System Software Researches Towards Next-Generation Supercomputing

TOKYO TECH Pursuing Excellence

Dealing with Deeper Memory Hierarchy

Overview of Project

On Exa-scale supercomputers, the ***"Memory Wall"*** problem will become even more severe, which prevents the realization of ***Extremely Fast&Big Simulations***.
This project promotes research towards this problem via co-design approach among application algorithms, system software, architecture.

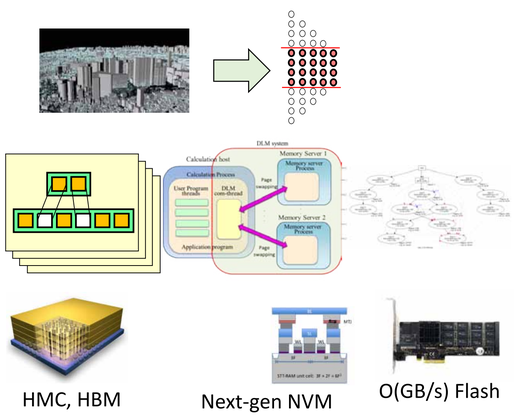
Comm/BW reducing algorithms

+

System software for mem hierarchy mgmt

+

HPC Architecture with hybrid memory devices

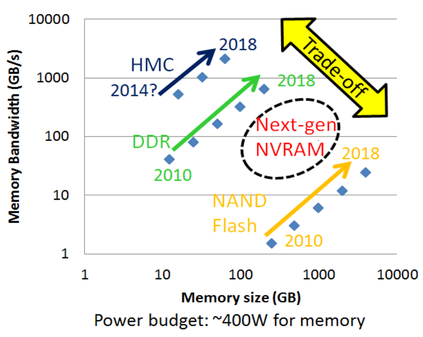


HMC, HBM　Next-gen NVM　O(GB/s) Flash

Target: Realizing ***extremely Fast&Big simulations*** of O(100PB/s) & O(100PB) in Exa-scale era

Target Architecture

Deeper memory hierarchy that consists of heterogeneous memory devices



Power budget: ~400W for memory

Hybrid Memory Cube (HMC):
DRAM chips are stacked with TSV technology. It will have advantage in bandwidth over DDR, but capacity will be smaller.

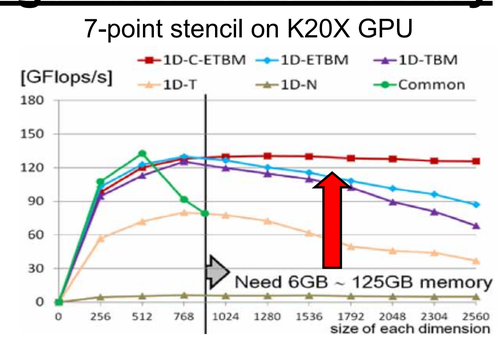
NAND Flash:
SSDs are already commodity. Newer products, such as IO-drive have O(GB/s) bandwidth.

Next-gen non-volatile RAM (NVRAM):
Several kinds of NVRAM such as STT-MRAM, ReRAM, FeRAM, etc, will be available in a few years.

Highly Optimized Stencils Larger than GPU Memory

For extremely large stencil simulations, we implemented temporal blocking (TB) technique and clever optimizations on GPUs [1][2].

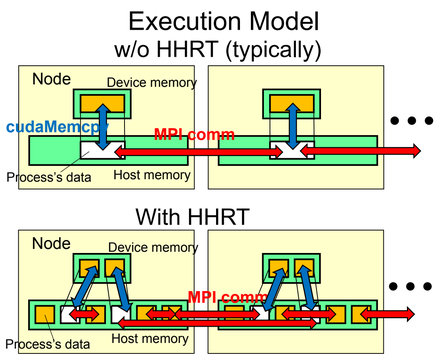
- Eliminating redundant computation
- Reducing memory footprint of TB algorithm



HHRT: System Software for GPU Memory Swap

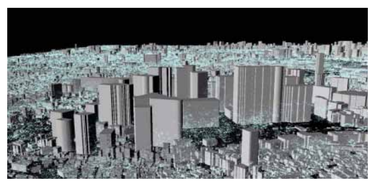
For easier programming, we implemented system software, named HHRT (hybrid hierarchical runtime) [3].

