Towards Efficient Communication and I/O on Oakforest-PACS: Large-scale KNL+OPA Cluster

Toshihiro Hanawa

Joint Center for Advanced HPC (JCAHPC)

Information Technology Center, the University of Tokyo

Agenda

- Introduction of JCAHPC
- Overview of Oakforest-PACS system
- MPI performance issues and remedies
  - Collective communication
  - MPI startup
- IME performance

JCAHPC: Joint Center for Advanced High Performance Computing

http://jcahpc.jp

- JCAHPC was established in 2013 under agreement between
  - Center for Computational Sciences (CCS) at University of Tsukuba, and
  - Information Technology Center (ITC) at the University of Tokyo.
- Design, operate and manage a next-generation supercomputer system by researchers belonging to two universities

JCAHPC

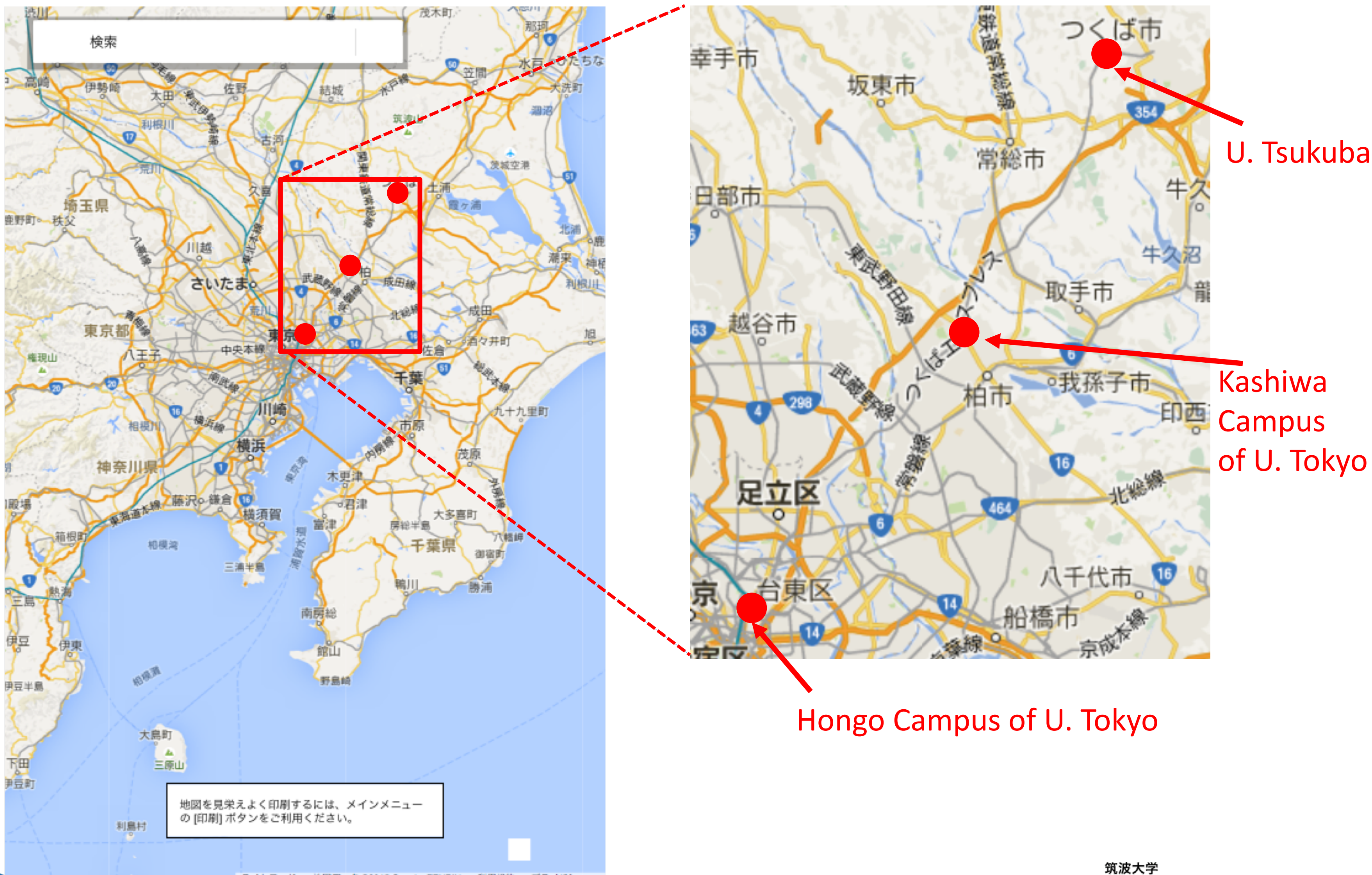
JCAHPC Philosophy

- Organized to manage everything smoothly
- Very tight collaboration for “post-T2K” with two universities based on T2K efforts
  - Originally, T2K Open Supercomputer Alliance by three universities: Tsukuba, Tokyo, and Kyoto since 2008
- For main supercomputer resources, *uniform specification* to ***single shared system***
  - Each university is financially responsible to introduce and operate the system
    -> unified procurement toward single system with ***largest scale in Japan*** (at that moment)

⇒ Oakforest-PACS (OFP)

Machine location: Kashiwa Campus of U. Tokyo

Google マップ https://www.google.com/maps/@?dg=dbrw&newdg=1



Oakforest-PACS (OFP)

U. Tokyo convention　U. Tsukuba convention

⇒ Don't call it just "Oakforest" !
"OFP" is much better

- 25 PFLOPS peak
- 8208 KNL CPUs
- FBB Fat-Tree by OmniPath
- HPL 13.55 PFLOPS
  #1 in Japan → #2
  #6→#12
- HPCG #3→#7
- Green500 #6→#25
- IO500 #1

- Full operation started Dec. 2016
- Official Program started on April 2017

51st TOP500 List (ISC18, June 2018)

JCAHPC

	Site	Computer/Year Vendor	Cores	R_{max} (PFLOPS)	R_{peak} (PFLOPS)	Power (MW)
1	Oak Ridge National Laboratory, USA	Summit, IBM P9 22C 3.07GHz, Mellanox EDR, NVIDIA GV100, 2018 IBM	2,282,544	122.3	187.7	8.8
2	National Supercomputing Center in Wuxi, China	Sunway TaihuLight , Sunway MPP, Sunway SW26010 260C 1.45GHz, 2016 NRCPC	10,649,600	93.0	125.4	15.4
3	Lawrence Livermore National Laboratory, USA	Sierra, IBM P9 22C 3.1GHz, Mellanox EDR, NVIDIA GV100, 2018 IBM	1,572,480	71.6	119.1	
4	National Supercomputing Center in Tianjin, China	Tianhe-2A, Intel Xeon E5-2692v2, TH Express-2, Matrix-2000, 2018 NUDT	4,981,760	61.4	100.6	18.5
5	AIST, Japan	AI Bridging Cloud Infrastructure (ABCI) , Intel Xeon Gold 20C 2.4GHz, IB-EDR, NVIDIA V100, 2018 Fujitsu	391,680	19.9	32.6	1.65
6	Swiss National Supercomputing Centre (CSCS) , Switzerland	Piz Daint, Cray XC50, Intel Xeon E5 12C 2.6GHz, Aries, NVIDIA Tesla P100, 2017 Cray	361,760	19.6	25.3	2.27
7	Oak Ridge National Laboratory, USA	Titan Cray XK7/NVIDIA K20x, 2012 Cray	560,640	17.6	27,1	8.21
8	Lawrence Livermore National Laboratory, USA	Sequoia BlueGene/Q, 2011 IBM	1,572,864	17.2	20,1	7.89
9	Los Alamos NL / Sandia NL, USA	Trinity, Cray XC40, **Intel Xeon Phi 7250 68C 1.4GHz, Cray Aries, 2017 Cray**	979,968	14.1	43.9	3.84
10	DOE/SC/LBNL/NERSC USA	Cori, Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Cray Aries, 2016 Cray	632,400	**14.0**	**27.9**	3.94
11	KISTI, Korea	Nurion, Cray CS500, Intel Xeon-Phi 7250 68C 1.4GHz, Intel Omni-Path, 2018 Cray	570,020	**13.9**	25.7	
12	Joint Center for Advanced High Performance Computing, Japan	Oakforest-PACS, PRIMERGY CX600 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path, 2016 Fujitsu	557,056	**13.5**	**24.9**	2.72

http://www.top500.org/

IO 500 Ranking (June, 2018)

