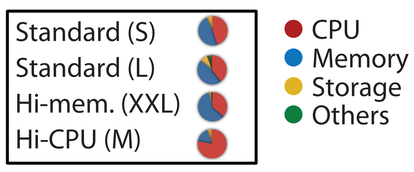
Memory-Effcient GroupBy-Aggregate using Compressed Buffer Trees

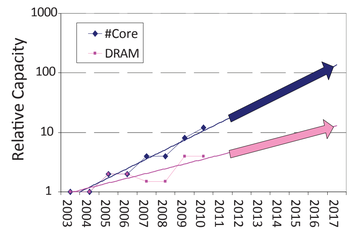
Hrishikesh Amur (GT), Wolfgang Richter (CMU), David G. Andersen (CMU), Michael Kaminsky (Intel),
Karsten Schwan (GT), Athula Balachandran (CMU), Erik Zawadzki (CMU)

NEED FOR MEMORY-EFFICIENCY

- **Cost of DRAM**
- **Memory capacity per core decreasing [Lim, et al. ISCA'09]**
- **Future architectures including 3D-stacked DRAM have capacity constraints [Loh, et al., SHAW'12]**

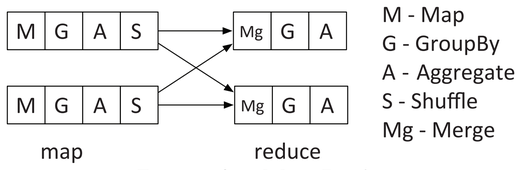


Amazon EC2 resource costs using linear regression from Nov'12



Memory capacity per core extrapolated to decrease [Lim et al. ISCA'09]

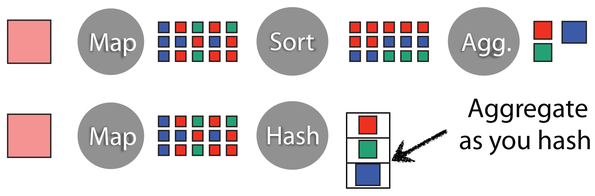
GROUPBY-AGGREGATE



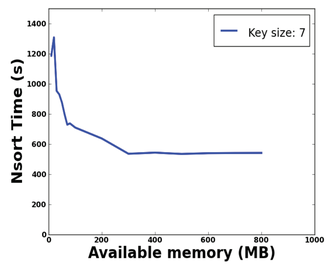
Example: MapReduce

Sort vs. Hash-based GroupBy-Aggregate

- **Google's MapReduce and Hadoop use *sorting* as the GroupBy operator**
- ***Hash*-based grouping (common in RDBMS) performs better for many MapReduce applications. [Yu, Gunda, Isard, SOSP'09], [Li, et al. SIGMOD'11]**



Sort- vs. Hash-based GA



External sorting works well with limited memory

Allocator	Per-entry memory (B)	
	unordered map	google sparsehash
hoard	64.9	67.8
tcmalloc	57.2	43
jemalloc	58.1	41

Implementation of hashing entails memory overheads; Dataset: key: 8B C-style string, value: 4B integer

Can hashtables be compressed?

