Status of the Simplified Sustained System Performance Benchmark Project

Miwako Tsuji*, Mitsuhisa Sato*, Bill Kramer**

*RIKEN Center for Computational Science

**National Center for Supercomputing Applications / University of Illinois

RIKEN R-CCS NCSA

9th JLESC Workshop @ UTK 2019.04.15-17

Agenda

- Overview of the SSSP
- Status of the SSSP
  - Result for a SKL cluster
  - Result for large scale experiments
- Summary
- Mini-symposium about performance metrics at SIAM PP20

Benchmarks in HPC

- Benchmarks in various stages of system development life cycles
  - design, development, implement and maintenance
  - to estimate, evaluate and assure the effectiveness of the system
- **Application performance of systems**
- High Performance Linpack (HPL)
  - (One of) most widely used benchmark
  - Top500
  - **no longer strongly correlates to real application performance**

Performance evaluation w/ (mini) applications

- A system performance for applications can be directly examined

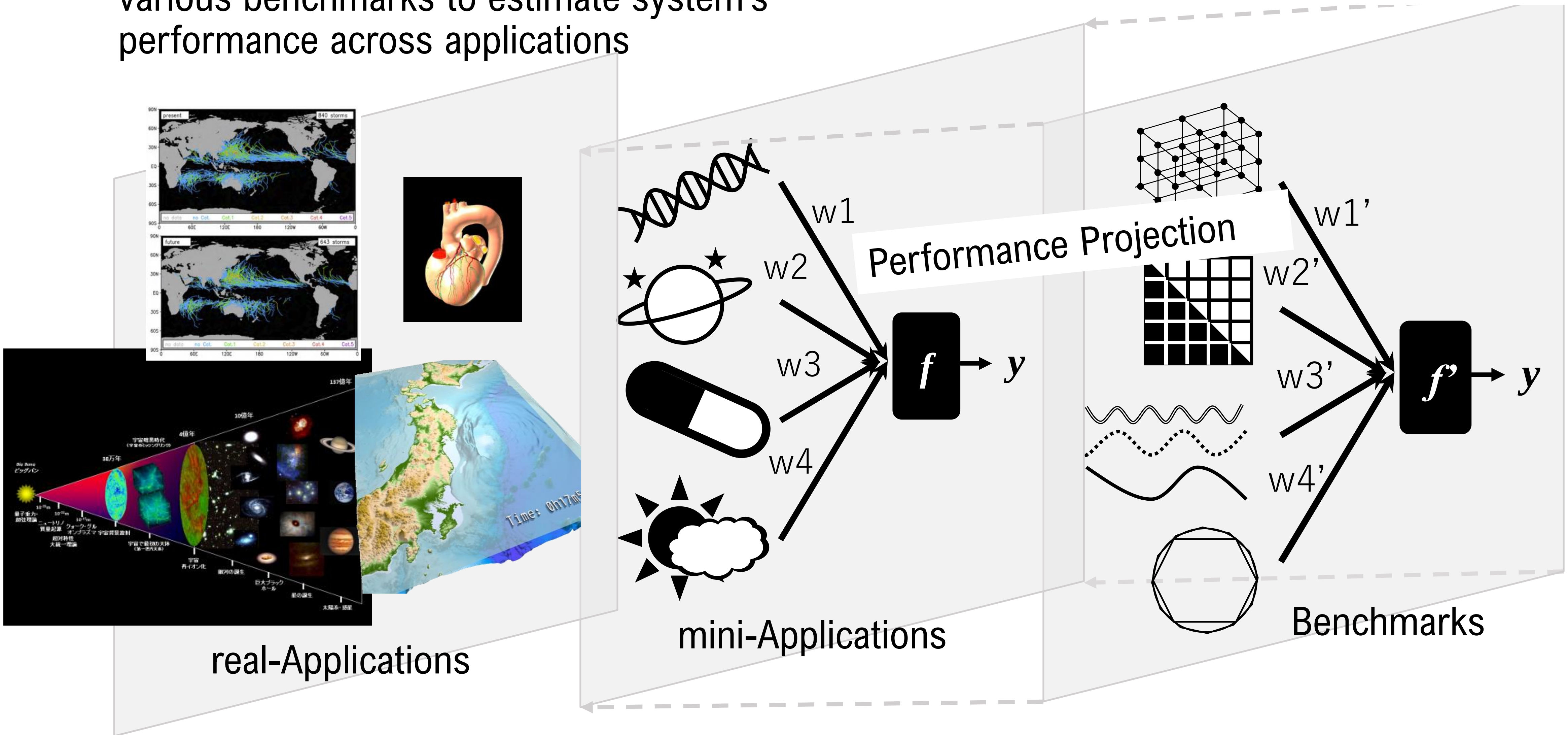
- A set of (mini) applications
  - Fiber by RIKEN
- Sustained System Performance (SSP) metric [Kramer, 2005]
  - enables evaluation of a system's performance using time-to-solution while at the same time accommodating any number of application areas
  - used to procure systems in NERSC, NCSA, etc.…

- Performance evaluation using full, or mini applications **requires more effort** than using the simple/kernel benchmark programs

