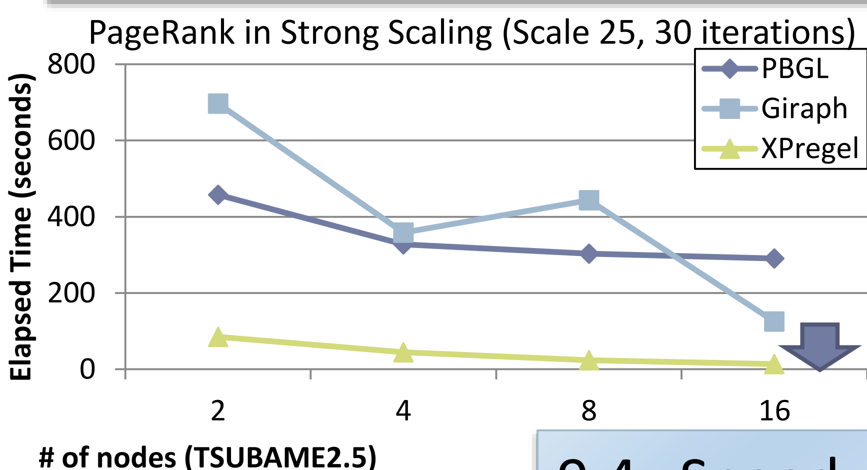
Degree of Separation



Degree of Separation

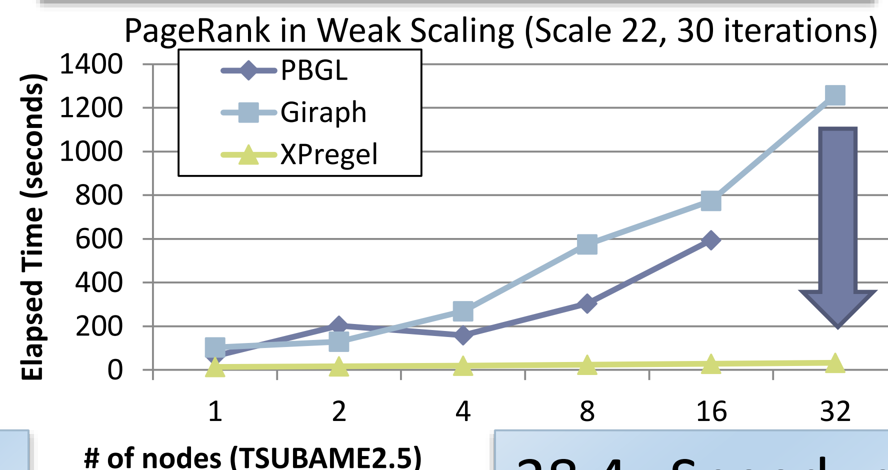


ScaleGraph vs. Giraph, PBGL



9.4x Speedup

ScaleGraph vs. Giraph, PBGL



38.4x Speedup

Performance Summary for ScaleGraph 2.2

- **Artificial big graph that follows various features of Social Network**
  - **Largest data** : **4.3 billion** vertices and 68.7 billion edges (RMAT : Scale 32, 128 nodes)
  - PageRank : 16.7 seconds for 1 iteration
  - HyperANF (B=5) = 71 seconds for 1 iteration
- **Twitter Graph** (**0.47 billion** vertices vertices and 7 billion vertices – around Scale 28.8)
  - PageRank (128 nodes): 2.56 seconds for 1 iteration
  - Spectral Clustering (128 nodes) : 1,839 seconds
  - HyperANF (B=5, 128 nodes): 28 seconds for 1 iteration
  - Degree Distribution (128 nodes): 128 seconds
- We will support out-of-core processing with external memory (NVRAM) in the future

