Using a Flash as Large and Slow Memory for Stencil Computations

Seikei University, Tokyo, Japan

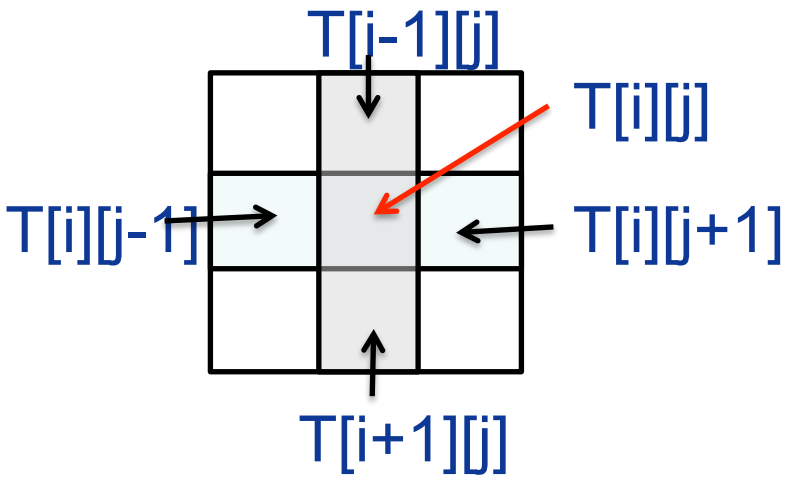
midori@st.seikei.ac.jp

Hiroko Midorikwa

Stencil Computation
an Important core computation in HPC

- Widely used in scientific/engineering simulations
- Iterative data updates with limited data set at every time step
- Memory access: regular, sweeps entire domain data

2-dimensional grid domain



General update form

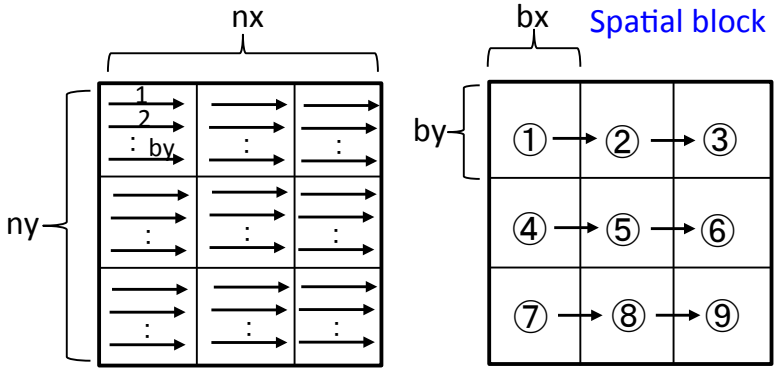
```
T_new[i][j] =
      c1 * T[i][j] +
      c2 * T[i][j-1] + c3 * T[i][j+1] +
      c4  * T[i-1][j] + c5 * T[i+1][j];
  ck: Coefficients
```

5-point stencil computation

Spatial and Temporal blocking tuned for Flash SSDs

An Algorithm extracting Data Access Locality

Spatial Blocking for Two-dimensional grid domain (nx x ny)



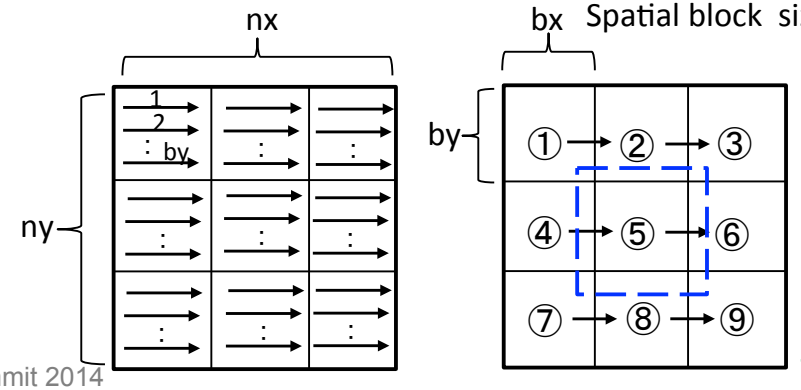
Spatial block size: (bx x by)

Update for one time step at each block

- Blocks from ① to ⑨ are calculated in order.
- After one block is calculated, the next block is processed.