Purpose of Research

- To construct a metric by combining various benchmarks to estimate system's performance across applications

・systems' performance insights for applications
・without executing and porting applications



Sustained System Performance (SSP) Metric

- A metric to evaluate systems from the view point of applications' throughputs

Per Processor Performance (PPP)

$$\mathrm{SSP}(s) = N \sum_{i,j} \frac{f_{i,j}}{m_{i,j} \times t_{i,j}}$$

The # of nodes: N

# of floating point operations: $f_{i,j}$

concurrency: $m_{i,j}$

elapsed time: $t_{i,j}$

$i \in I$ a set of **applications**

$j \in J_i$ a data set for i

Simplied Sustained System Performance (SSSP) Metric

- Performance projection of the SSP metric

$$\mathrm{SSP}(s) = N \sum_{i,j} \frac{f_{i,j}}{m_{i,j} \times t_{i,j}}$$

$i \in I$

a set of **applications**

If SSSP(s) ≒ SSP(s)
we don't have to evaluate a set of applications.
All we have to evaluate is a set of simple benchmarks!

$$\mathrm{SSSP}(s) = N \sum_{i,j} w_{i,j} \frac{f_{i,j}}{m_{i,j} \times t_{i,j}}$$

$i \in I$

a set of **benchmarks**

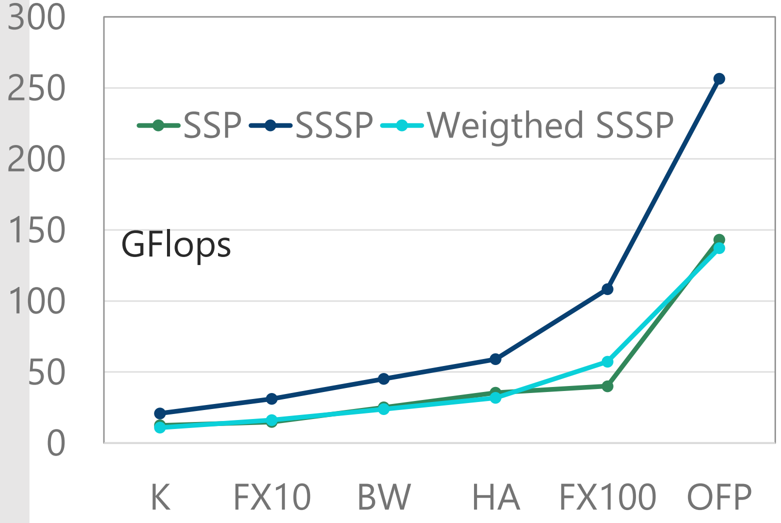
weighing factors to minimize $\sum_s |SSSP(s) - SSP(s)|$

