Lecture 20: Big Data, Memristors

- Today: architectures for big data, memristors

FAWN

Andersen et al., SOSP'09

- Targeted at key-value pair based workloads
- Such workloads demand high bandwidth for random data look-ups
- Each node tries to maximize query rate for a given power budget (3-6 W): this is achieved with low-EPI cores and Flash-based storage
- It uses a log-structured data store to make sure that writes are append-only and sequential
- In-DRAM hash tables help locate data on reads

Cost Analysis

- A workload has a given capacity and bandwidth requirement – once you create a node, you'll need enough nodes to meet these requirements
- The cost for a node is the sum of CapEx and OpEx: low-EPI processors help reduce both
- Workloads limited by capacity will benefit from disk and Flash, not DRAM; workloads limited by bw prefer DRAM

Cost Analysis

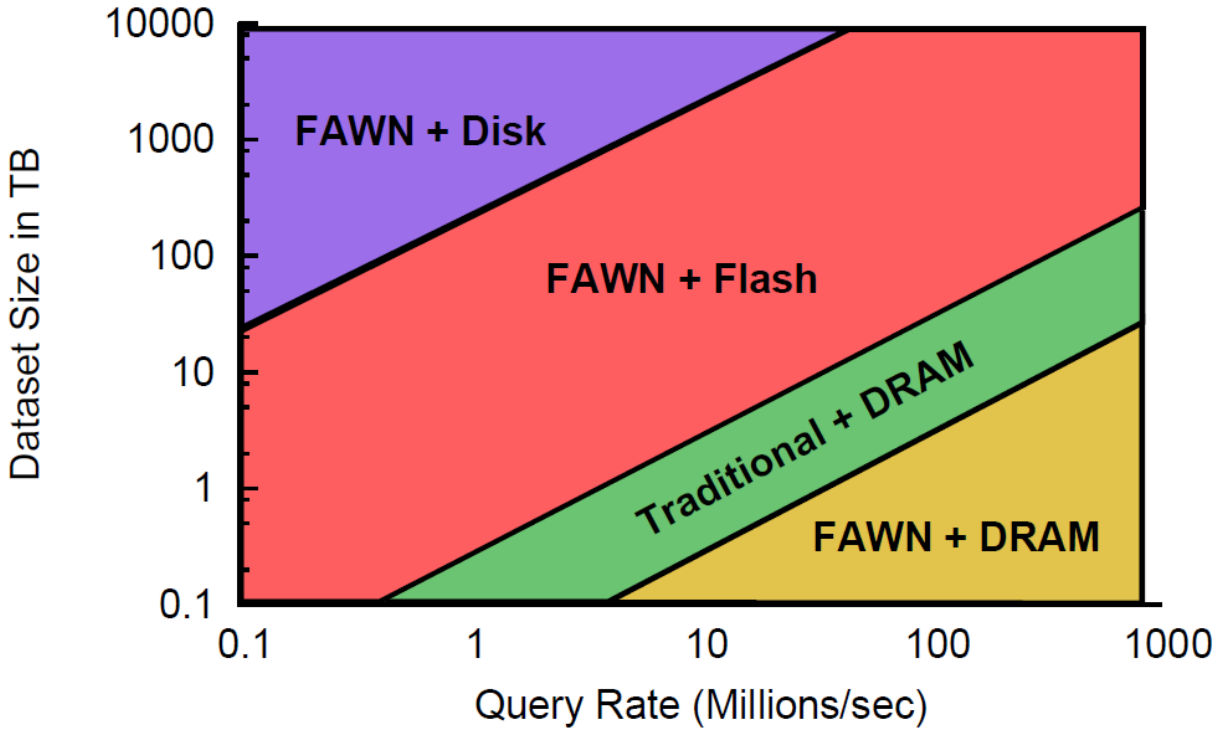


Figure 16: Solution space for lowest 3-year TCO as a function of dataset size and query rate.

RAMCloud

Ongaro et al., SOSP'11

- Data is stored in DRAM across several nodes
- DRAM volatility and frequent node failures in a large cluster force the use of data replication and data striping for fast recovery
- Replication is done in Flash in the background with log-based structures; logs are first placed in DRAM buffers in multiple nodes and periodically written to Flash