- HHRT supports user programs written in MPI and CUDA with little modification
- Oversubscription based execution model
- HHRT implicitly supports memory swapping between GPU memory and host

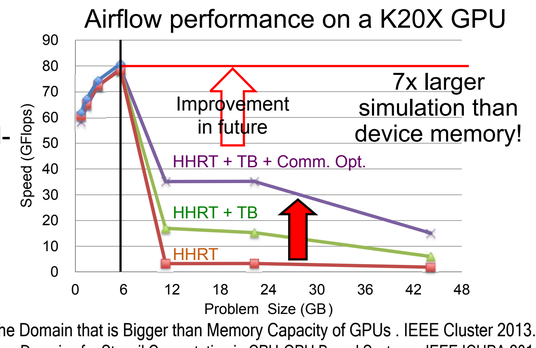


Integration with Real Simulation Application

We integrated our techniques with the city airflow simulation.



Original code on MPI+CUDA was developed by Naoyuki Onodera, Tokyo Tech. We integrated TB into it and executed on HHRT.



[1] G. Jin, T. Endo, S. Matsuoka. A Parallel Optimization Method for Stencil Computation on the Domain that is Bigger than Memory Capacity of GPUs . IEEE Cluster 2013.
[2] G. Jin, J. Lin, T. Endo. Efficient Utilization of Memory Hierarchy to Enable the Computation on Bigger Domains for Stencil Computation in CPU-GPU Based Systems. IEEE ICHPA 2014.
[3] T. Endo, G. Jin: Software Technologies Coping with Memory Hierarchy of GPGPU Clusters for Stencil Computations. IEEE Cluster 2014.

PI: Toshio Endo (endo@is.titech.ac.jp), supported by JST-CREST

Extreme Scale Resilience

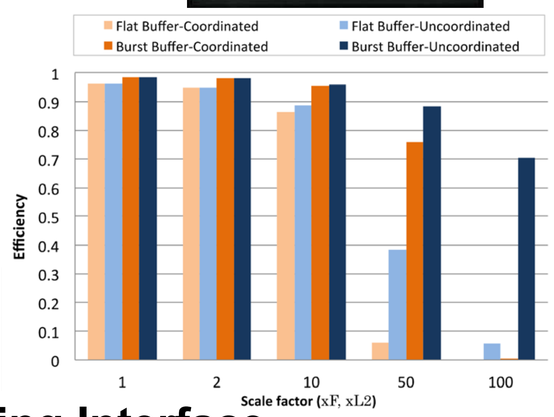
Collaborative work with Lawrence Livermore National Laboratory

Multi-tiered Burst Buffer Storage and Modeling

We developed a user-level InfiniBand-based I/O interface (IBIO), and explored how burst buffers can improve system efficiency. This work won IEEE/ACM CCGrid2014 best paper award through joint works with LLNL.

IEEE/ACM CCGrid2014 Best Paper Award

mSATA × 8　Adaptec RAID × 1



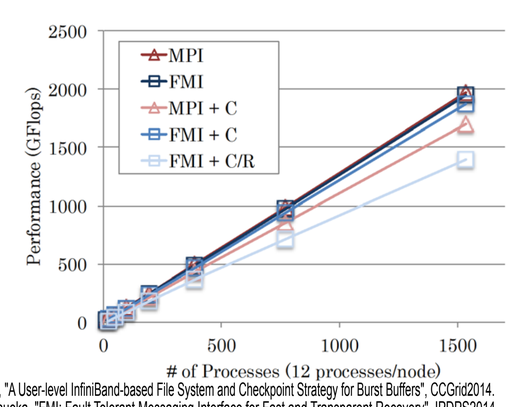
SSD	Crucial m4 mSATA 256GB CT256M4SSD3 (Peak read: 500MB/s, Peak write: 260MB/s
SATA Converter	KOUTECH IO-ASS110 mSATA to 2.5' SATA Device Converter with Metal Fram
RAID Card	Adaptec RAID 7805Q ASR-7805Q Single