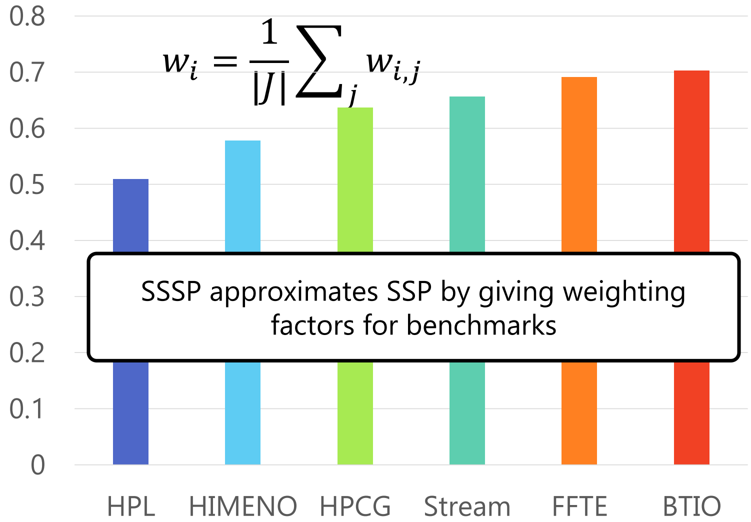
Previous works

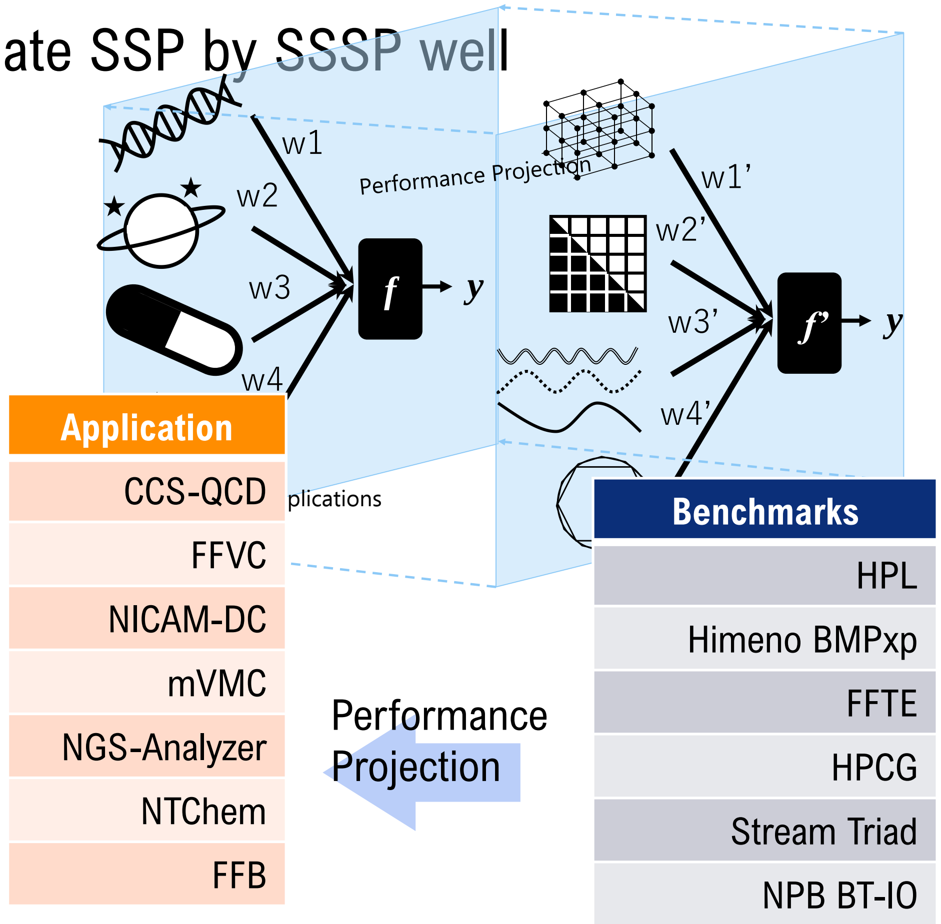
- We had computed the SSP, SSSP metrics over 6 different systems
  - K, FX10, FX100, BlueWaters, HA-PACS, OFP
- We had confirmed that
  - SSP and SSSP are consistent without any weight
  - with appropriate weighting factors, we could approximate SSP by SSSP well
- Note: reference weighting factors to minimize

$$\sum_s |SSSP(s) - SSP(s)|$$

are computed with Least Absolute Deviation algorithm







Update(1): SSP and SSSP on a new system (SKL)

1. execute a set of benchmarks on a SKL cluster
2. compute SSSP metric from the results and weighting factors obtained from previous 6 systems

$$\sum_{i,j} w_{i,j} \frac{f_{i,j}}{m_{i,j} \times t_{i,j}}$$

computed by 6 systems (w/o SKL) — results from SKL

3. execute a set of applications on the SKL cluster to compute SSP
4. compare SSSP (obtained at 2.) and SSP (obtained at 3.)

※ We omit N to compare SSP and SSSP

INTI HPC R&D cluster installed at CEA, France

	SKL@inti.cea
CPU Theoretical Peak per K	Intel Skylake 8168 Platinum 2.7GHz 24 cores x 2sockets 2073 GFlops 16.2
Memory	DDR4-2666 192 GB, ? GB/s (par socket)
Cache	32 KB L1 inst/core 32 KB L1 data/core 24MB L2/node 33MB L3/node
Network	

Update(1): SSP and SSSP on a new system (SKL)

application	data	K	SKL@inti.cea	ratio
CCS-QCD	class1	18.431	96.894	5.257102
	class2	11.039	60.264	5.459112
NICAM-DC	gl05rl00z40pe10	4.900	7.590	1.548934
	gl05rl00z80pe5	5.870	5.225	0.890221
FFVC	1024^3	12.718	168.145	13.22144
	256^3	13.962	120.578	8.636109
NTChem	h2o	10.914	114.472	10.48861
	taxol	61.179	1062.133	17.36108
FFB	sample	5.718	23.991	4.195572
	sample	5.718	23.991	4.195572
mVMC	job_middle	19.114	246.619	12.90258
	job_tiny	4.308	6.793	1.576712
NGS Analyzer	dummy	0.011	0.348	32.63896
	dummy	0.011	0.348	32.63896
	(peak)	128	2073.600	16.2

Per processor performance for SSP (target)

$$\sum_{i,j} \frac{f_{i,j}}{m_{i,j} \times t_{i,j}} = 138.385\text{G}$$

Update(1): SSP and SSSP on a new system (SKL)

benchmark	data	Per Processor Performance, p	weight 6 (w/o SKL)
HIMENO	M	94.5	0.6635
	L	73.3	0
FFTE	1024^3	18.4	1.0151
	512^3	24.7	1.2825
HPL	80000	2106	0.1833
	160000	1988.8	0.4191
HPCG	256^3	12.97	1.2346
	512^3	12.90	1.231
Stream Triad	2^15	27.5	1.3371
	2^29	14.7	1.0938
NAS BTIO	C	43.6	2.0003
	D	84.5	1.6106

$$\sum_{i,j} \frac{f_{i,j}}{m_{i,j} \times t_{i,j}} = 357.8\ \text{G}$$

$$\sum_{i,j} w_{i,j} \frac{f_{i,j}}{m_{i,j} \times t_{i,j}} = 136.7\ \text{G}$$

Performance Projection

SSP (Target)

