CREST
Core Research for Evolutionary Science and Technology

JST Japan Science and Technology Agency

Research Area : Development of System Software Technologies for post-Peta Scale High Performance Computing

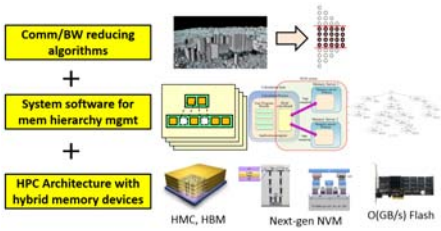
Software Technology that Deals with Deeper Memory Hierarchy in Post-petascale Era

Toshio Endo[1] Yukinori Sato[2] Hiroko Midorikawa[3]

[1]Tokyo Tech [2]JAIST [3]Seikei University

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.



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

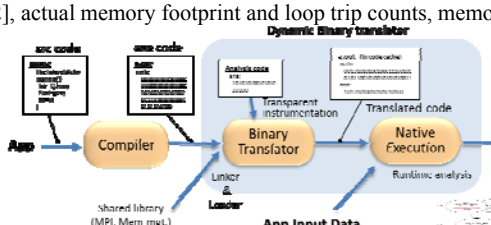
Hybrid Memory Cube (HMC): DRAM chips are stacked with TSV technology. It will have advantage in bandwidth over DDR, but capacity will be smaller.

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.

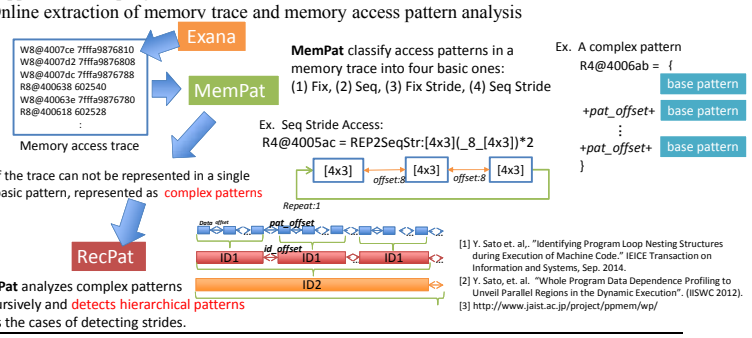
Exana: Application Profiling and Optimization Infrastructure (Sato Group, JAIST)

We have been developing the ***Exana*** for accelerating system with deeply hierarchical memory. Using an already complied executable binary code and its input data set as its input, the Exana can profile various application behaviors such as precise loop nest structures [1], data dependencies among loop regions [2], actual memory footprint and loop trip counts, memory access patterns.



The key features:

- Aiming for Static and dynamic analysis without specific compiler
  - Accept application package's original makefiles and highly optimized compiler options
- Analysis for shared libraries dynamically loaded at runtime
- Support for MPI programs
- Online extraction of memory trace and memory access pattern analysis



[1] Y. Sato et. al,. "Identifying Program Loop Nesting Structures during Execution of Machine Code." IEICE Transaction on Information and Systems, Sep. 2014.
[2] Y. Sato, et. al. "Whole Program Data Dependence Profiling to Unveil Parallel Regions in the Dynamic Execution". (IISWC 2012).
[3] http://www.jaist.ac.jp/project/ppmem/wp/

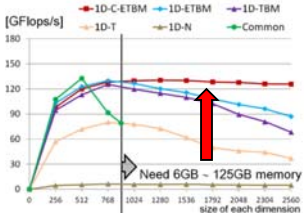
Integration of Application Algorithms, System Software and Architecture for Large Data Applications (Endo Group)

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

7-point stencil on a K20X GPU

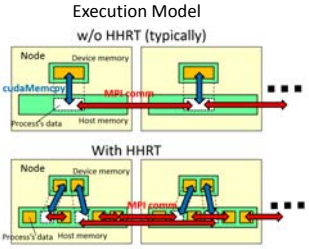


HHRT: System Software for GPU Memory Swap

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

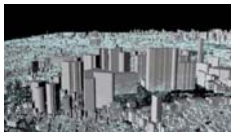
- HHRT supports user programs written in MPI and CUDA with little modification
- Oversubscription based execution model
- HHRT implicitly supports memory swapping between GPU memory and host

Execution Model



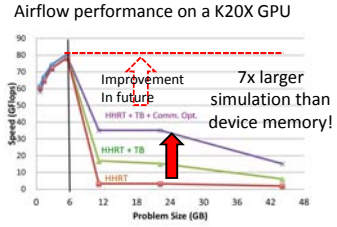
Integration with Real Simulation Application

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.