FMI: Fault Tolerant Messaging Interface

FMI is an MPI-like survivable messaging interface that enables scalable failure detection, dynamic node allocation, fast and transparent recovery.

```
int main (int *argc, char *argv[]) {
  FMI_Init(&argc, &argv);
  FMI_Comm_rank(FMI_COMM_WORLD, &rank);
  /* Application's initialization */
  while ((n = FMI_Loop(…)) < numloop) {
    /* Application's program */
  }
  /* Application's finalization */
  FMI_Finalize();
}
```

FMI_Loop (void *ckpt, size_t *sizes, int length)



[1] Kento Sato, Kathryn Mohror, Adam Moody, Todd Gamblin, Bronis R. de Supinski, Naoya Maruyama and Satoshi Matsuoka, "A User-level InfiniBand-based File System and Checkpoint Strategy for Burst Buffers", CCGrid2014.
[2] Kento Sato, Adam Moody, Kathryn Mohror, Todd Gamblin, Bronis R. de Supinski, Naoya Maruyama and Satoshi Matsuoka, "FMI: Fault Tolerant Messaging Interface for Fast and Transparent Recovery", IPDPS2014.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52- 07NA27344. LLNL-POST-661083
This work was supported by JSPS Grants-in-Aid for Scientific Research: Grant Number 22320003.

Application Framework

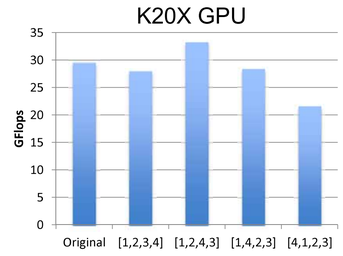
Collaborative work with RIKEN AICS and Oak Ridge National Laboratory

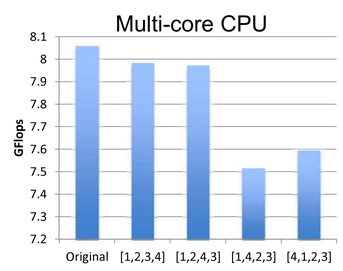
Data Layout Transformation for Performance Portability

Data layout transformation is effective optimization for accelerators but it is also one of the cause of the low performance portability. We propose OpenACC directive extension for data layout transformation. For example,

```
#pragma acc transform \
        transpose(A[0:Z][0:Y][0:X][0:4]::[4,1,2,3])
```

transforms array A[Z][Y][X][4] to A'[4][Z][Y][X]. We implement a translator and evaluate it with Himeno Benchmark (3D stencil)

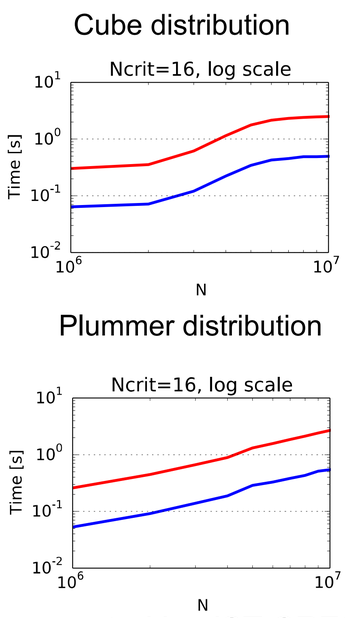




[1] T. Hoshino, N. Maruyama, S. Matsuoka. An OpenACC Extension for Data Layout Transformation. WACCPD 2014 in conjunction with SC14.

Performance modeling of FMM for arbitrary particle distributions

FMM is a hierarchical N-body algorithm, of which computation time highly depends on input data (=particle distribution) and algorithmic parameter (Ncrit).
Conventional models based on math expressions cannot represent the highly data-dependent behavior. We develop a modeling technique based on Aspen, a modeling language.
The composability and flexibility allows represent complicated control flow of FMM.



PI: Naoya Maruyama (nmaruyama@riken.jp), supported by JST-CREST

http://www.gsic.titech.ac.jp/sc14