$$\sum_{i,j} \frac{f_{i,j}}{m_{i,j} \times t_{i,j}} = 138.385\text{G}$$

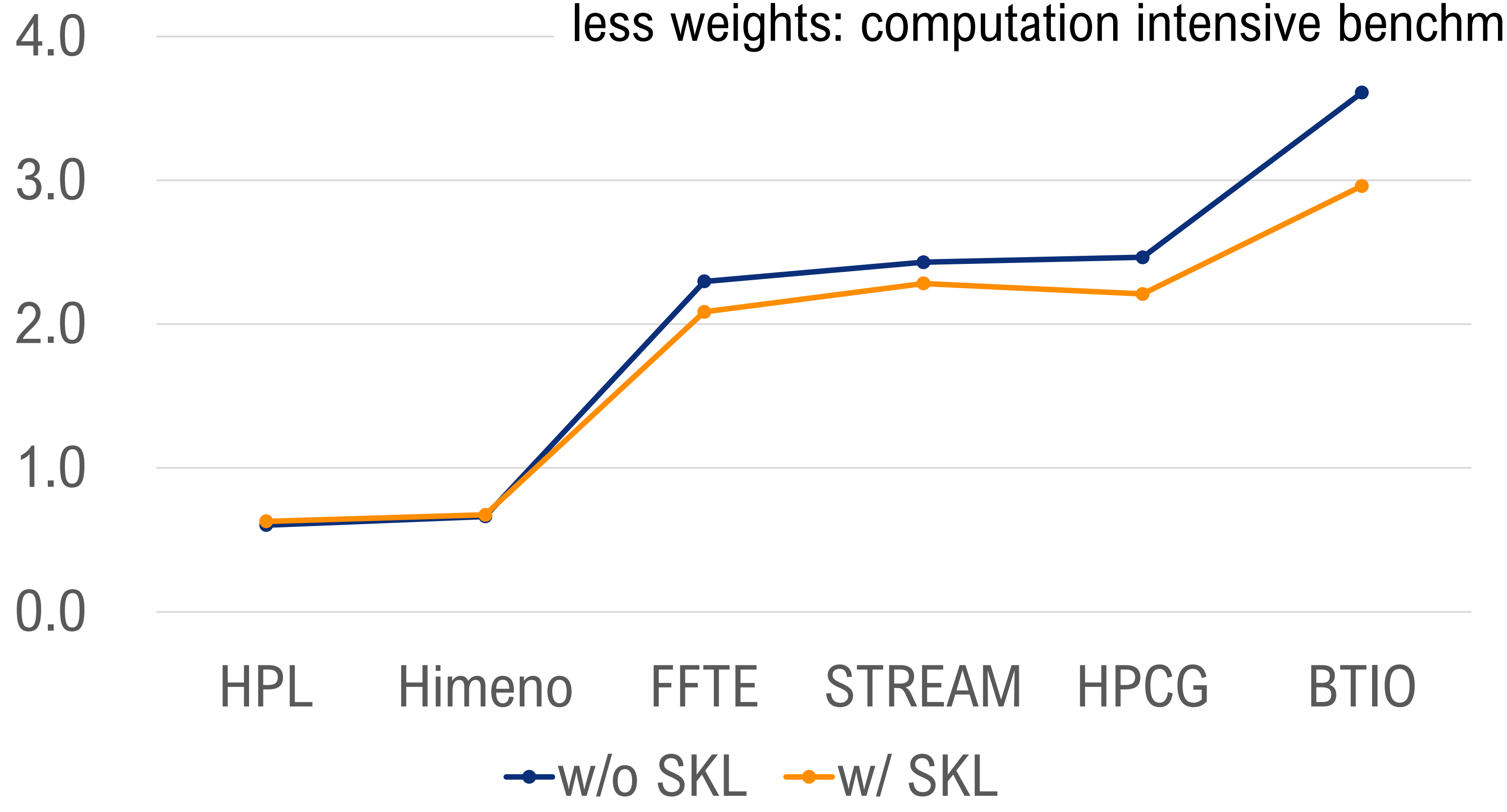
Update(1): SSP and SSSP on a new system (SKL)

benchmark	data	Per Processor Performance, p	weight 6 (w/o SKL)	weight 7 (w/ SKL)		
HIMENO	M	94.5	0.6635	0.6745	SSSP	353.751
	L	73.3	0	0	(w/o weight)	
FFTE	1024^3	18.4	1.0151	0.9337	SSSP	136.721
	512^3	24.7	1.2825	1.1524	weight-6	
HPL	80000	2106	0.1833	0.3447	SSSP	138.193
	160000	1988.8	0.4191	0.2846	weight-7	
HPCG	256^3	12.97	1.2346	1.105	SSP	138.385
	512^3	12.90	1.231	1.1049	(Target)	
Stream Triad	2^15	27.5	1.3371	1.2143		
	2^29	14.7	1.0938	1.0691		
NAS BTIO	C	43.6	2.0003	1.5145		
	D	84.5	1.6106	1.4471		