Sweep whole domain Nt time steps

Temporal Blocking for Two-dimensional grid domain (nx x ny)



Spatial block size: (bx x by)

Temporal block size: (bx x by)

Update for bt time steps at each block

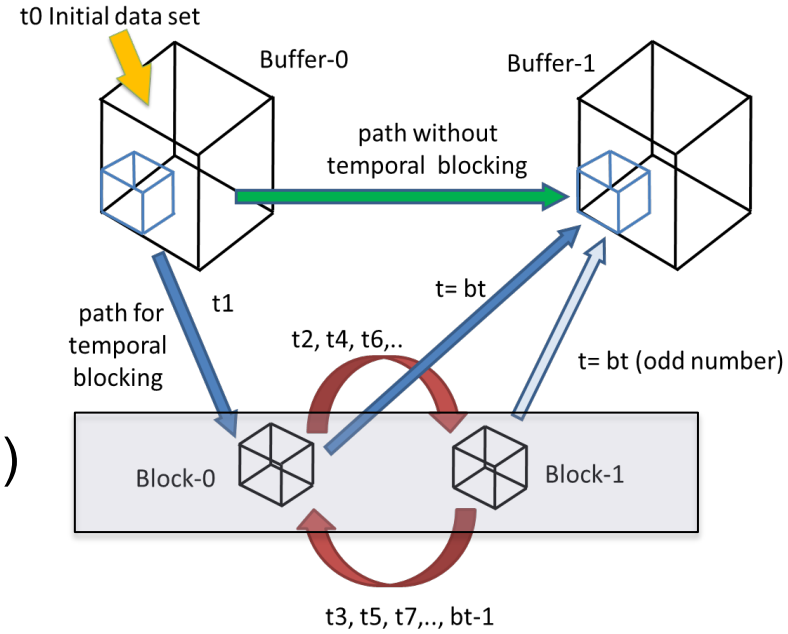
- Blocks from ① to ⑨ are calculated in order.
- After one block is calculated, the next block is processed.

Sweep whole domain Nt/bt time steps

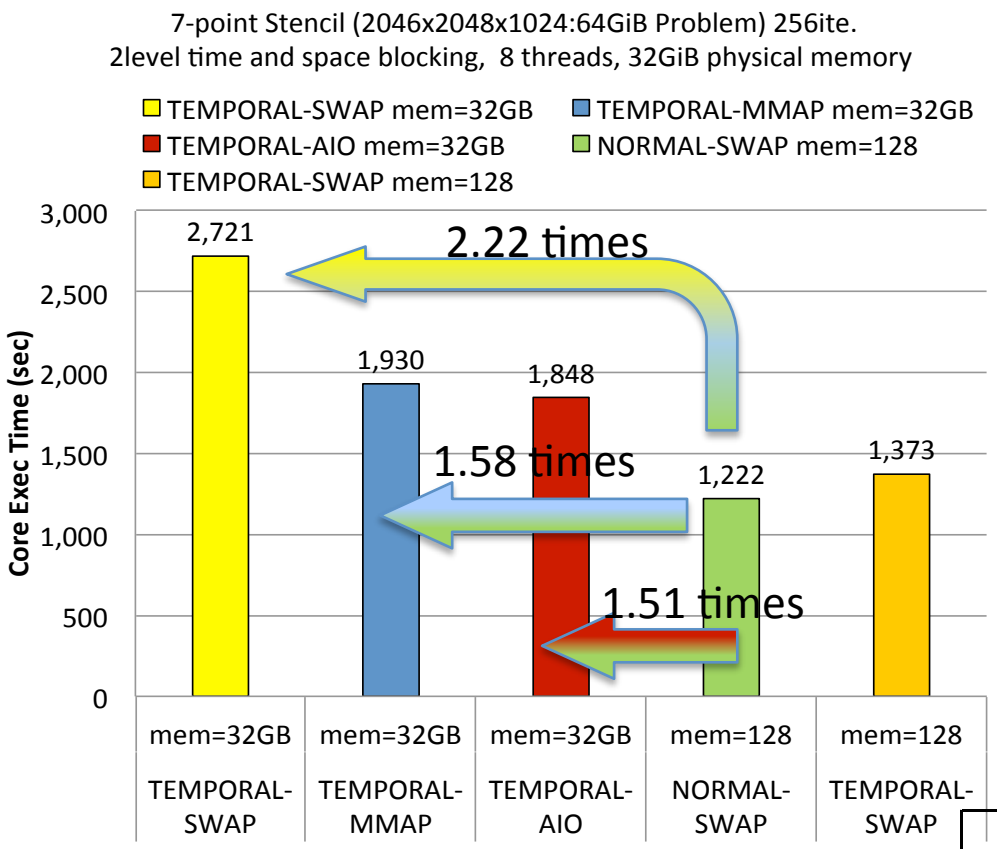
Stencil Computation using Flash SSD for a larger-size problem than physical memory (DRAM)