Airflow performance on a K20X GPU

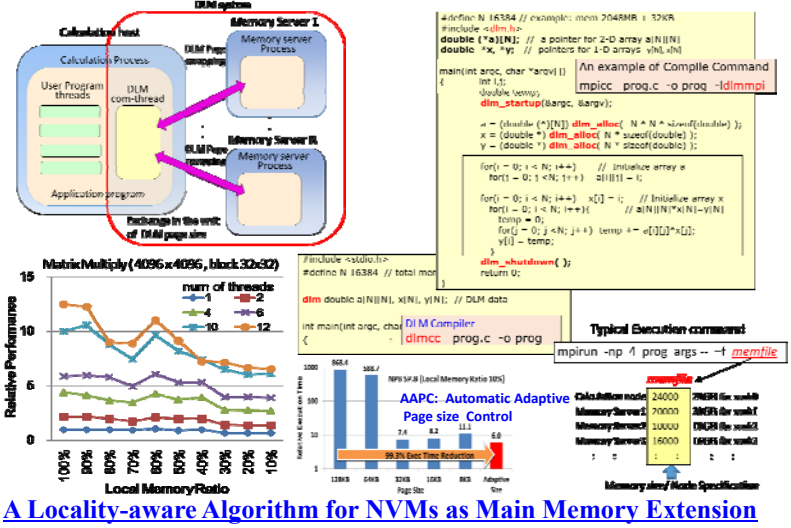


[1] G. Jin, T. Endo, S. Matsuoka. A Parallel Optimization Method for Stencil Computation on the Domain that is Bigger than Memory Capacity of GPUs . IEEE Cluster 2013.
[2] G. Jin, J. Lin, T. Endo. Efficient Utilization of Memory Hierarchy to Enable the Computation on Bigger Domains for Stencil Computation 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.

Horizontal and Vertical Memory Extensions for Large Data Applications (Midorikawa Group)

User-level Remote Memory Paging: *DLM* (Distributed Large Memory)

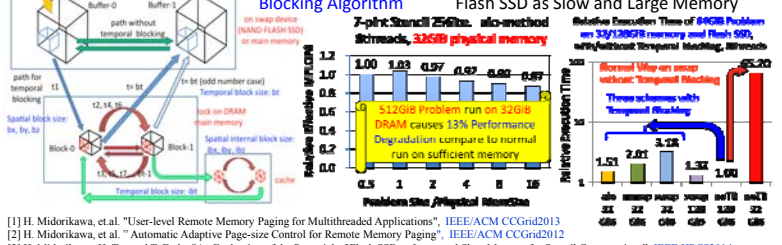
DLM offers a virtual large memory using distributed node memories in a cluster for multithreaded applications (OpenMP and pthread programs).



A Locality-aware Algorithm for NVMs as Main Memory Extension

Spatial and Temporal Blocking Algorithm

Stencil Computation using a PCIe-based Flash SSD as Slow and Large Memory



[1] H. Midorikawa, et.al. "User-level Remote Memory Paging for Multithreaded Applications", IEEE/ACM CCGrid2013
[2] H. Midorikawa, et al. " Automatic Adaptive Page-size Control for Remote Memory Paging", IEEE/ACM CCGrid2012
[3] H. Midorikawa, H. Tan and T. Endo, "An Evaluation of the Potential of Flash SSD as Large and Slow Memory for Stencil Computations", IEEE HPCS2014 ,
[4] H. Midorikawa, "Using a Flash as Large and Slow Memory for Stencil Computations", Flash Memory Summit 2014