Update(1): SSP and SSSP on a new system (SKL)

- Averaged Weighting Factors over data set for each benchmark

more weights : IO, memory intensive benchmarks
less weights: computation intensive benchmarks



Update(2): large scale experiments on K and OFP

	K	FX10	FX100	HAPACS	Blue Waters	Oakforest PACS
CPU	SPARC64TM VIIIfx 2GHz 8 cores	SPARC64TM IXfx 1.65GHz 16 cores	SPARC64TM XIfx 1.975GHz 32+2 cores	Intel E5 2670 2.6 GHz 8 cores x 2sockets	AMD 6276 Interlagos 2.3 GHz 16 Bulldozercores x2	Intel Xeon Phi Nights Landing 1,4GHz 68 cores
Theoretical Peak	128 GFlops	211.2 GFlops	1011.2 GFlops	332.8 GFlops	313.6 GFlops	3046 GFlops
per K	1.00	1.65	7.90	2.60	2.25	23.8
Memory	DDR3 SDRAM 16 GB, 64 GB/s	DDR3 SDRAM 32 GB, 8 GB/s	HMC 32 GB 240R+240W GB/s	DDR3 SDRAM 128 GB, 102.8 GB/s	DDR3 SDRAM 64 GB	MCDRAM+DDR4 16GB +96GB
Cache	32 KB L1 inst/core 32 KB L1 data/core 6 MB L2/node	32 KB L1 inst/core 32 KB L1 data/core 12MB L2/node	32 KB L1 inst/core 32 KB L1 data/core 12MBx2 L2/node	32 KB L1 inst/core 32 KB L1 data/core 256 KB L2 cache/core 20MB L2 cache/node	64 KB L1 inst/2core 16 KB L1 data/core 2 MB L2/2core 8MB L3/4core	32 KB L1 inst/core 32 KB L1 data/core 2MB L2/2node
Network	Tofu Interconnect 5 GiB/s x 2	Tofu Interconnect 5 GiB/s x 2	Tofu Interconnect 2 12.5 GiB/s x 2	Fat-Tree 4 GiB/s x 2	Cray Gemini torus interconnect 9.6 GiB/s Injection	Intel Omni Path 100 gbps

Previous experiments: up to 64 nodes → 8192 nodes　　up to 8 nodes → 1024 nodes

Update(2): large scale experiments on K and OFP

Applications

- up to 8000 nodes of the K computer and 1000 nodes of the OFP, 10 % of each system)

		K		OFP	
Applications	**Data**	**procs x thrds**	**# of nodes**	**procs x thrds**	**# of nodes**
CCS-QCD	class-3	512x8	512	512x8	64
	class-4	8000x8	8000	8000x8	1000
NICAM-DC	gl08rl03z80pe640	640x8	640	640x8	80
	gl09rl04z80pe2560	2560x8	2560	2560x8	320
FFVC	1024	512x8	512	512x8	64
	4096	4096x8	4096	4096x8	512
NTChem	c60atc60h28	128x8	128	128x8	16
	c54h18x2_cc-pvdz	4x8	4	4x18	1
FFB	115	4096x8	4096	4096x8	512
	69	512x8	512	512x8	64
mVMC	1024	1024x8	1024	1024x8	128
	8192	8192x8	8192	8192x8	1024

Update(2): large scale experiments on K and OFP

Benchmarks

- up to 8092 nodes of the K computer and 1024 nodes of the OFP, 10 % of each system)

		K		OFP	
Benchmarks	**Data**	**procs x thrds**	**# of nodes**	**procs x thrds**	**# of nodes**
FFTE	4096^3	1024x8	1024	1024x8	128
	4096^3	8192x8	8192	8192x8	1024
HPL	480000	8192x1	1024	8192x1	128
	960000	65536x1	8192	65536x1	1024
HPCG	512^3	4096x1	512	4096x1	64
	1024^3	4096x1	512	4096x1	64
Stream Triad	2^15	1x8	1	1x32	1
	2^29	1x8	1	1x16	1