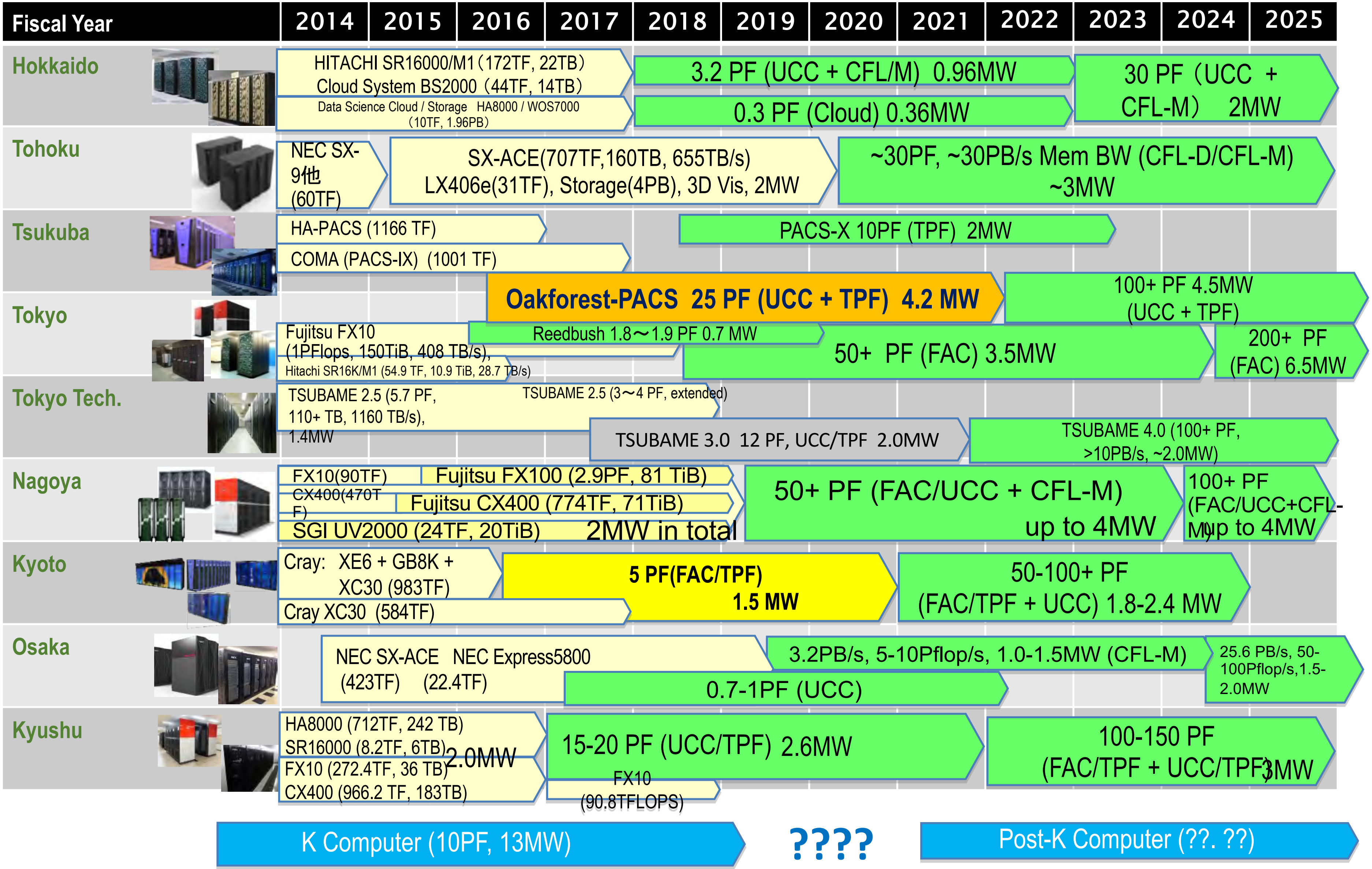
	Site	Computer	File system	Client nodes	IO500 Score	BW (GiB/s)	MD (kIOP/s)
1	JCAHPC, Japan	Oakforest-PACS	DDN IME	2048	137.78	560.10	33.89
2	KAUST, Saudi	Shaheen2	Cray DataWarp	1024	77.37	496.81	12.05
3	KAUST, Saudi	Shaheen2	Lustre	1000	41.00	54.17	31.03
4	JSC, Germany	JURON	BeeGFS	8	35.77	14.24	89.81
5	DKRZ, Germany	Mistral	Lustre2	100	32.15	22.77	45.39
6	IBM, USA	Sonasad	Spectrum Scale	10	24.24	4.57	128.61
7	Fraunhofer, Germany	Seislab	BeeGFS	24	16.96	5.13	56.14
8	DKRZ, Germany	Mistral	Lustre1	100	15.47	12.68	18.88
9	Joint Institute for Nuclear Research	Govorun	Lustre	24	12.08	3.34	43.65
10	PNNL, USA	EMSL Cascade	Lustre	126	11.12	4.88	25.33

