MemSpan: Toward Efficient Data Placement in Heterogeneous Memory Systems

Thaleia-Dimitra Doudali, Ada Gavrilovska
{tdoudali, ada}@cc.gatech.edu

1. Motivation

- Big Data and HPC applications require fast processing speeds.
- Commodity hardware (CPU, DRAM, Disk) face cost and scale issues.
- New hardware technologies emerge.
  - Processing Units
  - Memory Units
  - Storage Units
- New hardware technologies will operate together with the predominant ones.
  - Heterogeneous systems is the new norm.

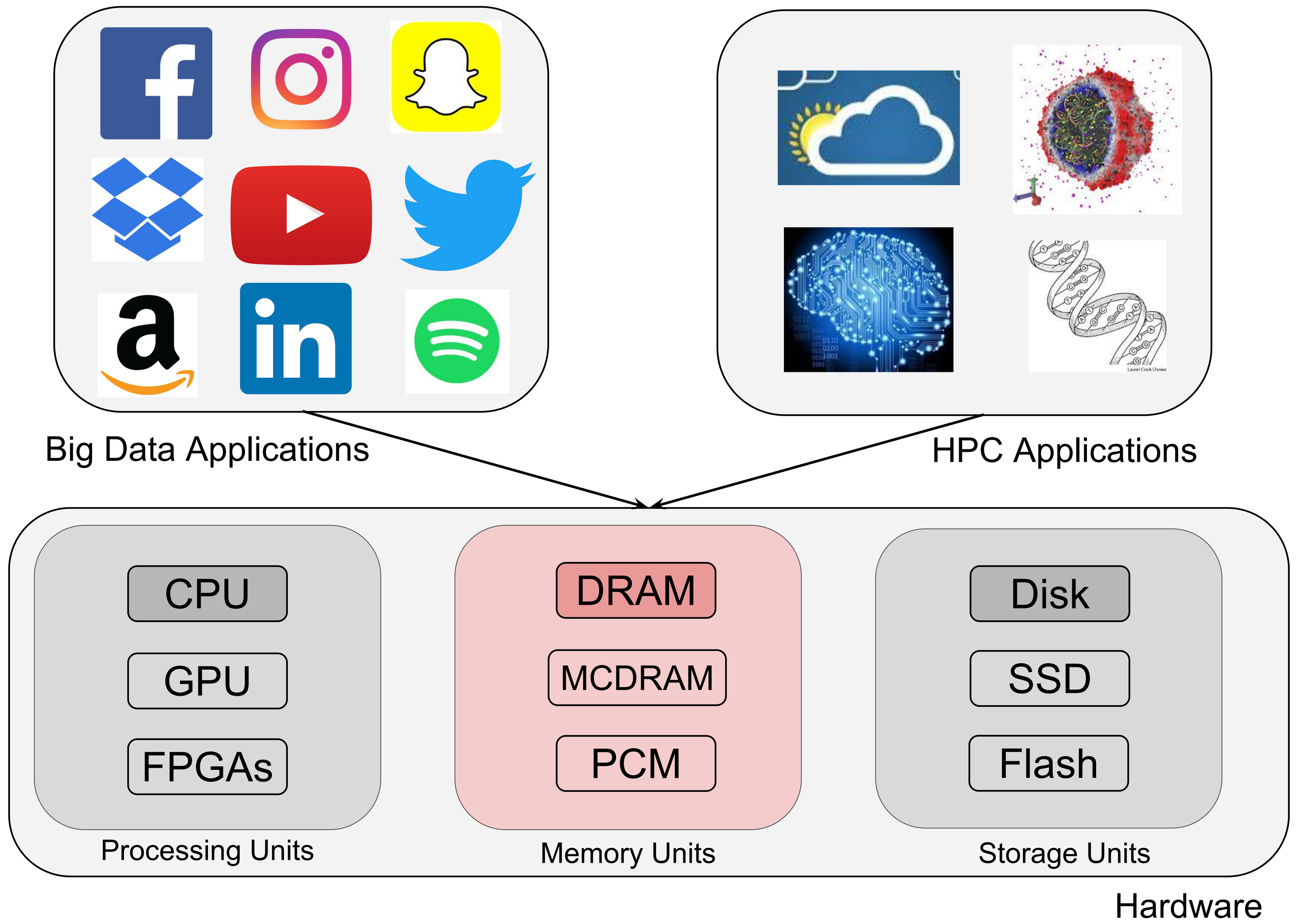

Figure 1: Data centers and High Performance Computing environments are going to couple heterogeneous hardware units with different characteristics, so as to enable faster computation, which the predominant hardware cannot provide anymore by itself.