※ Note: I/O benchmark and application are omitted due to storage restriction

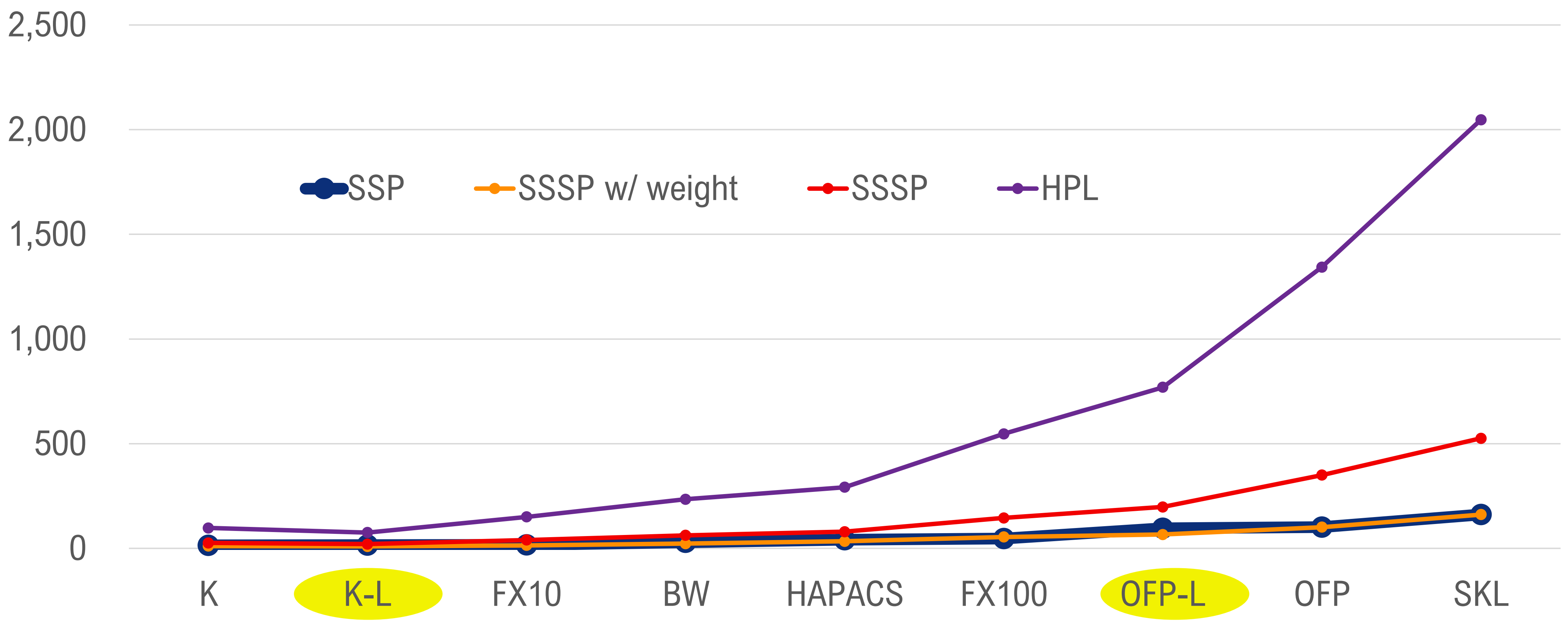
Update(2): large scale experiments on K and OFP

Per Processor Performances

Applications	Data	K	OFP
CCS-QCD	class-3	11.058	39.444
	class-4	13.310	38.776
NICAM-DC	gl08rl03z80pe640	5.099	6.792
	gl09rl04z80pe2560	5.129	6.729
FFVC	1024	10.556	65.289
	4096	12.489	60.155
NTChem	c60atc60h28	69.660	721.000
	c54h18x2_cc-pvdz	10.140	86.512
FFB	115	6.296	5.453
	69	5.827	5.284
mVMC	1024	19.139	61.133
	8192	17.933	60.468

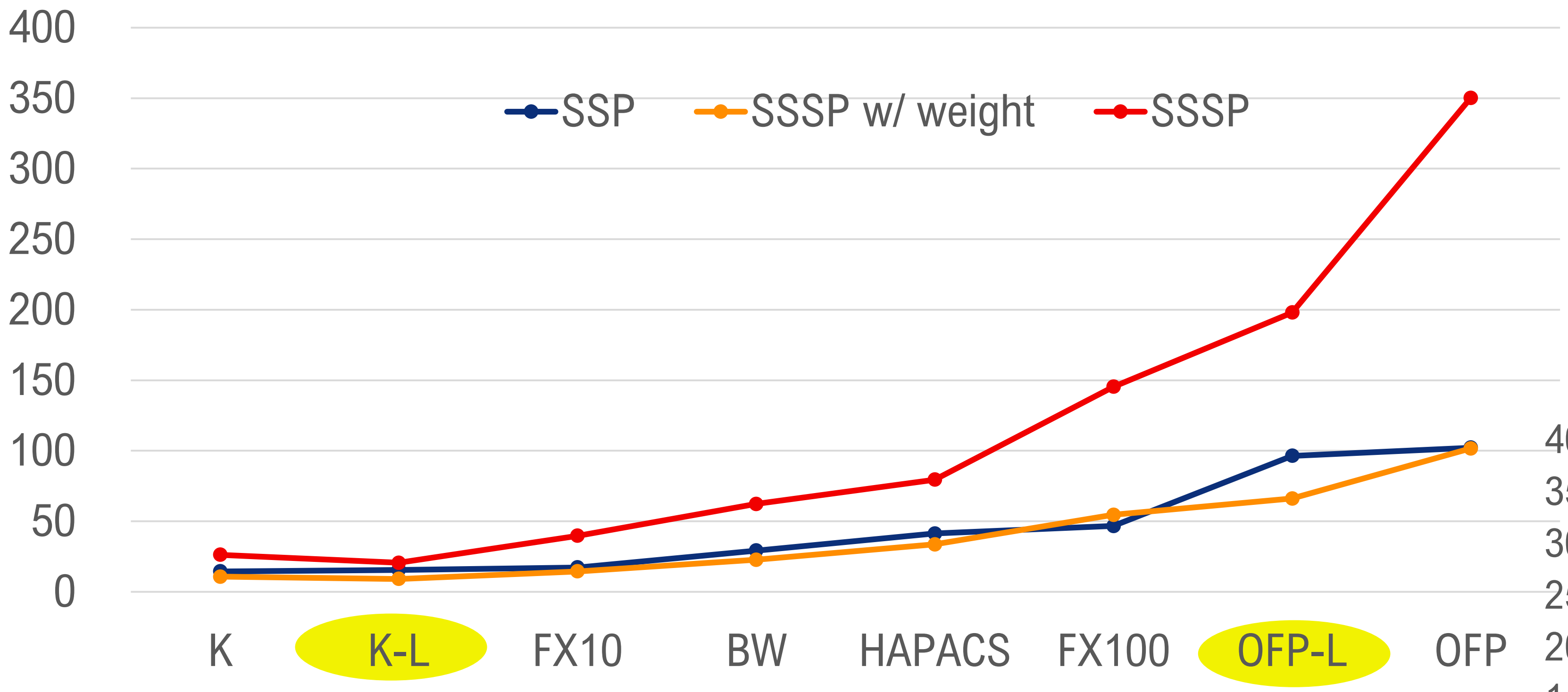
Benchmarks	Data	K	OFP
FFTE	4096^3	1.476	9.243
	4096^3	1.602	5.401
HPL	480000	83.320	874.219
	960000	68.262	664.941
HPCG	512^3	1.782	6.993
	1024^3	1.464	6.987
Stream Triad	2^15	3.855	10.923
	2^29	2.490	5.358

