CREST
Core Research for Evolutionary Science and Technology

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

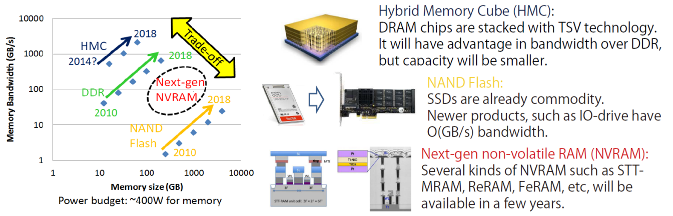
Software Technology that Deals with Deeper Memory Hierarchy in Post-petascale Era

Achieving Extremely Big&Fast Computations on Post-Petascale Supercomputers

Overview

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 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.

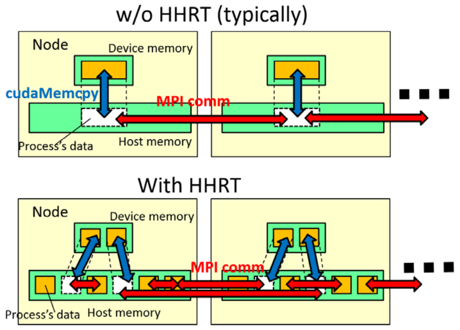
Next-gen non-volatile RAM (NVRAM):
Several kinds of NVRAM such as STT-MRAM, ReRAM, FeRAM, etc, will be available in a few years.

Integration of Application Algorithms, System Software and Architecture for Large Data Applications

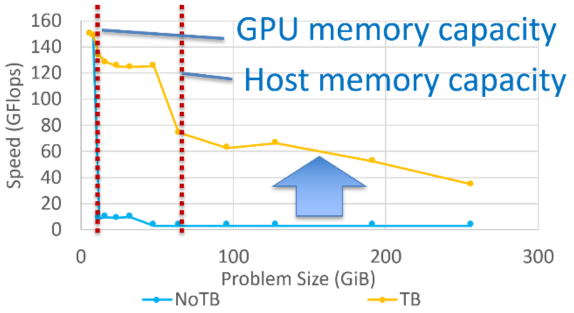
HHRT: System Software for GPU Memory Swap:
For easier programming, we implemented system software, HHRT (hybrid hierarchical runtime)

- HHRT automatically supports data swapping among three memory layers, GPU memory ⇔ Host memory ⇔ Flash SSD.
- HHRT supports "process-wise" swapping, not "page-wise" like OS.
- On the other hand, users still have responsibility to improve locality for better performance, such as Temporal blocking (TB) for stencils

Execution Model

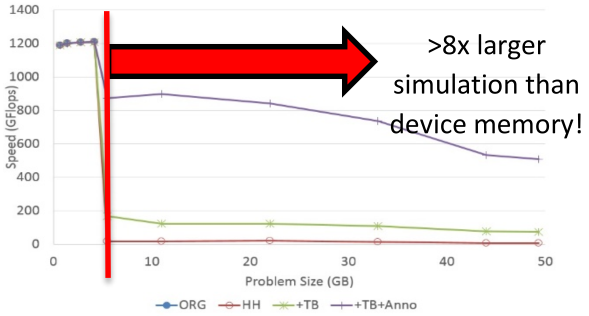


Performance on 7p-stencil with TB on a K40 GPU & 950 pro m.2 SSD



Integration with Real Simulation Application:
We integrated our techniques with the dendrite simulation [Shimokawabe et al. SC11]. By integrating the implementation written in CUDA+MPI with HHRT and temporal blocking technique, >8x larger simulation is achieved with moderate overhead.

Dendrite performance on a K20X GPU



Tool chains for Memory Locality Profiling and Performance Tuning

We have been developing tool chains for accelerating system with deeply hierarchical memory. Starting with source code or its executable code, these tools profile and translate applications for the underlying memory subsystems.

Exana is a memory locality profiler and show runtime information of dynamic application behaviors such as precise call and loop nest structures [1], data dependencies among loop regions [2], actual memory footprint and loop trip counts, memory access patterns. Exana-C2Sim is a cache-line conflict detector and can analyze root cases of the line conflicts. ExanaDBT is a dynamic compilation system for transparent optimization at runtime based on

Japan Science and Technology Agency

HHRT (Hybrid Hierarchical RunTime)

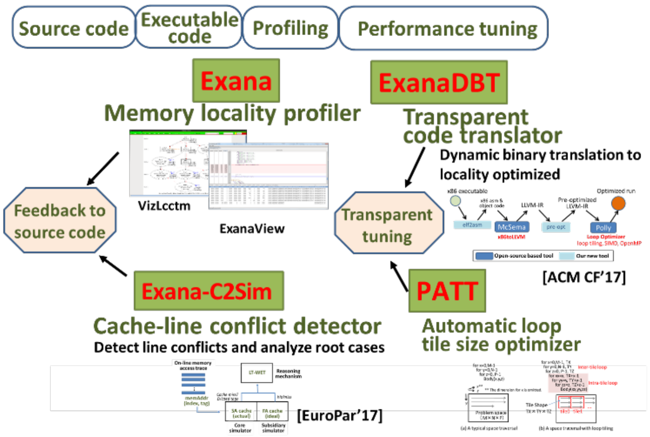
Objective: HHRT is a wrapper library of MPI to expand memory capacity visible to user applications. It supports CUDA+MPI applications. Now Xeon-Phi version is under testing.