Thin Servers with Smart Pipes Lim et al., ISCA'13

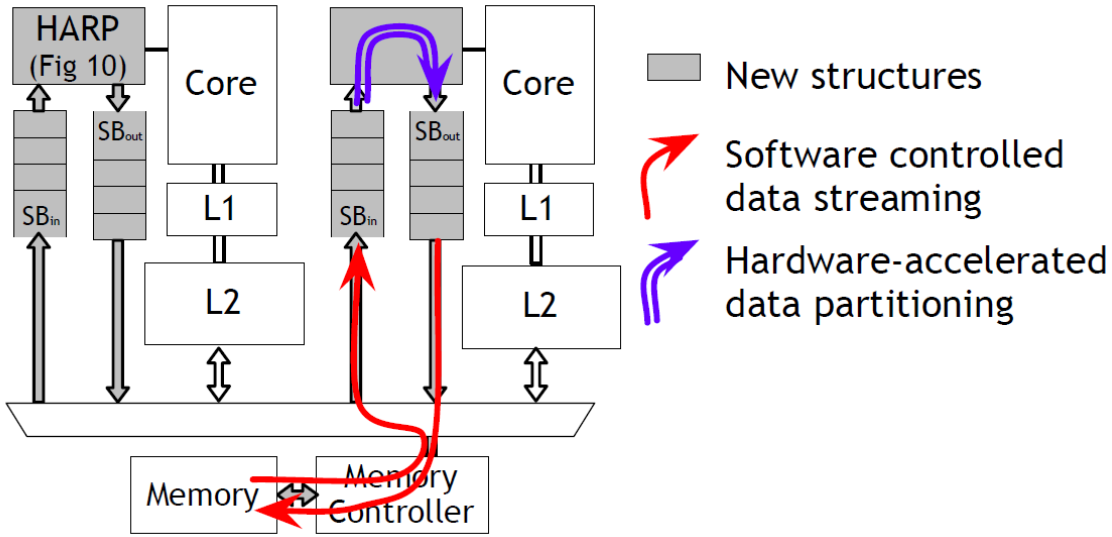
- Designs a custom architecture for Memcached
- Memcached: table of unique keys and their values stored in DRAM and managed with an LRU policy; GET and SET are used for reads and writes
- Modern Xeon or Atom based designs are both highly under-utilized, primarily because of per-packet processing overheads in the NIC and OS: high icache/itlb misses and branch mispredictions
- Customized NIC and accelerator take care of packet processing and partial hash table look-ups in hardware

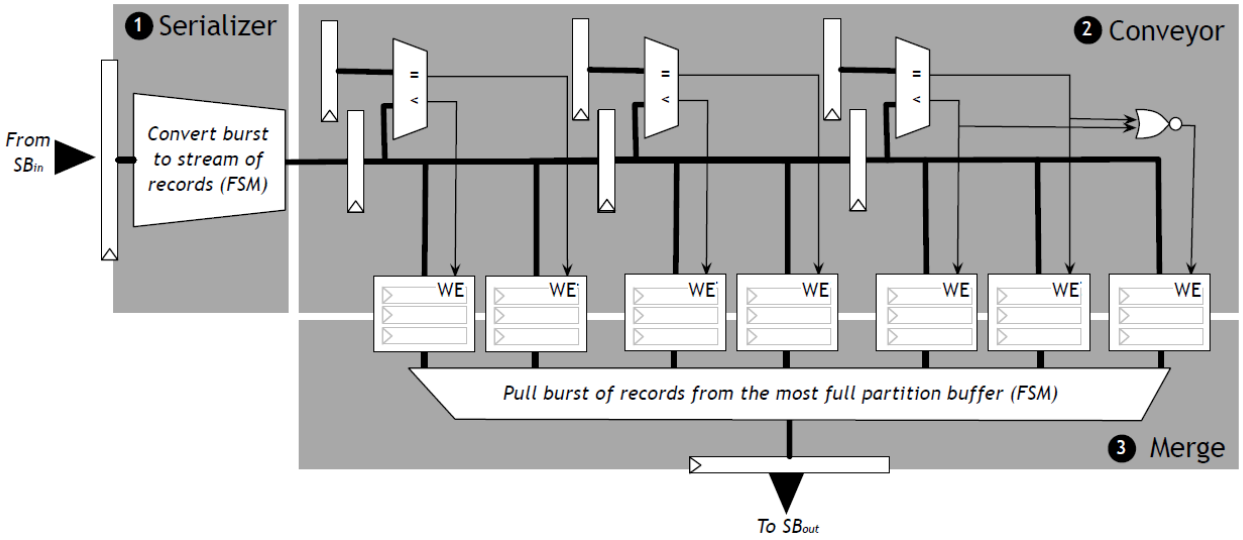
HARP

Wu et al., ISCA'13

- Data partitioning is an important component in many database operations, e.g., joins
- Range partitioning is one example, where partitions are defined by the ranges for a given key
- New software that simply loads/drains stream buffers and a new pipeline that reads in streamed inputs and partitions them into different buffers
- Full buffers get placed into the streamed outputs
- The partitioning logic is a pipeline of comparators

HARP


HARP
(Fig 10)
Core
Core
SBout
SBin
SBout
SBin
L1
L1
L2
L2
Memory
Memory
Controller
New structures
Software controlled
data streaming
Hardware-accelerated
data partitioning