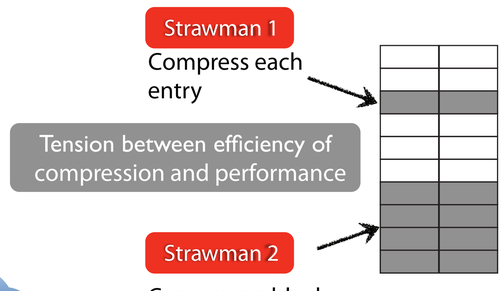
Strawman 1

Compress each entry

Tension between efficiency of compression and performance

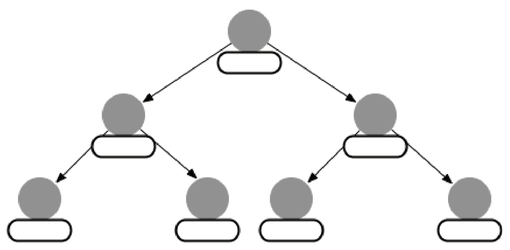
Strawman 2

Compress a block



COMPRESSED BUFFER TREES

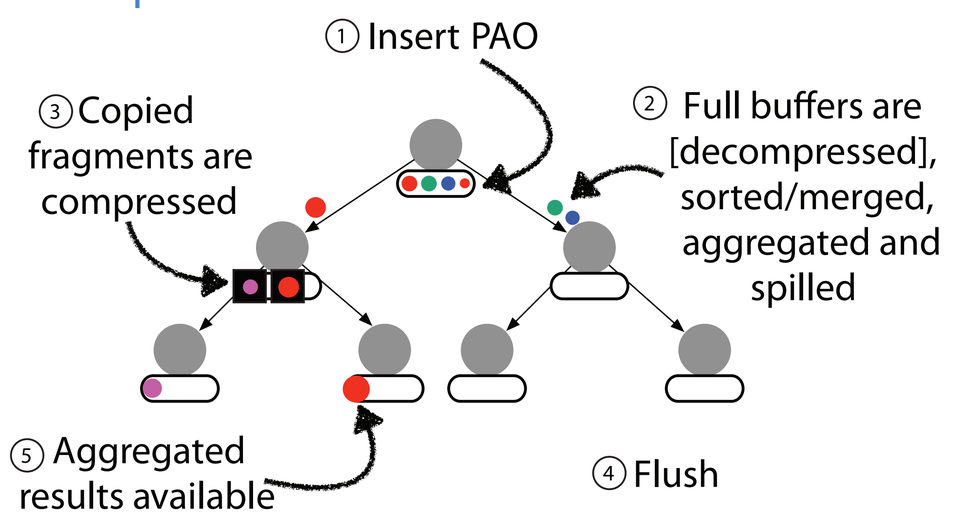
How to use compression to build memory-efficient and fast GroupBy-Aggregate?



The Compressed Buffer Tree

- **In-memory B-tree with each node augmented with a memory buffer**
- **Inspired by the buffer tree [Arge, Algorithmica'03]**

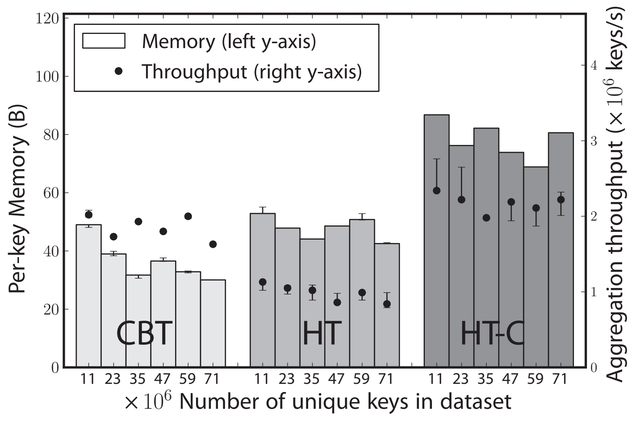
CBT Operation

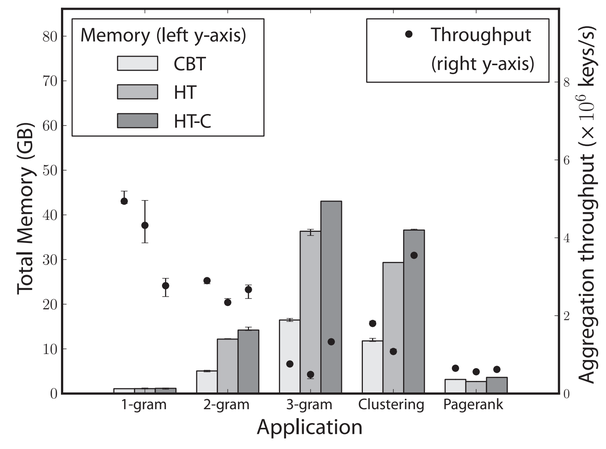


Memory efficiency through compression

Effectiveness through compressing large buffers

High performance through buffering





Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · PRINCETON UNIVERSITY · UC Berkeley