2. Problem Statement

- The memory substrate will couple small amounts of "fast" access units, with larger portions of "slower" access units.
- Data-intensive applications will span their dataset across all available memory units.
- Need to cleverly map application's data across the different memory components.
  - Critical to performance data should be in fast memory.
- Need for an OS-level solution.
  - No changes needed in the source code.
  - Can work for any type of application.

3. Existing Solutions

- Explicit API to allocate memory on the new units.
  - Needs programming efforts.
- Application profiling tools. [1] [2]
  - Offline run: keeps track of number of accesses on a data structure / object granularity.
  - cost = number of accesses per byte.
  - Order data objects according to cost.
  - Online run: places data objects with high cost in fast memory, until capacity is full.

[1] Subramanya R. Dulloor, Amitabha Roy, Zheguang Zhao, Narayanan Sundaram, Nadathur Satish, Rajesh Sankaran, Jeff Jackson, and Karsten Schwan. 2016. Data tiering in heterogeneous memory systems. In *Proceedings of the Eleventh European Conference on Computer Systems* (EuroSys '16). ACM, New York, NY, USA, , Article 15 , 16 pages. DOI=http://dx.doi.org/10.1145/2901318.2901344

[2] Du Shen, Xu Liu, and Felix Xiaozhu Lin. 2016. Characterizing emerging heterogeneous memory. In *Proceedings of the 2016 ACM SIGPLAN International Symposium on Memory Management* (ISMM 2016). ACM, New York, NY, USA, 13-23. DOI: https://doi.org/10.1145/2926697.2926702

4. Observations

1. Not all applications are sensitive to the presence of slower memory.

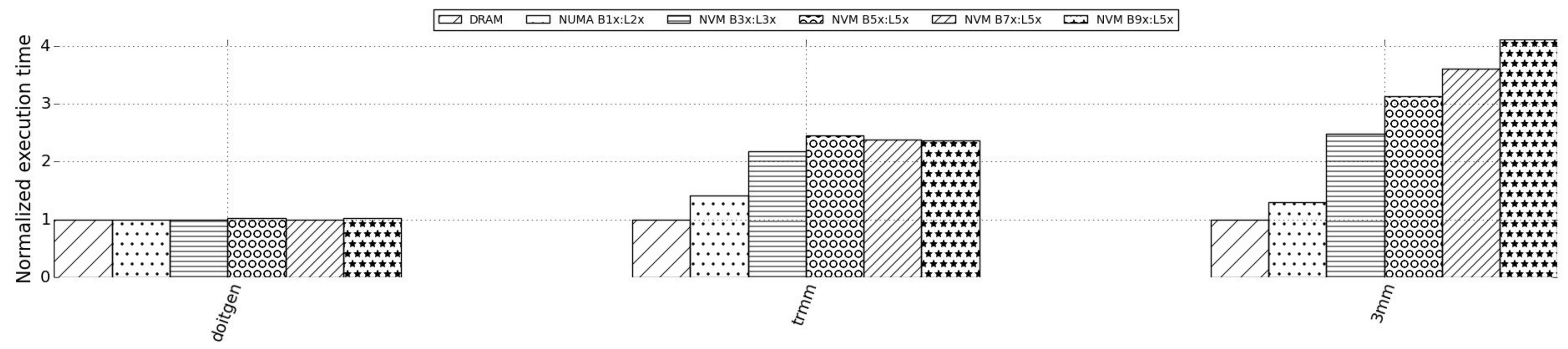

Figure 2: Execution of 3 kernels from Polybench/C benchmark suite on an "all-fast" memory environment (DRAM) versus an "all-slow" setup, for different scales of memory access latency and bandwidth slowdown.

2. Not all the data structures of an application are critical to performance.

Objects allocated in fast memory	*Execution Time*	*Cost / Benefit*
All objects	56.44 s	1
Matrix A	130.25 s	0.04
Matrix B	56.98 s	0.99
None	133.05 s	0

Table 1: For the triangular matrix multiply kernel (trmm) *B = B * A*, placing array A in fast memory contributes only by a factor of 0.04 to the overall performance. However, placing matrix B guarantees performance almost same to an "all-fast" configuration.

5. Proposal

- Build a tool that can on-the-fly:
  - Identify if an application overall is sensitive to slow memory.
  - Identify which data structures are critical to performance.
  - Place in slow memory the non-sensitive components.
  - Prioritize the placement of data objects with high cost in fast memory, until capacity is full, similarly to existing work.
- In this way, we can:
  - Leverage the existence of slow memory.
  - Make existing solutions simpler / faster; can become practical for dynamic workloads.
  - Provide fairness or SLA guarantees in multi-tenant environments, where applications will compete for the available fast memory.