http://www.io500.org/

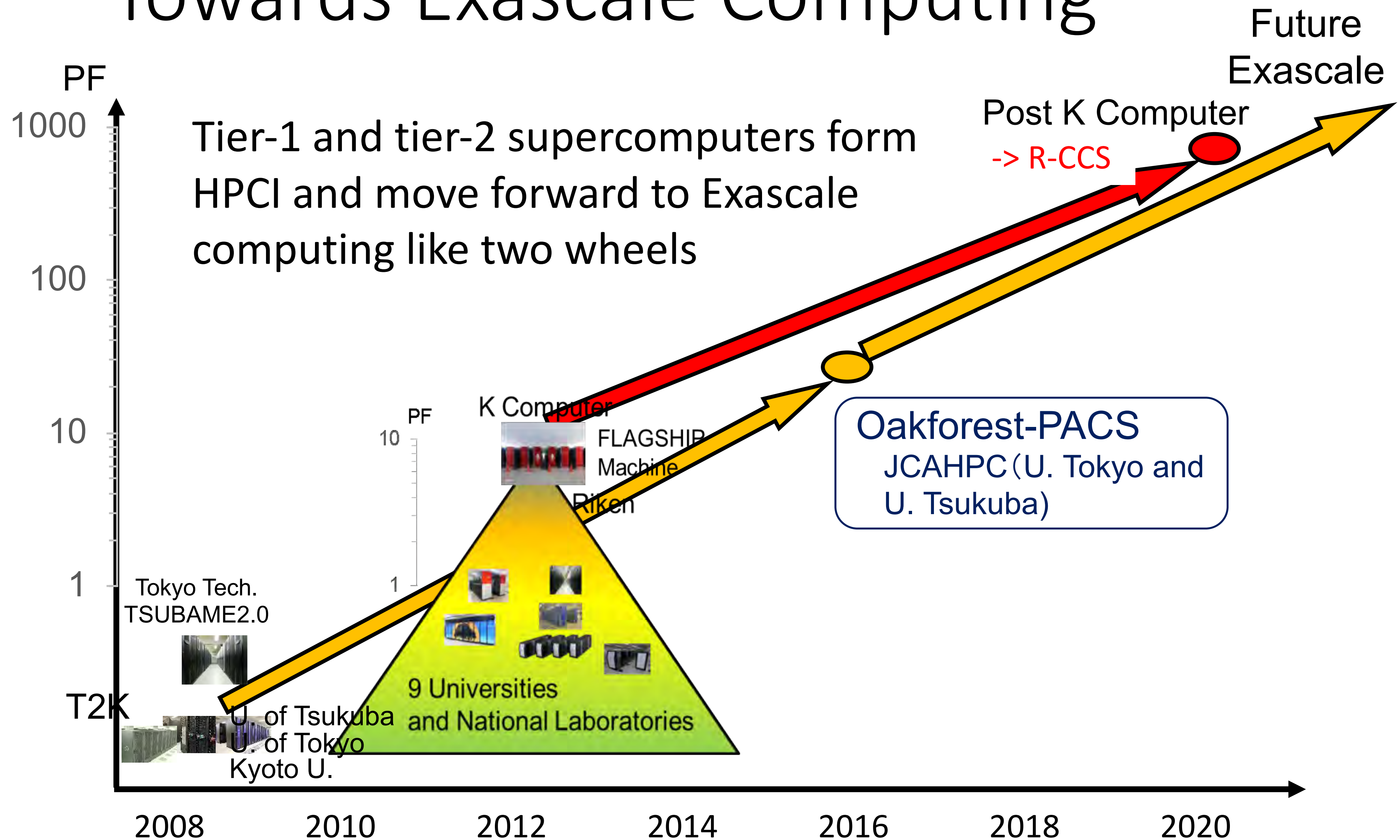
Deployment plan of 9 supercomputing center（Feb. 2017）

JCAHPC

Power consumption indicates maximum of power supply (includes cooling facility)

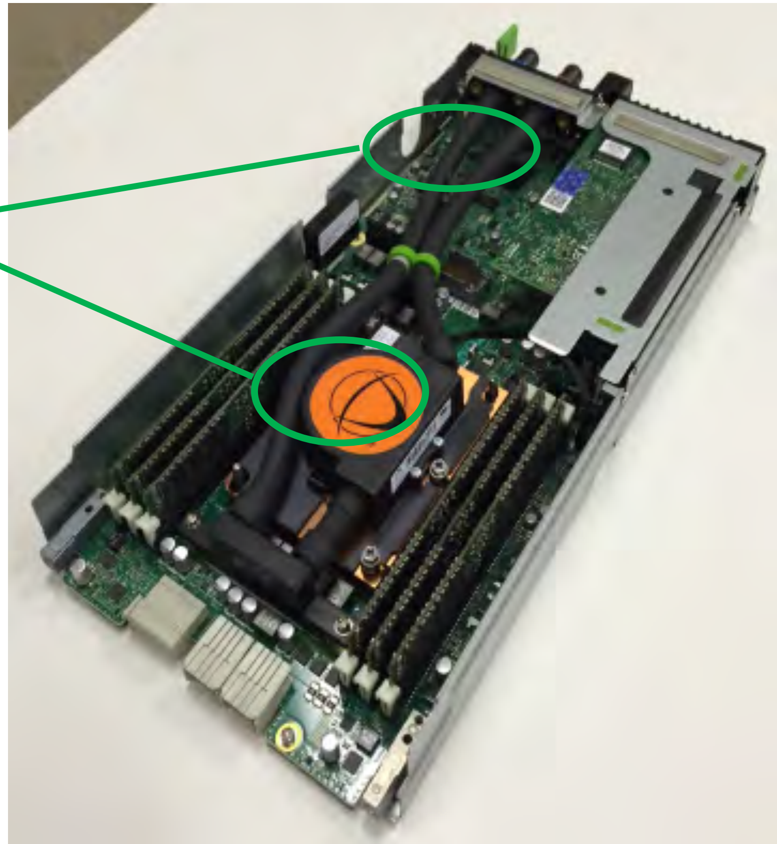


Towards Exascale Computing



Computation node & chassis

Water cooling pump & pipe



Chassis with 8 nodes, 2U size (Fujitsu PRIMERGY CX600 M1)

Computation node (Fujitsu PRIMERGY CX1640)
with single chip Intel Xeon Phi 7250 (68c Knights Landing, 3+TFLOPS)
and Intel Omni-Path Architecture card (100Gbps)

Water cooling pipes and rear door cooling (radiator)

Storage system of OFP

- Shared File System
  - Lustre, 26PB
- File Cache System
  - DDN IME, Burst Buffer: 940 TB, 50 servers