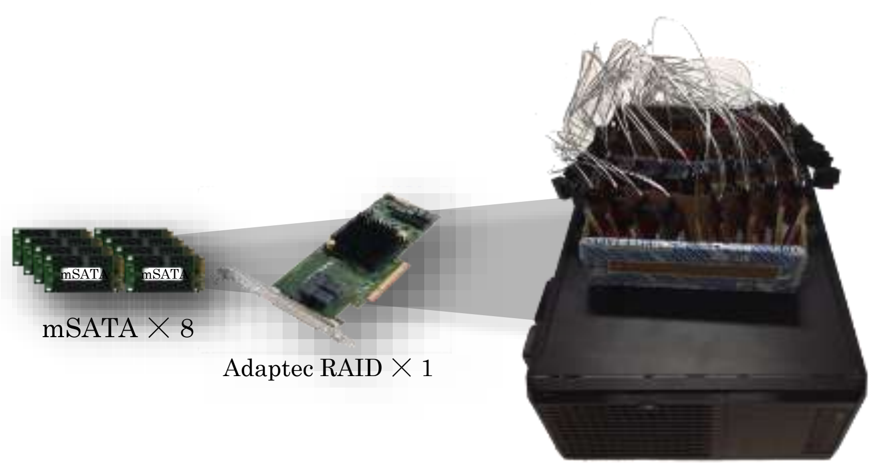
* Hyper ANF is an algorithm of degree of separation

4. Extreme Big Data – System Software Software, Distributed Objects

Distribution, Instrumentation, Scaling, Resilience, Bandwidth Reduction, , …

Extreme scale I/O for burst buffers [w/LLNL, CCGrid2014 Best Paper]

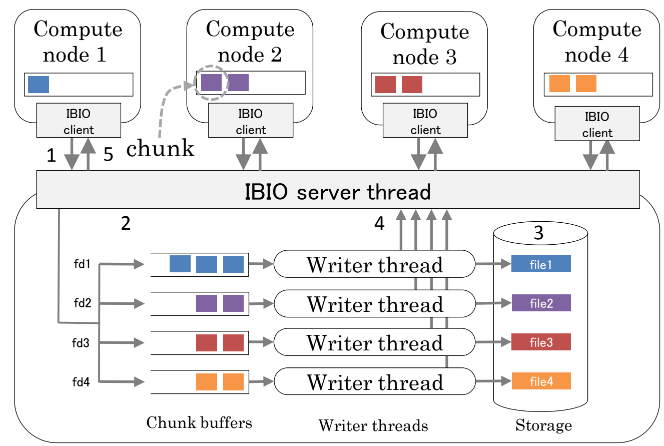
- Provide POSIX I/O interfaces
  - open, read, write and close
  - Client can open any files on any servers
    - open(“hostname:/path/to/file”, mode)
- IBIO use ibverbs for communication between clients and servers
  - Exploit network bandwidth of infiniBand



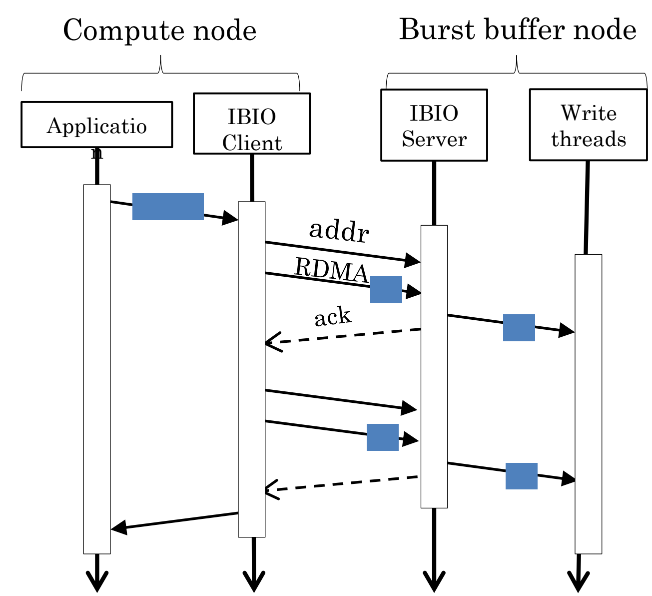
EBD I/O

SSD	Crucial m4 msata 256GB CT256M4SSD3 (Peak read: 500MB/s, Peak write: 260MB/s)
SATA converter	KOUTECH IO-ASS110 mSATA to 2.5’ SATA Device Converter with Metal Fram
RAID Card	Adaptec RAID 7805Q ASR-7805Q Single

IBIO write



IBIO write: four IBIO clients and one IBIO server



Extreme scale resilience modeling

- To find out the best checkpoint/restart strategy for systems with burst buffers, we model checkpointing strategies

