The Flash Effect on Big Data

Bruce Moxon

CTO, Systems and Solutions

sTec®

Discussion Outline

- Defining Big Data
- Big Data's HPC Legacy and Hyperscale
- Data Warehousing and Analytics
- NoSQL
- Solid State Storage Key Features
- Big Data acceleration scenarios
  - Hadoop, Cassandra, Splunk
- Future

What is Big Data? How Big is Big?

"Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization."

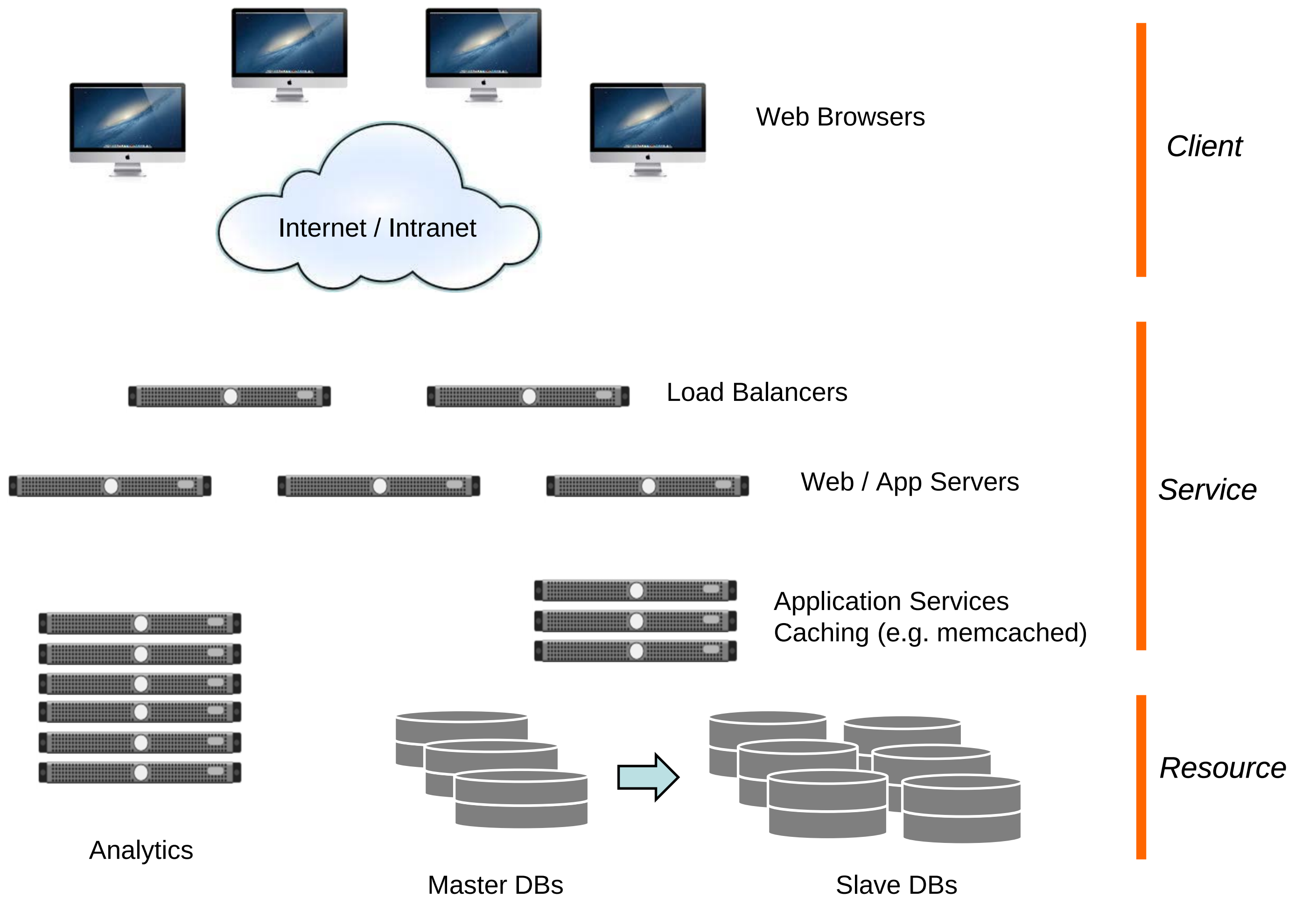
- Gartner (based on Doug Laney's 3V's)
http://en.wikipedia.org/wiki/Big_data