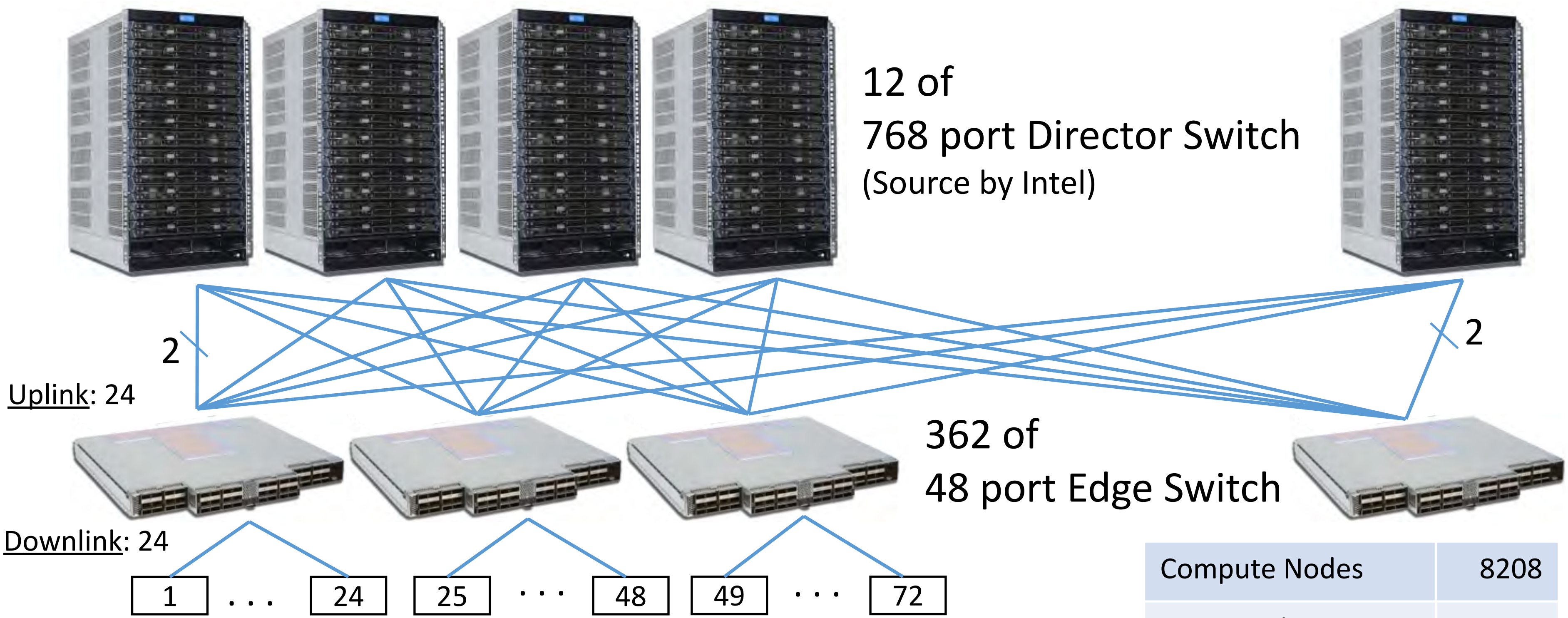
Specification of Oakforest-PACS system

Total peak performance			25 PFLOPS
Total number of compute nodes			8,208
Compute node	Product		Fujitsu PRIMERGY CX600 M1 (2U) + CX1640 M1 x 8node
	Processor		Intel® Xeon Phi™ 7250 (Code name: Knights Landing), 68 cores, 1.4 GHz
	Memory	High BW	16 GB, 490 GB/sec (MCDRAM, effective rate)
		Low BW	96 GB, 115.2 GB/sec (peak rate)
Inter-connect	Product		Intel® Omni-Path Architecture
	Link speed		100 Gbps
	Topology		Fat-tree with (completely) full-bisection bandwidth

Specification of Oakforest-PACS system (I/O)

Parallel File System	Type		Lustre File System
	Total Capacity		26.2 PB
	Meta data	Product	DataDirect Networks MDS server + SFA7700X
		# of MDS	4 servers x 3 set
		MDT	7.7 TB (SAS SSD) x 3 set
	Object storage	Product	DataDirect Networks SFA14KE
		# of OSS (Nodes)	10 (20)
		Aggregate BW	~500 GB/sec
Fast File Cache System	Type		Burst Buffer, Infinite Memory Engine (by DDN)
	Total capacity		940 TB (NVMe SSD, including parity data by erasure coding)
	Product		DataDirect Networks IME14K
	# of servers (Nodes)		25 (50)
	Aggregate BW		~1,560 GB/sec

Full bisection bandwidth Fat-tree by Intel® Omni-Path Architecture



Firstly, to reduce switches&cables, we considered :

- All the nodes into subgroups are connected with FBB Fat-tree
- Subgroups are connected with each other with >20% of FBB

But, HW quantity is not so different from globally FBB, and globally FBB is preferredfor flexible job management.

Compute Nodes	8208
Login Nodes	20
Parallel FS	64
IME	300
Mgmt, etc.	8
Total	**8600**

Facility of Oakforest-PACS system

<table>
<tr><td colspan="3">Power consumption</td><td>4.2 MW (including cooling)
→ actually around 3.0 MW</td></tr>
<tr><td colspan="3"># of racks</td><td>102</td></tr>
<tr><td rowspan="4">Cooling system</td><td rowspan="2">Compute Node</td><td>Type</td><td>Warm-water cooling
Direct cooling (CPU)
Rear door cooling (except CPU)</td></tr>
<tr><td>Facility</td><td>Cooling tower & Chiller</td></tr>
<tr><td rowspan="2">Others</td><td>Type</td><td>Air cooling</td></tr>
<tr><td>Facility</td><td>PAC</td></tr>
</table>

Software of Oakforest-PACS

- OS: Red Hat Enterprise Linux (Login nodes), CentOS or McKernel (Compute nodes, dynamically switchable)
  - McKernel: OS for many-core CPU developed by RIKEN R-CCS
    - Ultra-lightweight OS compared with Linux, no background noise to user program
    - Expected to be installed to post-K computer
- Compiler: GCC, Intel Compiler, XcalableMP
  - XcalableMP: Parallel programming language developed by RIKEN R-CCS and University of Tsukuba
    - Easy to develop high-performance parallel application by adding directives to original code written by C or Fortran
- Application: Open-source softwares
  - : OpenFOAM, ABINIT-MP, PHASE system, FrontFlow/blue, and so on

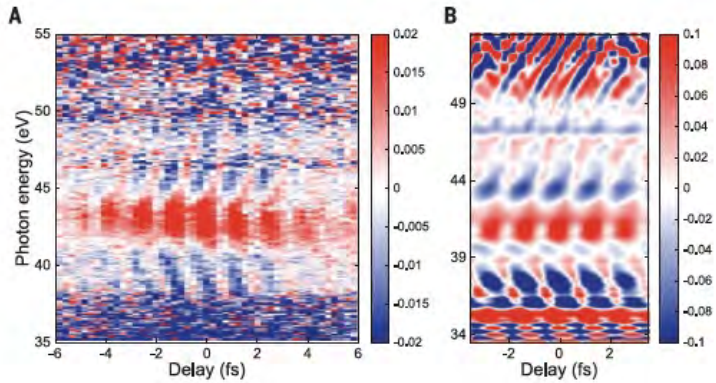
Software of Oakforest-PACS

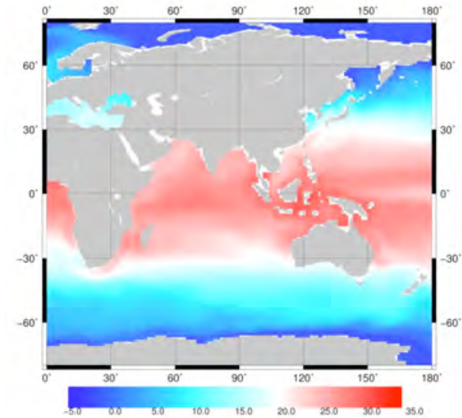
	Compute node	Login node
OS	CentOS 7, McKernel	Red Hat Enterprise Linux 7
Compiler	gcc, Intel compiler (C, C++, Fortran)	
MPI	Intel MPI, MVAPICH2	
Library	Intel MKL LAPACK, FFTW, SuperLU, PETSc, METIS, Scotch, ScaLAPACK, GNU Scientific Library, NetCDF, Parallel netCDF, Xabclib, ppOpen-HPC, ppOpen-AT, MassiveThreads	
Application	mpijava, XcalableMP, OpenFOAM, ABINIT-MP, PHASE system, FrontFlow/blue, FrontISTR, REVOCAP, OpenMX, xTAPP, AkaiKKR, MODYLAS, ALPS, feram, GROMACS, BLAST, R packages, Bioconductor, BioPerl, BioRuby	
Distributed FS		Globus Toolkit, Gfarm
Job Scheduler	Fujitsu Technical Computing Suite	
Debugger	Allinea DDT	
Profiler	Intel VTune Amplifier, Trace Analyzer & Collector	