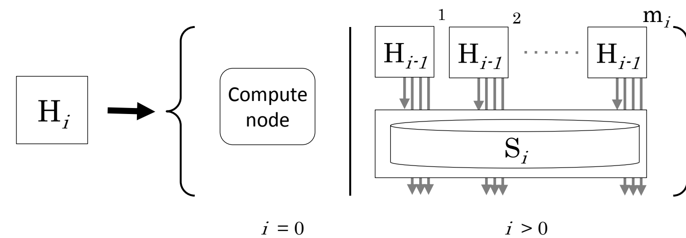
C/R strategy model

$$O_i = \begin{cases} C_i + E_i & \text{(Sync.)} \\ I_i & \text{(Async.)} \end{cases} \qquad L_i = C_i + E_i$$

$$C_i \text{ or } R_i = \frac{\text{< C/R date size / node >} \times \text{<\# of C/R nodes per } S_i^* \text{ >}}{\text{< write perf. (} w_i \text{) > or <read perf. (} r_i \text{) >}}$$

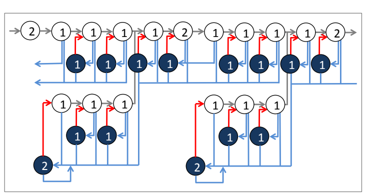
+

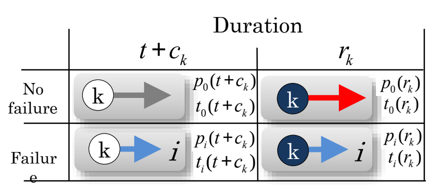
Recursive structured storage model



Storage Model : $H_N \{m_1, m_2, \ldots, m_N\}$

MLC model [2]





t : Interval
c_c : c-level checkpoint time
r_c : c-level recovery time
l_i : i-level checkpoint time

$$p_0(T) = e^{-\lambda T}$$
$$t_0(T) = T$$
$$p_i(T) = \frac{\lambda_i}{\lambda}(1 - e^{-\lambda T})$$
$$t_i(T) = \frac{1 - (\lambda T + 1) \cdot e^{-\lambda T}}{\lambda \cdot (1 - e^{-\lambda T})}$$

$p_0(T)$: No failure for T seconds
$t_0(T)$: Expected time when $p_0(T)$

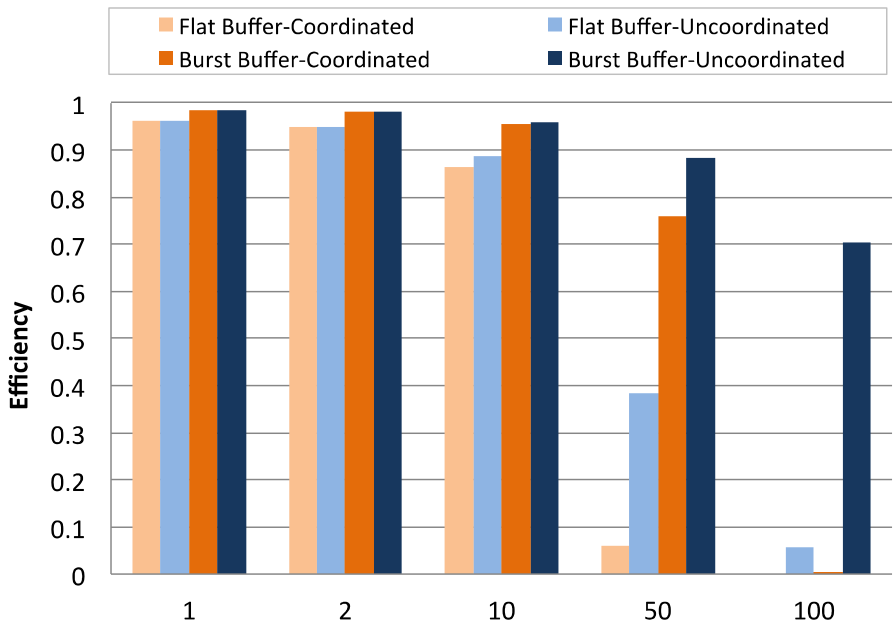
$p_i(T)$: i - level failure for T seconds
$t_i(T)$: Expected time when $p_i(T)$