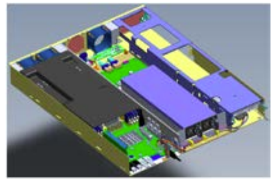
New (evolved) Technologies, Approaches, Tools

- Scale-out computing
- Application-specific storage (SDS)
- In-memory databases
- Columnar databases
- MapReduce and Distributed File Systems
- NoSQL (k-v, graph, document), NewSQL
- Complex Event Processing

A significant (dominant) Open Source model.

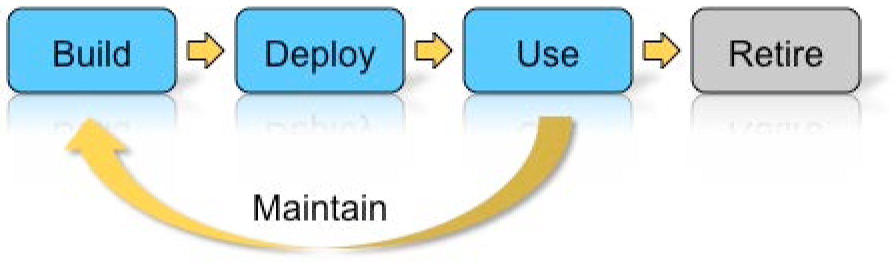
Scale-out, HPC, and Hyperscale Data Centers



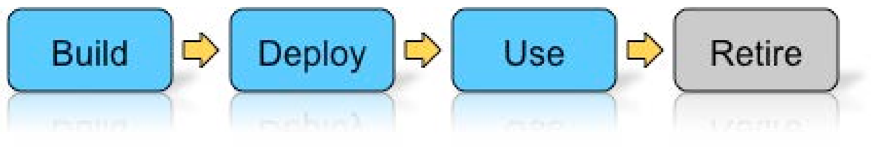


Why Open Compute Matters … for more than Facebook

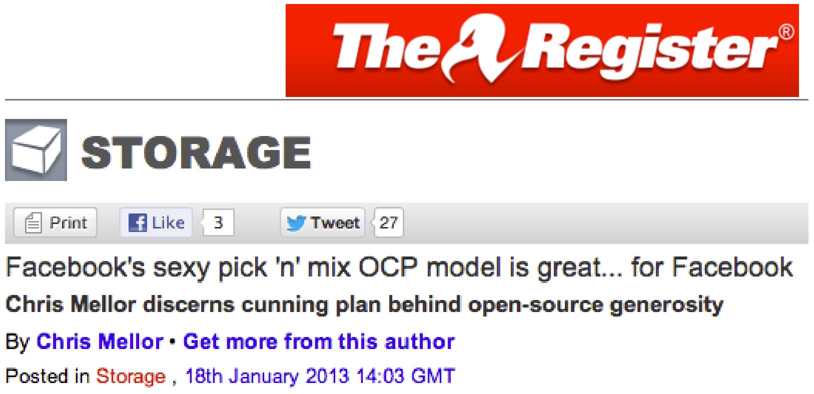
Traditional Infrastructure Lifecycle



Hyperscale Infrastructure Lifecycle



Scale Matters



The Register®

STORAGE

Print | Like 3 | Tweet 27

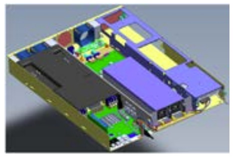
Facebook's sexy pick 'n' mix OCP model is great... for Facebook