Three Implementations

- swap method (using swap system)
  uses a flash as a swap device
- mmap method (using file memory map)
  uses a flash as a ext4 file system
- aio method (using asynchronous file io)
  uses a flash as a block device for direct input/output



The Effect of Temporal Blocking



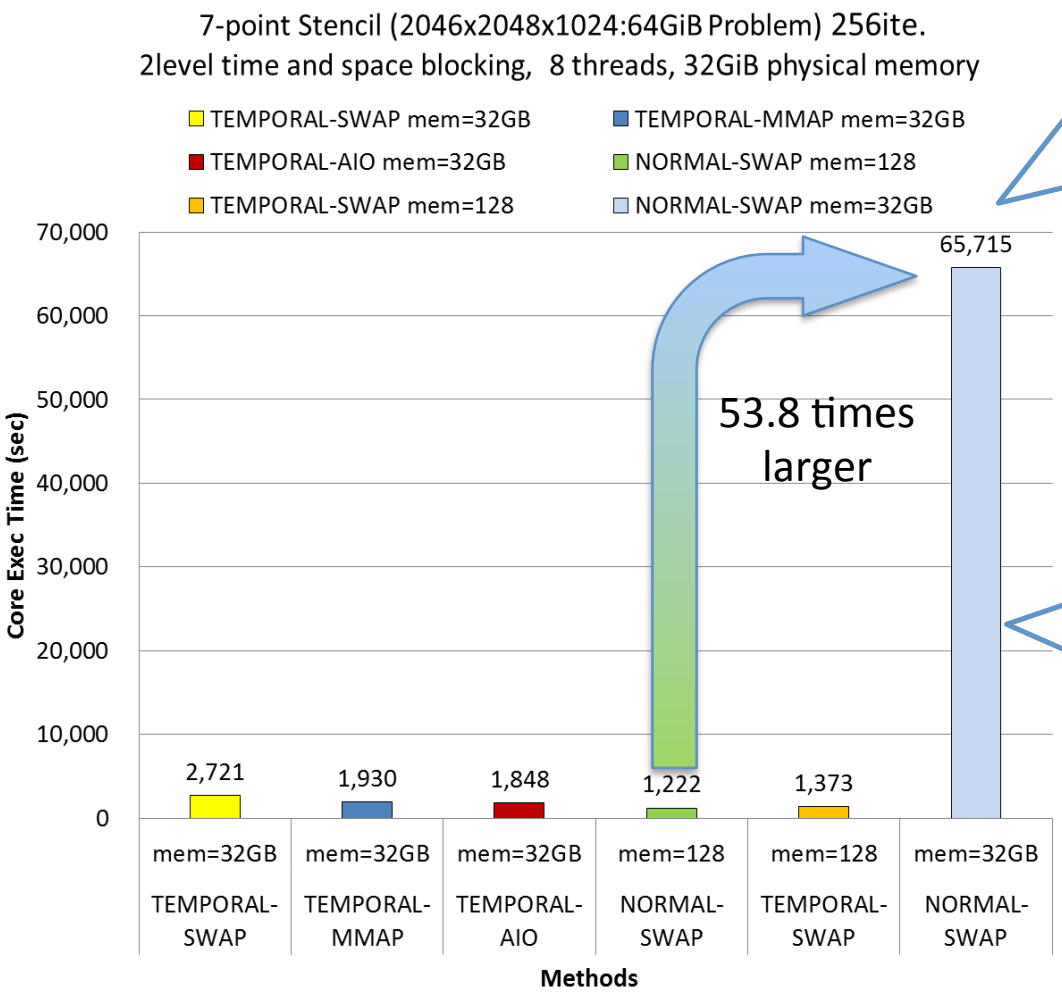
With spatial blocking and temporal blocking

- TEMPORAL-SWAP: 2.22 times larger
- TEMPORAL-MMAP: 1.58 times larger
- TEMPORAL-AIO: 1.51 times larger