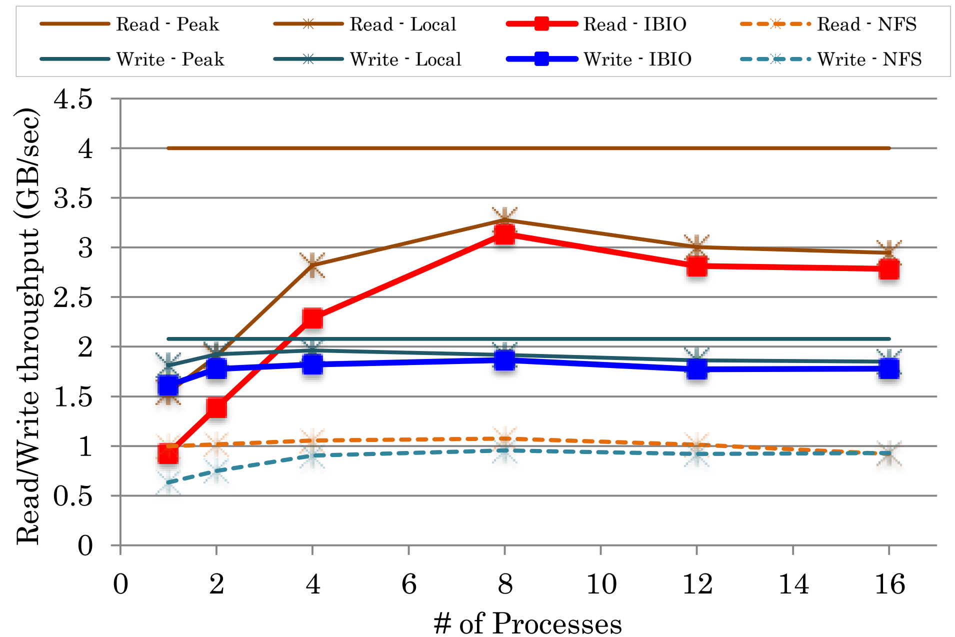
Efficiency

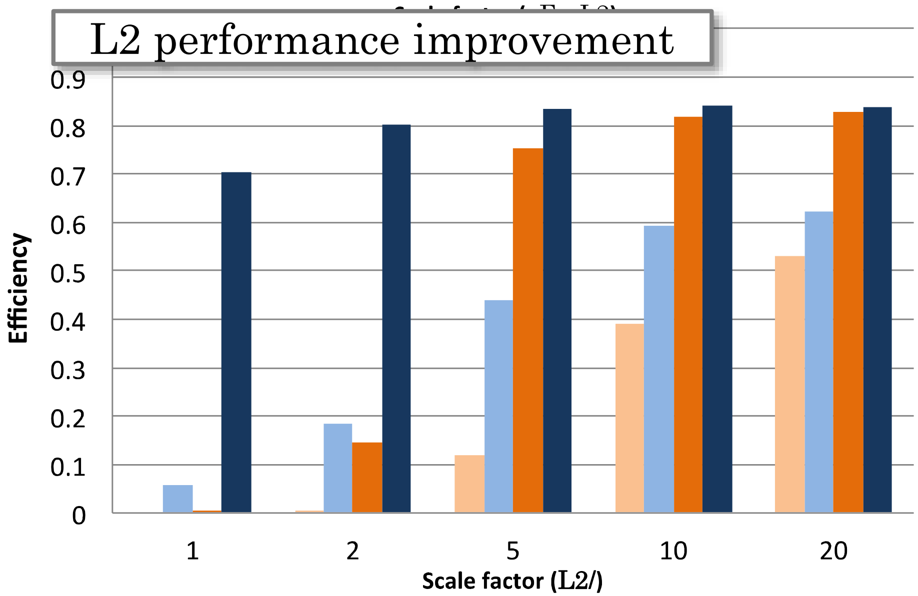
(Fraction of time an application spends only in useful computation)

[2] Kento Sato, Adam Moody, Kathryn Mohror, Todd Gamblin, Bronis R. de Supinski, Naoya Maruyama and Satoshi Matsuoka, "Design and Modeling of a Non-blocking Checkpointing System", SC12