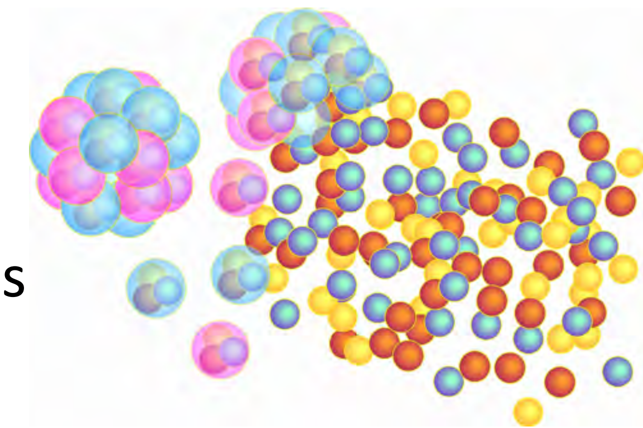
Applications on OFP

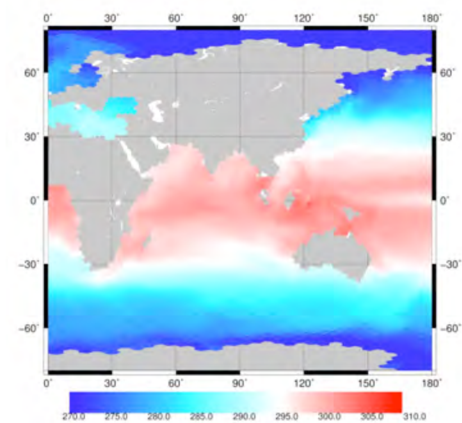
JCAHPC

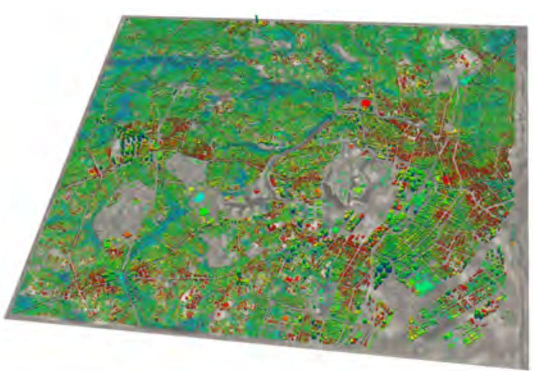
- **ARTED**
  - Ab-initio Electron Dynamics
- **Lattice QCD**
  - Quantum Chrono Dynamics
- **NICAM & COCO**
  - Atmosphere & Ocean Coupling
- **GAMERA/GHYDRA**
  - Earthquake Simulations
- **Seism3D**
  - Seismic Wave Propagation

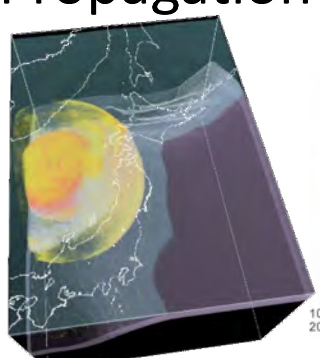


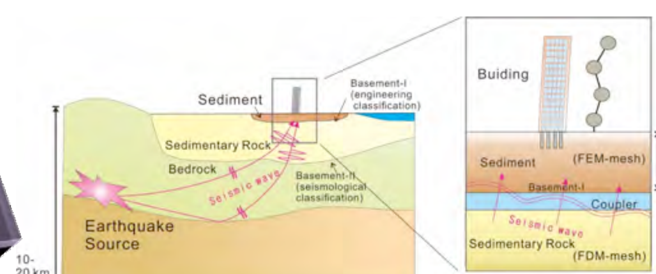


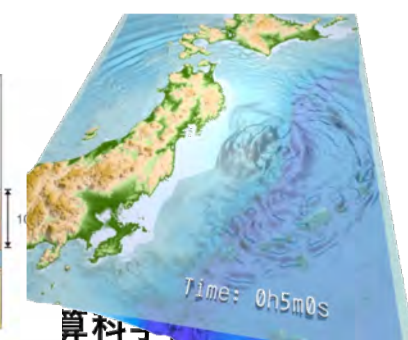






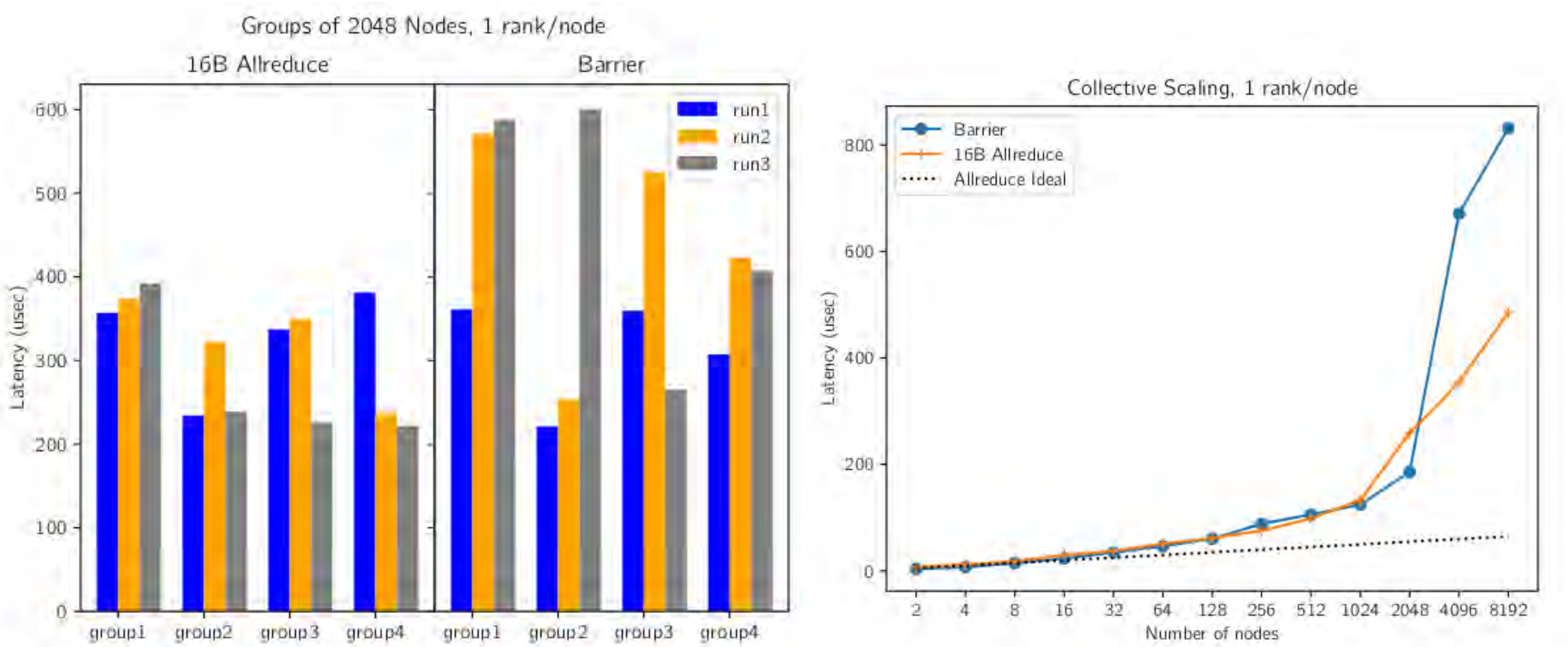






Center for Computational Sciences

Collective comm.: Run-to-run variability & Performance Issue



by courtesy of M. Horikoshi, Intel
M. Horikoshi et. al (incl. Hanawa), IXPUG 2018 in HPC Asia 2018 WS

Root cause of variance

Frequency transition (Turbo): 1.4GHz <-> 1.5GHz <-> 1.6GHz

- Transition stalls many microseconds.

Periodic MWAIT wake-up:

- Linux system default is using idle=mwait. MONITOR and MWAIT instructions on idle hardware threads.
- KNL forces a periodic wake-up of hardware threads in an MWAIT state 10 times per second and additionally cause frequency transitions on the entire processor .