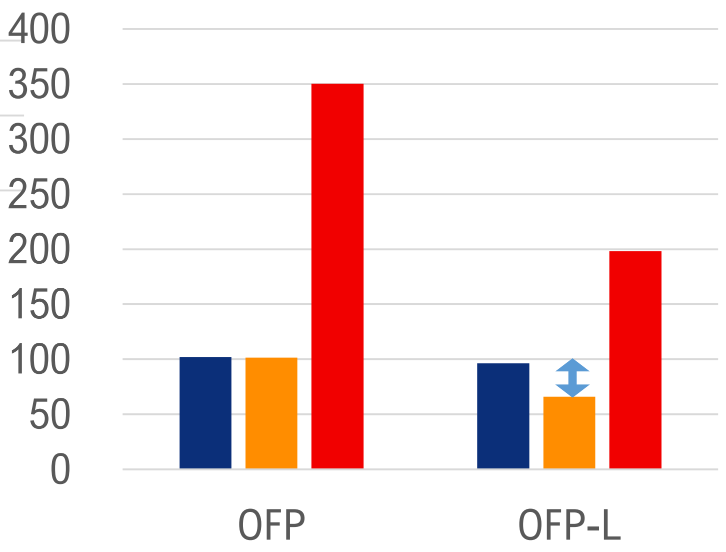
SSP, SSSP, weighted SSSP, HPL scores for 7 systems



- weights are computed based on “small scale” experiments on 7 systems
- (details are next page)

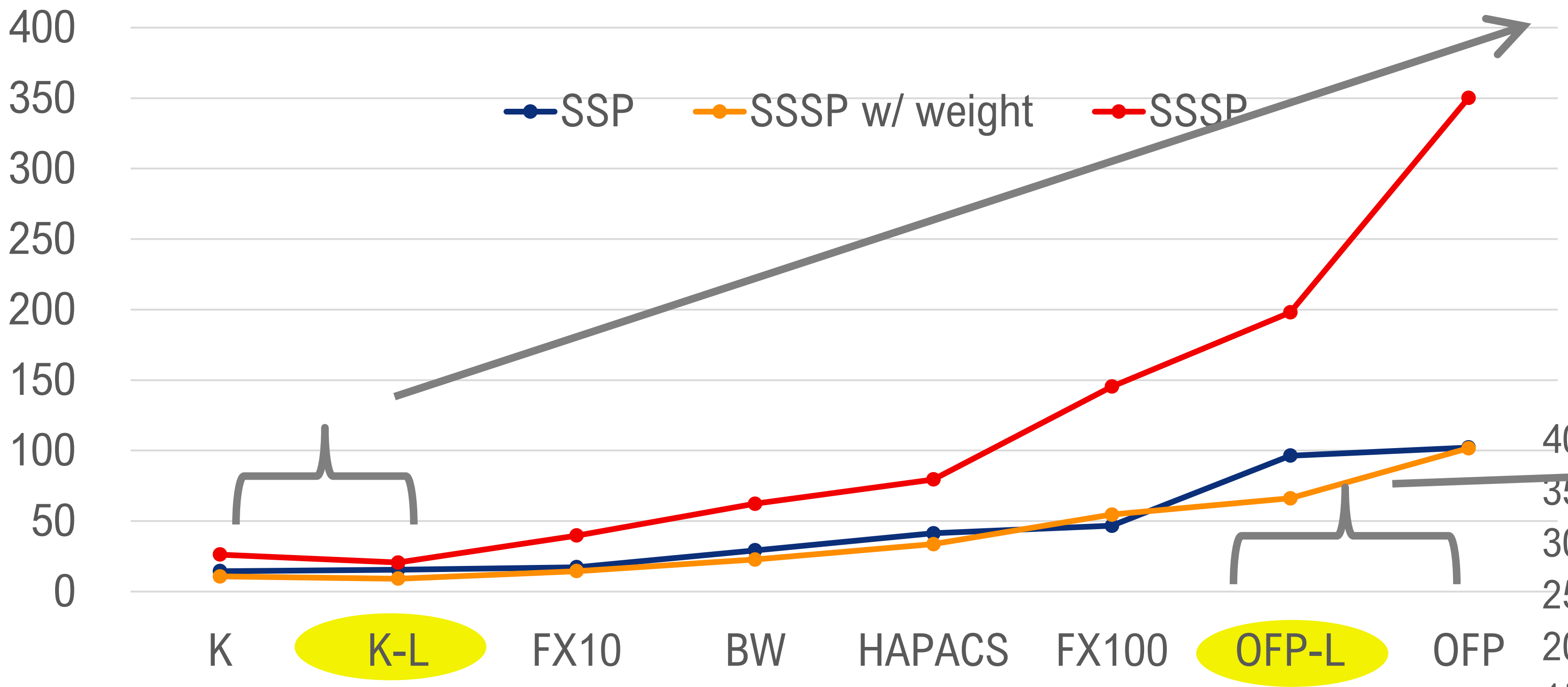
SSP, SSSP, weighted SSSP (detail)