EBD I/O performance and the overall efficiency





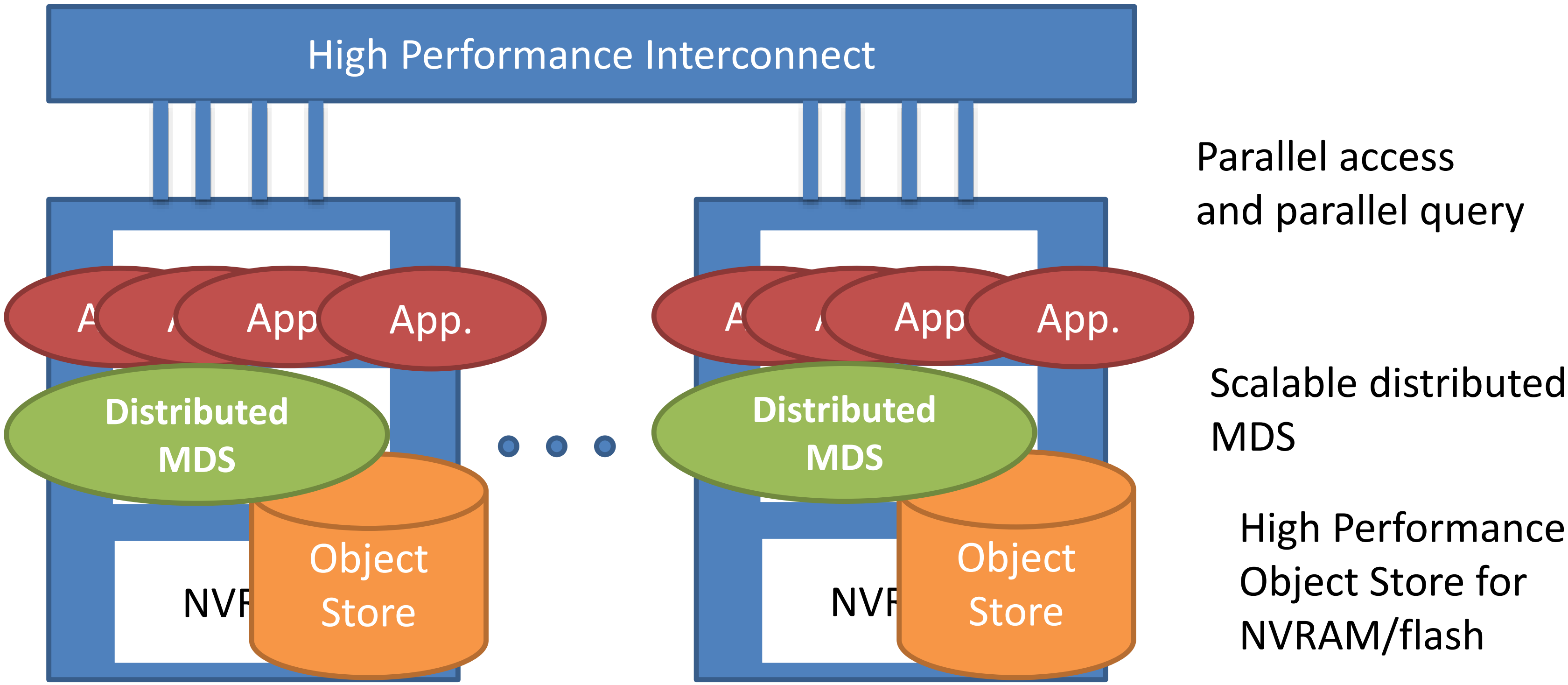


Increasing the performance of the PFS does impact system efficiency

L2 C/R overhead is a major cause of degrading efficiency, so reducing level-2 failure rate and improving level-2 C/R is critical on future systems

R&D of EDB Distributed Object Store (co-PI: Osamu Tatebe, U-Tsukuba)

- Key design issues for Scaled-out IOPS and I/O bandwidth
  - Scalable distributed MDS (1M IOPS Object Creation)
  - High Performance local object store
  - Efficient parallel access (100 TB/s) and parallel query