OS work:

- Daemons, hardware interrupts, middleware (system monitoring, scheduling). idle thread on the same core or tile is awakened to perform OS work, the application thread will be delayed and additionally cause frequency transitions.

by courtesy of M. Horikoshi, Intel
M. Horikoshi et. al (incl. Hanawa), IXPUG 2018 in HPC Asia 2018 WS

Remedies

Setting was done on OFP compute node

Impact of effect

idle=halt: Stopping MONITOR/MWAIT and single-tile turbo (No 1.6GHz)

Tickless mode (nohz_full=2-67,70-135,138-203,206-271): Decreasing OS timer interrupt from 1KHz to 1Hz except tile-0. And excluding tile-0 from application.

Binding Lustre daemon and system process to tile-0

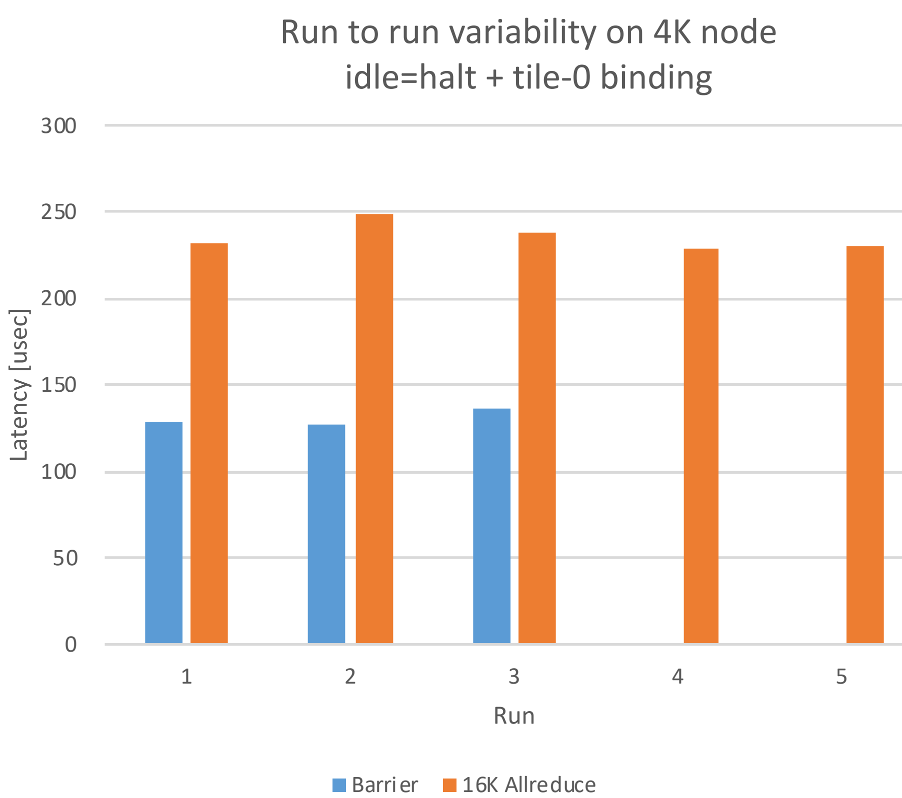
Using acpi-cpufreq driver rather than intel_pstate

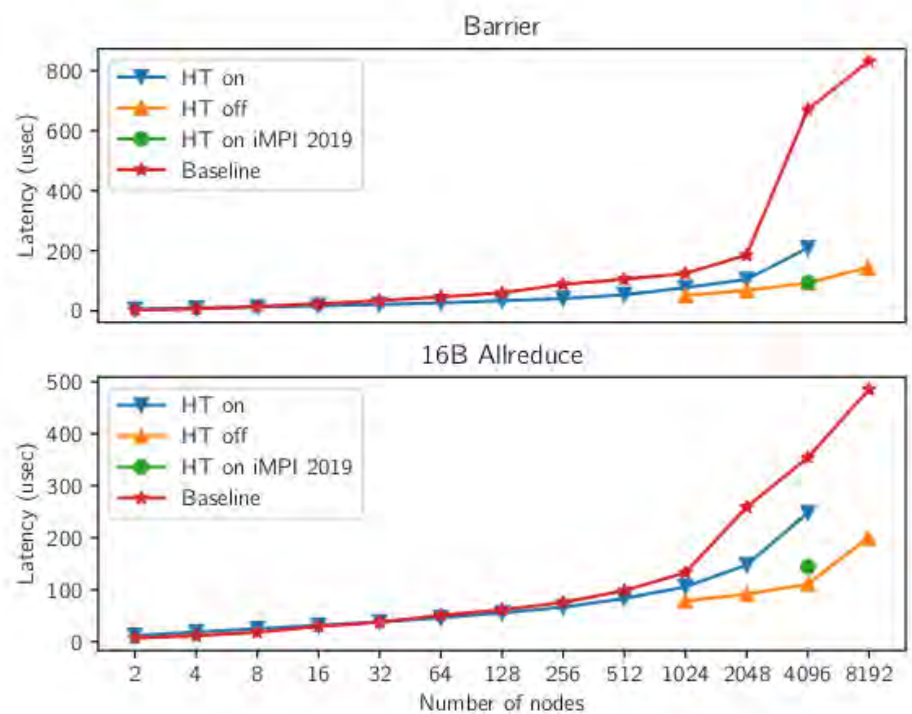
Tuning spinning: `PSM2_YIELD_SPIN_COUNT=10000` and `I_MPI_COLL_SHM_PROGRESS_SPIN_COUNT=100000`

* These remedies have cons side effects (effect depends on situation and application).

by courtesy of M. Horikoshi, Intel
M. Horikoshi et. al, IXPUG 2018 in HPC Asia 2018 WS