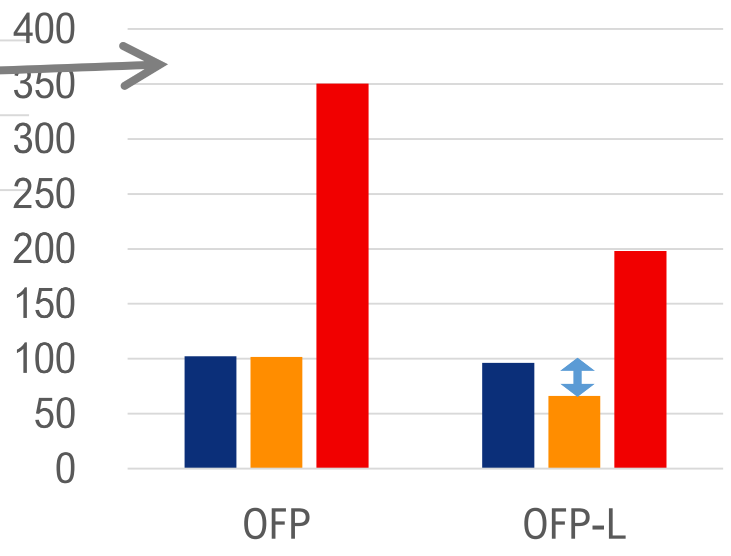




- Approximations for large scale SSP by the weight factors obtained from small experiments are
  - still better than HPL ☺
  - better than SSSP w/o weights
  - less precise than those for small scale SSP

SSP, SSSP, weighted SSSP (detail)





- Approximations for large scale SSP by the weight factors obtained from small experiments are
  - still better than HPL ☺
  - better than SSSP w/o weights
  - less precise than those for small scale SSP

Summary

- [EX1] SKL cluster
  - SSSP with weights from previous systems (without the SKL cluster) can estimate SSP for a new system appropriately

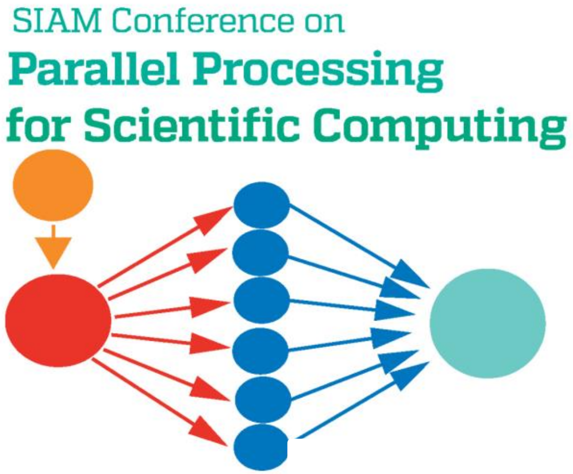
- [EX2] Two large scale systems
  - SSSP with weights from small systems can give **rough estimate** about application performance
    - better than HPL, SSSP w/o weights
  - Future work: compare sets of weights obtained from small systems and large systems

Minisymposium at SIAM PP20

- A mini symposium about performance metrics, benchmarking, etc.. at SIAM PP20
- SIAM PP20 (Society for Industrial and Applied Mathematics Conference on Parallel Processing for Scientific Computing)
  - Date: February of 2020
  - Place: somewhere in USA
  - Submission Deadline for Minisimposium proposal : Not Yet Announced

    (August in case of the PP18)

- Topic
  - benchmarks
  - performance metrics
  - mini apps
  - etc…



- Contact: miwako.tsuji@riken.jp