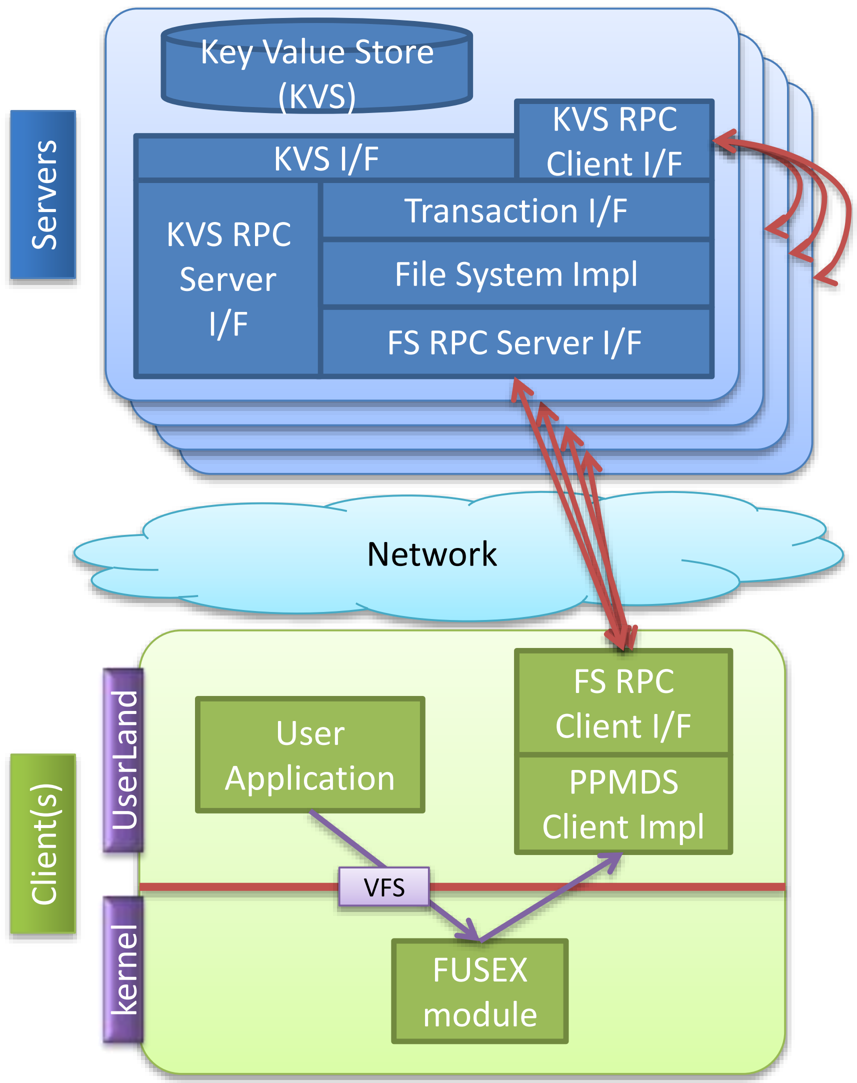


PPMDS – distributed Scale-out MDS
[Hiraga & Tatebe, U-Tsukuba]

Target: Scale-out distributed MDS for O(1M) IOPS

Problems:
- Single MDS **does not scale out**
- Parallel file creations in the same directory **require lock**

Features of PPMDS:
- **Distributed MDS**
- **Lock is not required** for parallel file creations in the same directory by data management of parent inumber and entry name
- **Nonblocking distributed transaction based on Dynamic Software Transaction Memory (DSTM)**