7-point Stencil
(2046x2048x1024:64GiB Problem) 256ite.
2level time and space blocking,
8 threads, 32GiB physical memory

CPU	Xeon E5-2650 2.00GHz x1 (8cores)
Memory	32GiB boot (Total memory 128GiB) 16GiB(DDR3 1600MHz ECC Reg) x 8
OS kernel	3.13.0
Compiler	gcc 4.4.7 20120313 -O3
SwapDevice	ioDrive2 (FusionIO)

The Result without Temporal Blocking

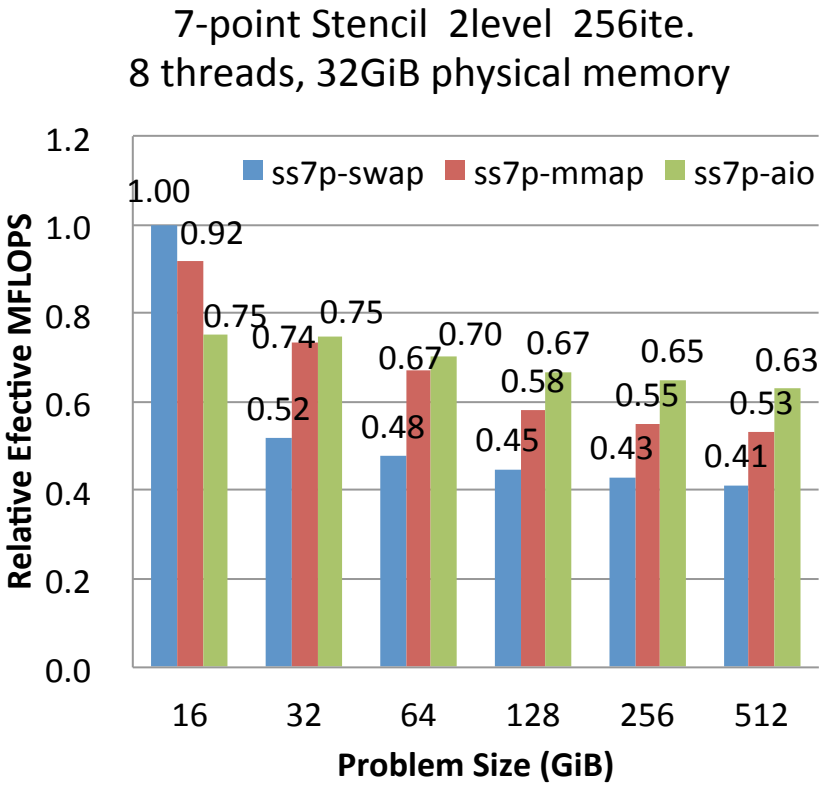


NORMAL-SWAP on 32GB memory with only spatial blocking takes 18 hours 16 minutes, 53.8 times longer time compared with the normal exec. time on 128GB.

NORMAL-SWAP employs only spatial blocking (32x32) for cache hit

7-point Stencil (2046x2048x1024:64GiB Problem) 256ite.
2level time and space blocking, 8 threads, 32GiB physical memory

7-point stencil computation Problem size vs. Performance on limited-size DRAM



	Spatial block size		2046 x 512 x 512		
	Temporal block size		128		
Problem Size (GiB)	Domain Shape	2 Buff Size (GiB)	Actual one Block Size (GiB)	Actual one Buffer Size (GiB)	Actual Total Size(GiB)
16	1022 x 1024 x 1024	16*	5.62	8.03	27.34
32	2046 x 1024 x 1024	32	10.12	16.06	52.36
64	2046 x 2048 x 1024	64	10.12	32.09	84.42
128	2046 x 2048 x 2048	128	10.12	64.13	148.48
256	2046 x 2048 x 4096	256	10.12	128.19	276.61
512	2046 x 4096 x 4096	512	10.12	256.25	532.73

16* : Spatial block size 1022 x 512 x 512

Flash SSD as Large and Slow Memory

- swap vs. mmap for APIs
  - Usually mmap performs better than swap
  - Fastswap + madvice sometimes gains good performance.
  - Both are easy to use.
- Linux kernel asynchronous IO: libaio
  - Much better perf. than the POSIX aio for multi-core systems. Higher level of parallelism & non_blocking IO
  - Block alignment: A limitation in program data structures.
- Application specific locality-aware algorithm
  - Flash SSDs are available for large and slow memory in practical use