After improvement



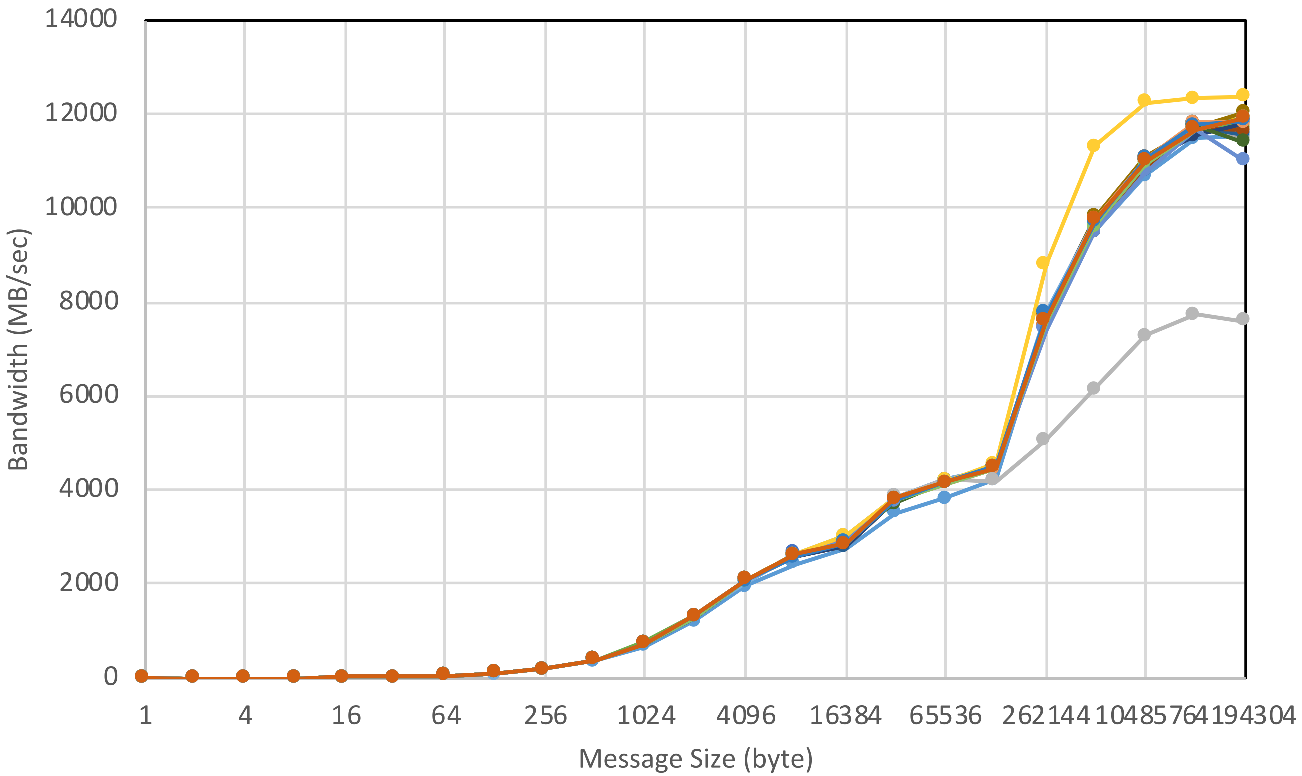


by courtesy of M. Horikoshi, Intel
M. Horikoshi et. al, IXPUG 2018 in HPC Asia 2018 WS

4K node collective	Target [usec]	Baseline [usec]	Optimized [usec]
Barrier	105	671	94
16B Allreduce	160	485	145

Core-awareness on MPI comm.

- osu_bw result on “core-by-core”
- Intel MPI 2018.2

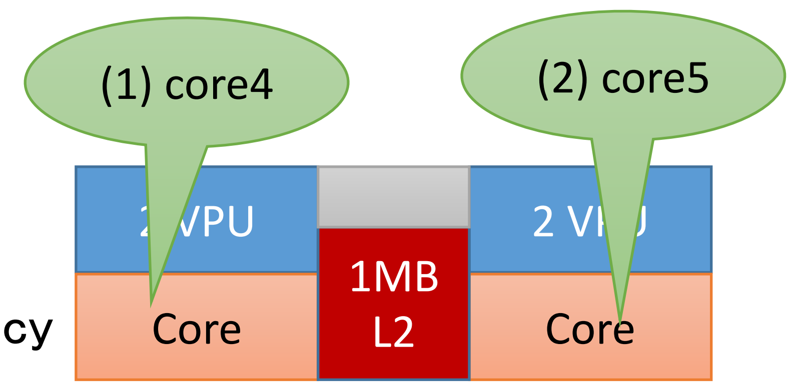


Why?

- Omni-Path hardware
  - 16ch of DMA engines for transmission (SDMA0 - 15)
  - After transmission, interrupting to core for following msg
- Interrupt vector to be handled is 19
  - SDMA0 - 15
  - hfi1
  - kctxt0 - 2
- "Omni-Path Fabric Performance Tuning User Guide" gives a solution
  - Adjusting the interrupt handler mapping to core improves the performance

Driver setting on OPA

- Setting core to handle interrupt (1)
  `/proc/irq/int #/smp_affinity_list`
  - "irqbalance" is suggested on KNL, then drivers are mapped to core 3 - 18 initially
- Core setting for SDMA usage (2)
  (core to do polling by kernel thread)
  `/sys/devices/pci0000:00/0000:00:01.0/0000:01:00.0/infiniband/hfi1_0/sdma[0-15]/cpu_list`
  - no setting by default
- How to set
  - different core in the same tile
  - sharing L2 cache improves efficiency

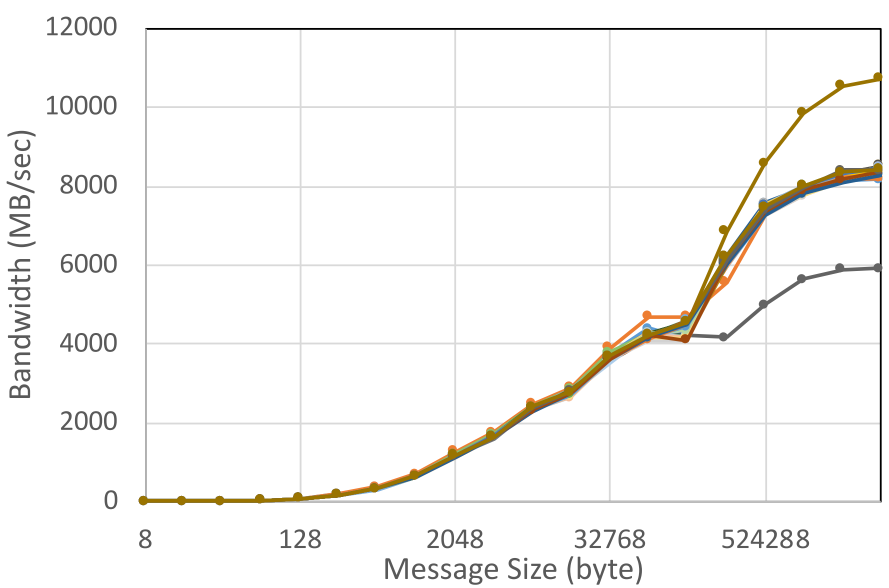


Optimal setting on OPA

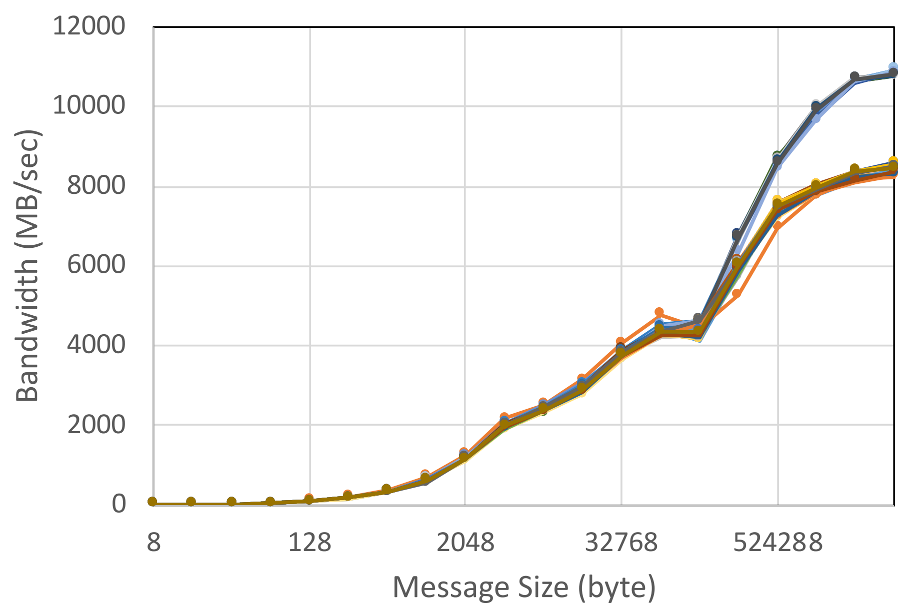
- Consideration
  - In case of MPI+OpenMP, it is highly possible where
  - "master" thread is mapped on "even" numbered core
  - core#0 receives interrupt by timer, then mapping kctxt to #2,#3,#4 and hfi1 to #2, respectively
- Interrupting cores of SDMA0~15: 5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,10
- Interrupt handling cores of SDMA: 4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,11
  - the last core specification is just for previous experience where core#11 was bad (why?)
  - this special core varies on KNL chip itself