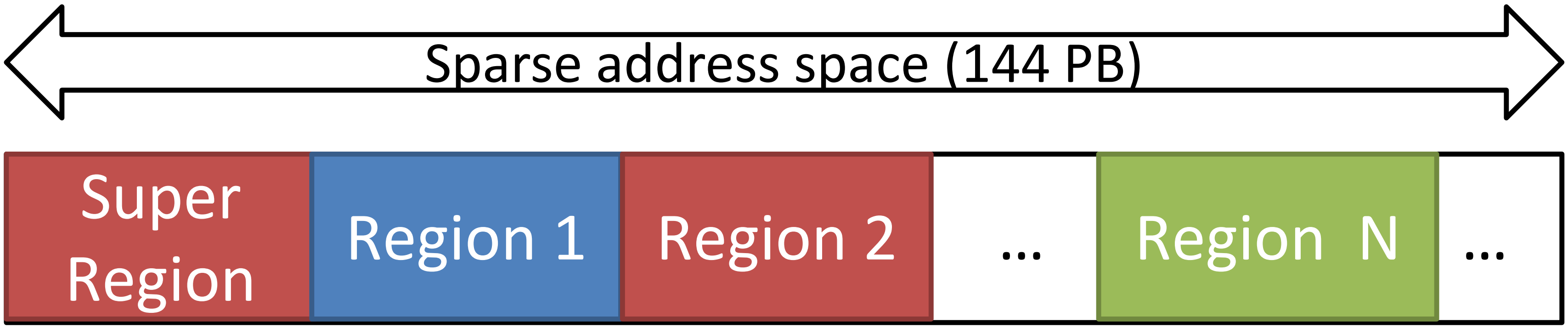


PPMDS – distributed Scale-out MDS Preliminary Performance

- GIGA+ [Swapnil Patil et al. FAST’11]
  - Incremental directory partitioning
  - Independent locking in each partition
- skyFS [Jing Xing et al. SC’09]
  - Performance improvement during directory partitioning in GIGA+
- Lustre
  - MT scalability in 2.X
  - Proposed clustered MDS
- PPMDS [Our JST CREST R&D]
  - Shared-nothing KV stores
  - Nonblocking software transactional memory (**No lock**)

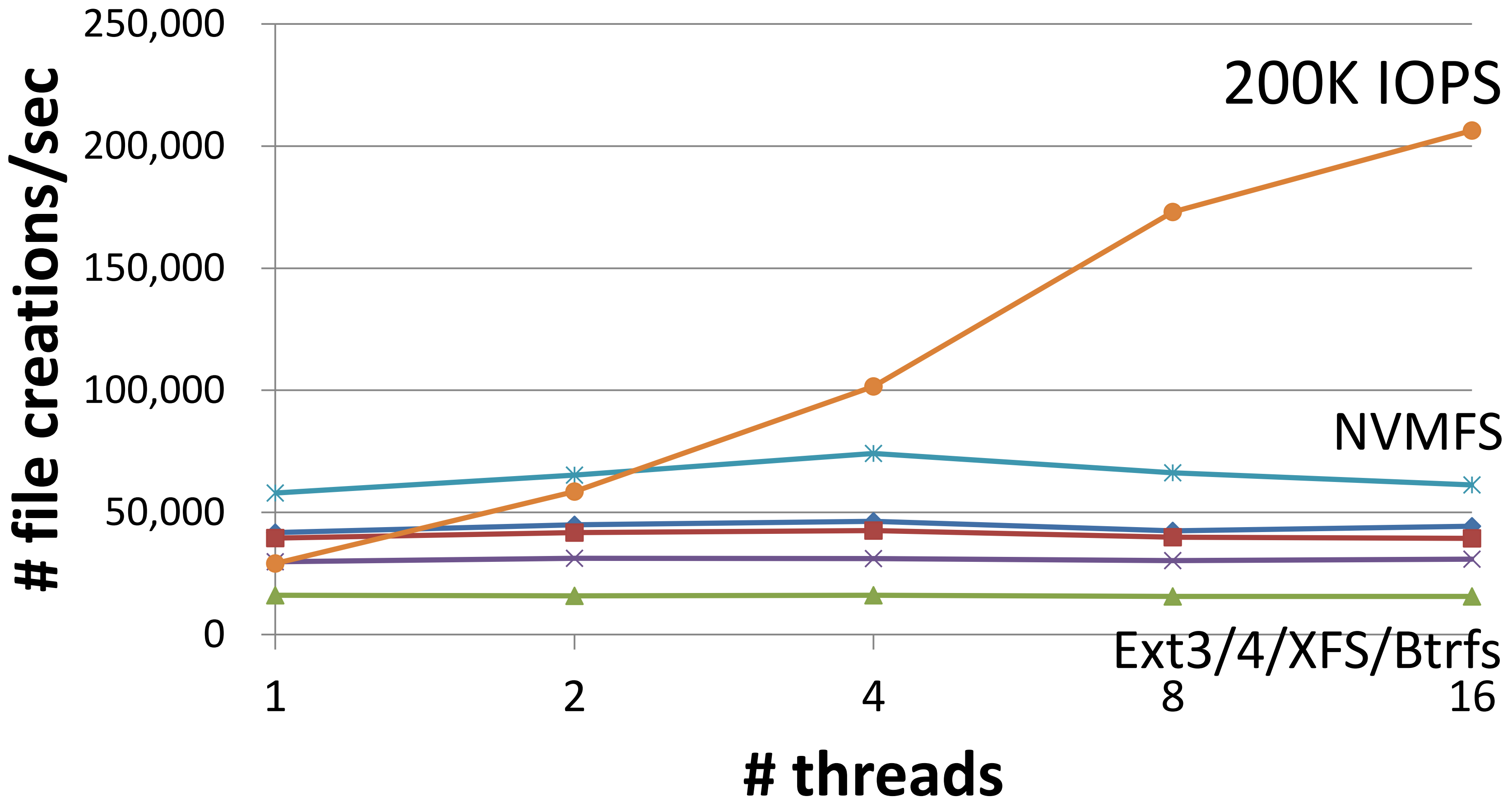
	IOPS (file creates per sec)	#MDS (#core)
GIGA+	98K	32 (256)
skyFS	100K	32 (512)
Lustre 2.4	80K	1 (16)
PPMDS	270K	15 (240)