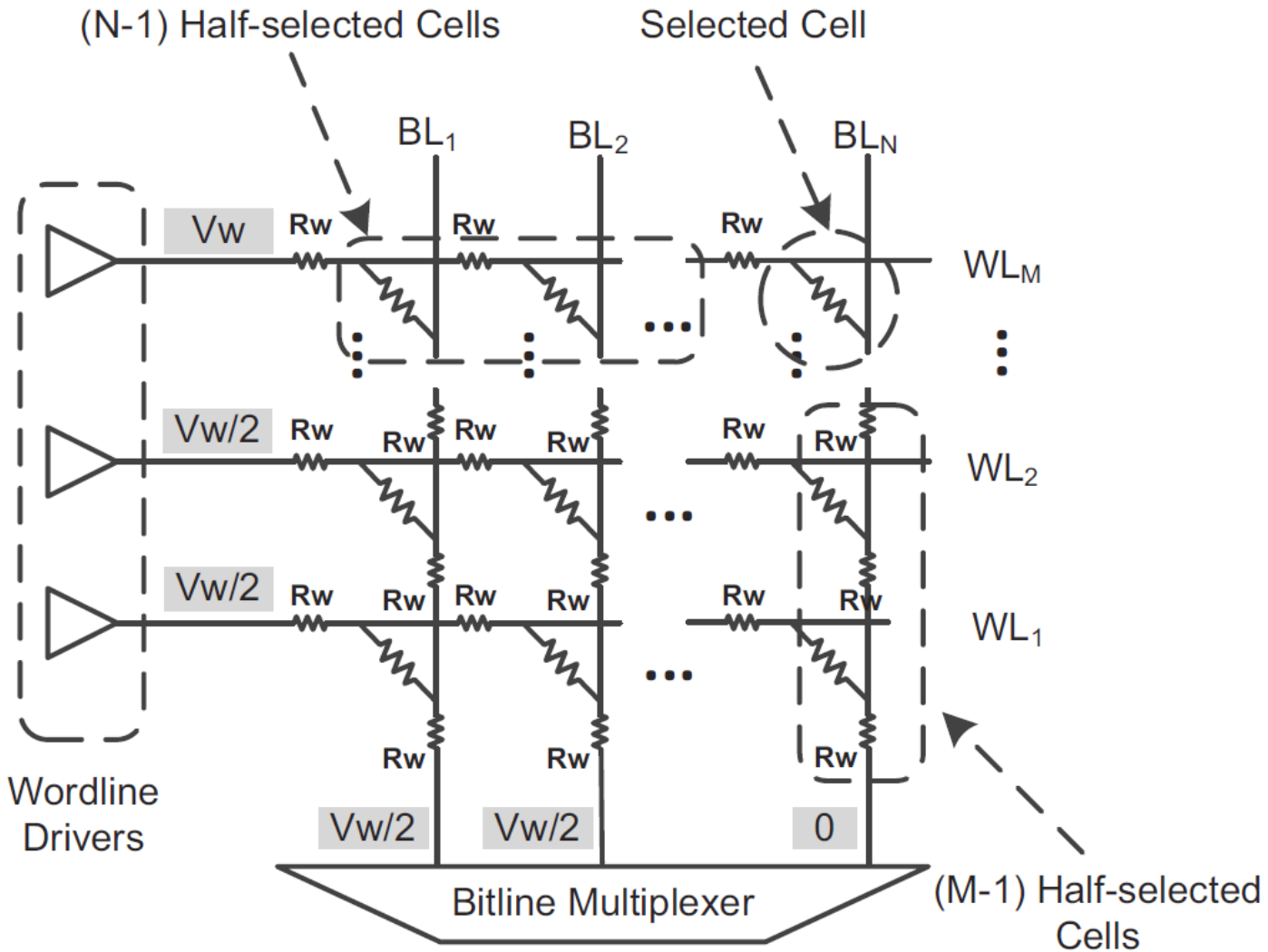

1 Serializer
2 Conveyor
From
SBin
Convert burst
to stream of
records (FSM)
=
<
WE
Pull burst of records from the most full partition buffer (FSM)
3 Merge
To SBout

Resistive RAM

- Most new NVMs are resistive
- The term “ReRAM” typically refers to memristors
- A metal-oxide material sandwiched between two electrodes; the resistance depends on the direction of the last current to flow through the material
- High density is achieved with a 0T1R cell implemented with a “cross-point” structure
- Density can be increased with “3D-stacking”; more metal layers ➔ more cells in a given area

ReRAM Cross-Point Structure

Niu et al., ICCAD'13



Memristor Reads/Writes

- Each cross-point array has limited size so that sneak currents are manageable
- Can either do a 2-phase write or only write 1 cell in each array – the latter is fine because a cache line can be interleaved across several arrays
- HWHB for writes: all unselected wordlines and bitlines are at half-voltage and the selected word/bitline are fully biased
- For reads: only the selected wordline has read voltage; all other wordlines/bitlines are grounded; each bitline current indicates resistance of that cell

Title

- Bullet