After optimization

- Intel MPI 2017u2 on “OFP-mini” testbed with XP7210 (since we cannot change the OS configuration from user mode) => difficult to determine optimal setting



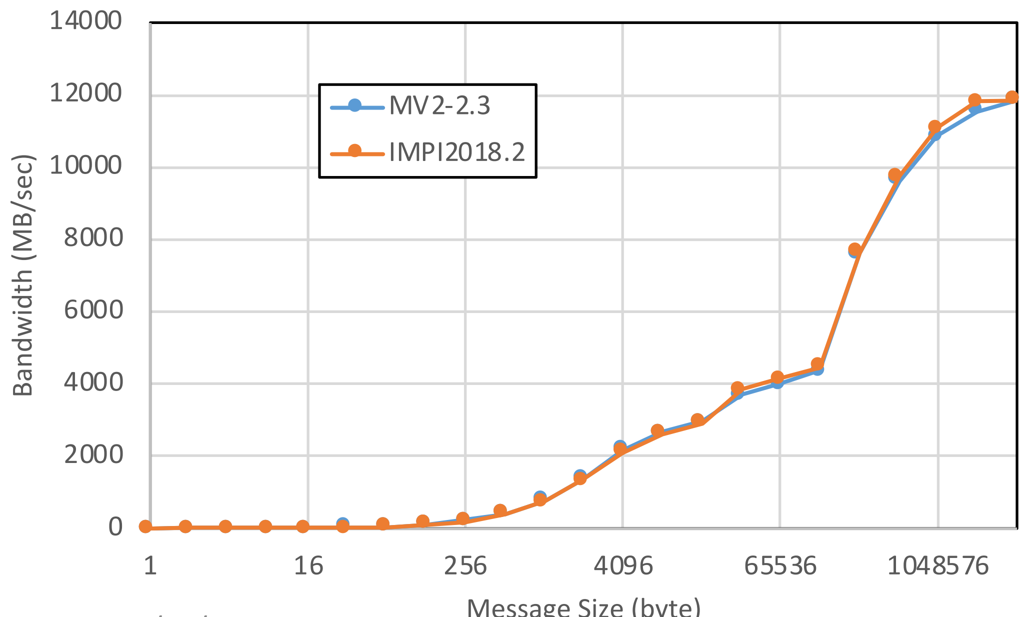
Before



After

Core-awareness on MVAPICH2

- MVAPICH2 2.3 with latest OPA-PSM2 driver
- Intel MPI 2018.2
- MVAPICH2 result is the same as the typical case of Intel MPI. => No core-awareness is observed in MV2!!

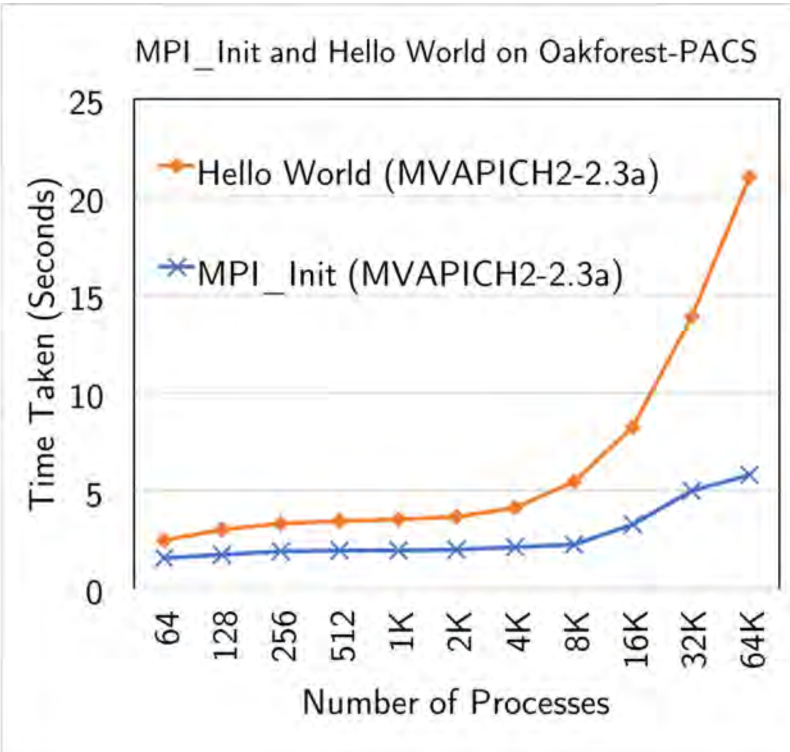


MPI startup issue

- MPI_Init cost up to 2K node (64 proc/node = 128K proc.) performance improvement on Intel MPI

Procs	Node	Procs/ node	IMPI (2017.1.132)	IMPI 2017 U3 PR
8192	128	64	74.553	8.799
16384	256	64	168.785	15.543
32768	512	64	427.791	36.698
40960	640	64	586.260	54.010
65536	1024	64	1223.968	124.607
131072	2048	64	3667.967	335.690

MPI startup improvement



from MVAPICH2 web page

- After this measurement (2017/3), remedies ("idle=halt," described in collective comm.) were applied, so slightly worse now.
  - Now, "Hello World" with 8K is 7.04 s

Latest "Hello World" results

# of MPI proc.	Node	IMPI 2018.2	IMPI 2018.3	MV2 2.3a with hydra	MV2 2.3 with rsh
8192	128 (64 ppn)	8.12	6.69	25.67	10.54

IOR benchmark result using MV2 on IME

- IOR-easy write in IO500 benchmark
  - 2MB transfer size, file per process, several GB file size per process
  - Regulation: over 5min, but MV2 case is shorter (too big filesize for my quota …)
- MVAPICH2-2.3+IME patch vs Intel MPI 2018.2
  - 128 node
  - FUSE: access to /cache/datadir for /work/datadir
  - native IME: ime:///work/datadir
    - MV2_ENABLE_AFFINITY=0 is required for good performance. Why??

# of MPI proc.	IMPI 2018.2 (FUSE) [GB/s]	MV2 2.3+IME [GB/s]
2K (16 ppn)	268.7	651.7
4K (32 ppn)	289.1	671.0
8K (64 ppn)	288.9	564.5

Request for MVAPICH2 on OFP

- Hydra support with same performance as "rsh" case
  - In current MVAPICH2 "mpirun_rsh" is obviously better performance than "mpiexec.hydra," but Hydra has better job scheduler support.
- Core "exclude list" like "I_MPI_PIN_PROCESSOR_EXCLUDE_LIST" environment variable
  - OFP specifies "tickless" core to core #1-67, so each job should not be assigned on core #0 (and #1).

Summary

- JCAHPC is a joint resource center for advanced HPC by U. Tokyo and U. Tsukuba
- Oakforest-PACS (OFP) with 25 PFLOPS peak performance
  - Intel Xeon Phi (Knights Landing) and Omni-Path Architecture
- OFP is used not only for HPCI and other resource offering program but also a testbed for McKernel and XcalableMP system software to support post-K development.
- Many issues on KNL+OPA combination, such as core-awareness, collective comm. scalability, and hyperthreading.
  - MVAPICH2 helps many performance improvements and complemental functions, such as native MPI-IO support for IME.
- We will continue the investigation using one of world largest KNL+OPA cluster OFP to overcome issues on new technology combination and pursue efficient way on manycore environment.
  - Multiple-Endpoint for OpenMP+MPI hybrid program is promising technique.
  - McKernel can also help performance stability without OS jitters.