https://github.com/toshioendo/hhrt

Exana series

Objective: To assist or automate performance tuning for deeply hierarchical memory, we developed tool chains composed of Exana, C2Sim, ExanaDBT. We released Exana and Exana-C2Sim on GitHub:

https://github.com/YukinoriSato/ExanaPkg



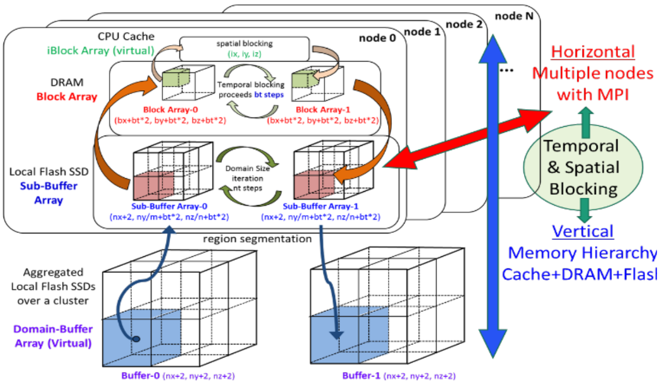
polyhedral optimization technique. PATT is an automatic loop tile size optimizer for tuning the underlying hierarchical memories.

We believe these tool chains enable us to perform automatic/semi-automatic optimizations for deeply ierarchical memory and can contribute to productive software performance engineering.

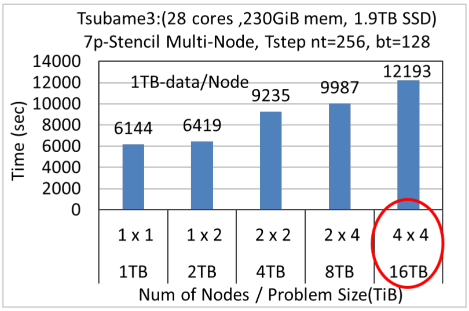
For more information, please visit our project web page: [http://www.el.gsic.titech.ac.jp/~yukinori/Exana.html]

Horizontal and Vertical Memory Extensions for Large data Applications

Large-scale Stencil Computations using Distributed Flash SSDs and Memories in a Cluster


CPU Cache
iBlock Array (virtual)
spatial blocking (ix, iy, iz)
node 0
node 1
node 2
node N
DRAM
Block Array
Temporal blocking proceeds bt steps
Block Array-0 (bx+bt*2, by+bt*2, bz+bt*2)
Block Array-1 (bx+bt*2, by+bt*2, bz+bt*2)
Local Flash SSD
Sub-Buffer Array
Domain Size iteration nt steps
Sub-Buffer Array-0 (nx+2, ny/m+bt*2, nz/n+bt*2)
Sub-Buffer Array-1 (nx+2, ny/m+bt*2, nz/n+bt*2)
region segmentation
Aggregated Local Flash SSDs over a cluster
Domain-Buffer Array (Virtual)
Buffer-0 (nx+2, ny+2, nz+2)
Buffer-1 (nx+2, ny+2, nz+2)
Horizontal
Multiple nodes with MPI
Temporal & Spatial Blocking
Vertical
Memory Hierarchy
Cache+DRAM+Flasl

The 1000-time latency gap between DRAM and flash is overcome by our advanced implementation using highly parallel AIO and a novel temporal blocking algorithm designed for flash. The available maximum problem size is not limited by the total capacity of DRAMs in a cluster. It can be expanded to the total capacity of distributed flash SSDs in a cluster. When using Tsubame3 (256 GB-DRAM, 2 TB-Flash / node), only 16 nodes are sufficient for 16 TB stencil problem, which usually requires 128 nodes when using only DRAM without our implementation.

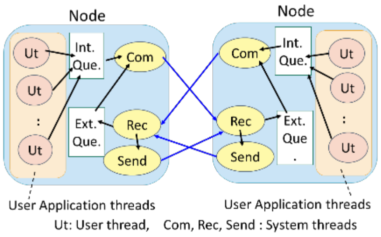


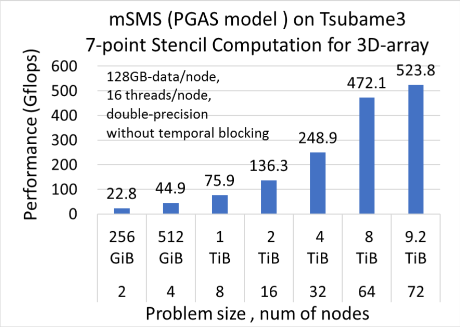
Blk-Tune (Just-in-Time Automatic Blocking Size Setting System for Flash-based Stencil Computations)

Blk-Tune automatically retrieves platform information and finds the globally optimal spatial / temporal blocking sizes to minimize the amount of I/O traffic to the flash SSD in run-time. It realizes just-in-time selection by a search algorithm without any preliminary executions, which differentiates the Blk-Tune from other auto-tuning systems.

mSMS (SDSM for General Purpose PGAS Model)

mSMS provides a transparent full-accessible globally shared memory distributed over multiple nodes with a data distribution API. It realizes highly flexible and productive parallel programming environment with OpenMP / pthread interface for each node with node identifiers. In a preliminary evaluation using Tsubame3, a naive stencil computation for large-size (9.2 TiB) problem can be easily implemented over 72 nodes, which achieves 524 Gflops.





Project information:

Project leader: Toshio Endo (Tokyo Tech)

Project member: Hiroko Midorikawa (Seikei Univ.), Yukinori Sato, Shimpei Sato, Noboru Tanabe (Tokyo Tech),

Contact: **endo@is.titech.ac.jp**