Design of Object Store for NVM [Takatsu and Tatebe, U-Tsukuba]



- Simplest object store format
  - Fixed size of region (e.g. 2 TB)
    - Large enough to avoid indirect accesses
  - No directory entry
- Reserved base region number assignment reduces the number of locks

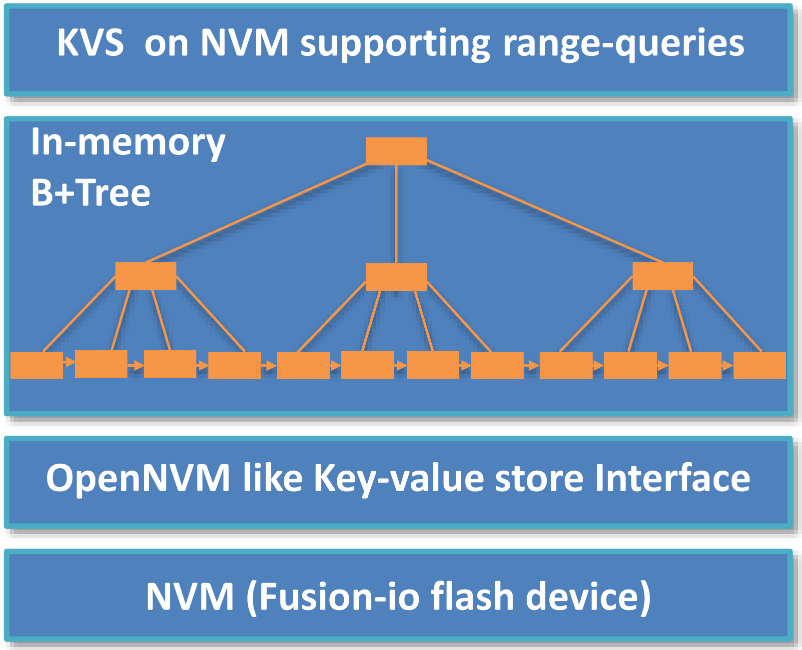
Initial performance of NVM Object Store for FusionIO ioDrive



NVM-BPTree [Jabri and Tatebe, U-Tsukuba]

NVM-BPTree is a Key-Value Stores (KVS) running natively over Non-Volatile-Memory (NVM), like flash, supporting range-queries.

- Take advantage of enterprise class NVM new capabilities: atomic writes, huge sparse address space, direct access to NVM device natively as a KVS
  - *Leverage NVMKV an Open source KVS API interface for NVM like flash.*
- Enable range-queries support for KVS running natively on NVM like fusionio ioDrive
  - *Keys stored in a in-memory B+ Tree with negligible overhead for KV pair insertion and retrieval.*
- Provide optional persistence to the BPTree structure and also snapshots



International Efforts on Big Data and Extreme Computing Convergence

- Europe, US, China, Japan collaboration

International Exascale Software Project (2009-2012)

Big Data and Extreme Computing (2013-)

TSUBAME4 2021-22 K-in-a-Box Convergent Architecture

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

10 Petaflops, 10 Petabyte 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"

EBD: Summary

- **Current "Big Data" not so "Big" but Next Gen will be!**
- **IDC&Clouds inadequate to handle such EBD! ➔ <u>"CONVERGENCE" a must!</u>**

- **<u>EBD Projects Objective: Develop fundamental "convergence" EBD systems and infrastructural technologies through "co-designs" with representative EBD applications</u>**
  - **(1) EBD Convergent Architecture**
  - **(2) EBD Algorithms**
  - **(3) EBD Proggramming Abstractions**
  - **(4) EBD System Software**

- **EBD Convergence will make the current IDCs "Jurassic"**