Chris Mellor discerns cunning plan behind open-source generosity

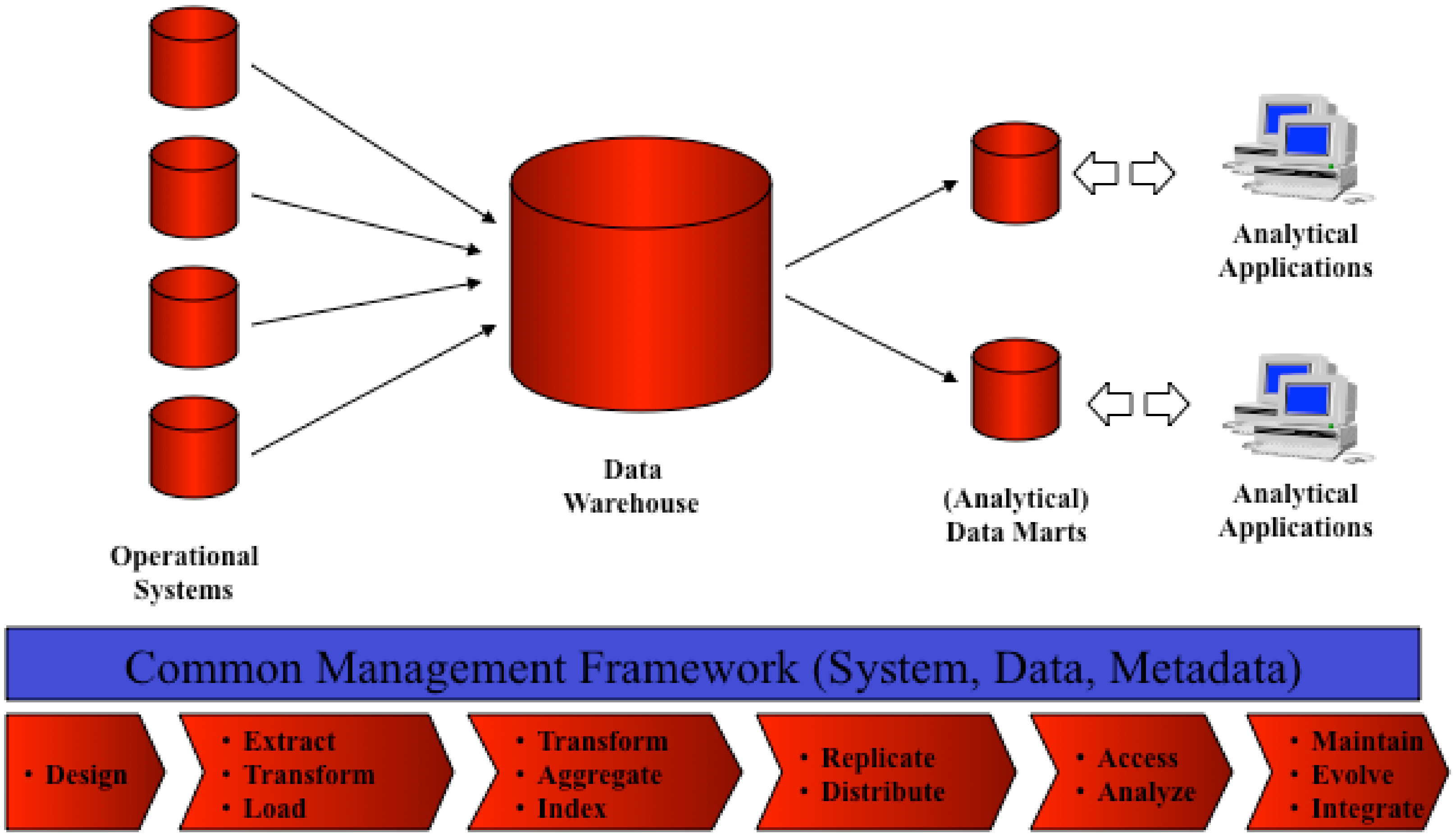
By **Chris Mellor** • **Get more from this author**

Posted in Storage , 18th January 2013 14:03 GMT

OPEN Compute Project



Traditional Data Warehousing *Process (DW/BI)*



Operational vs Analytical Schema

Organization of data should reflect the activities (processes) that use it

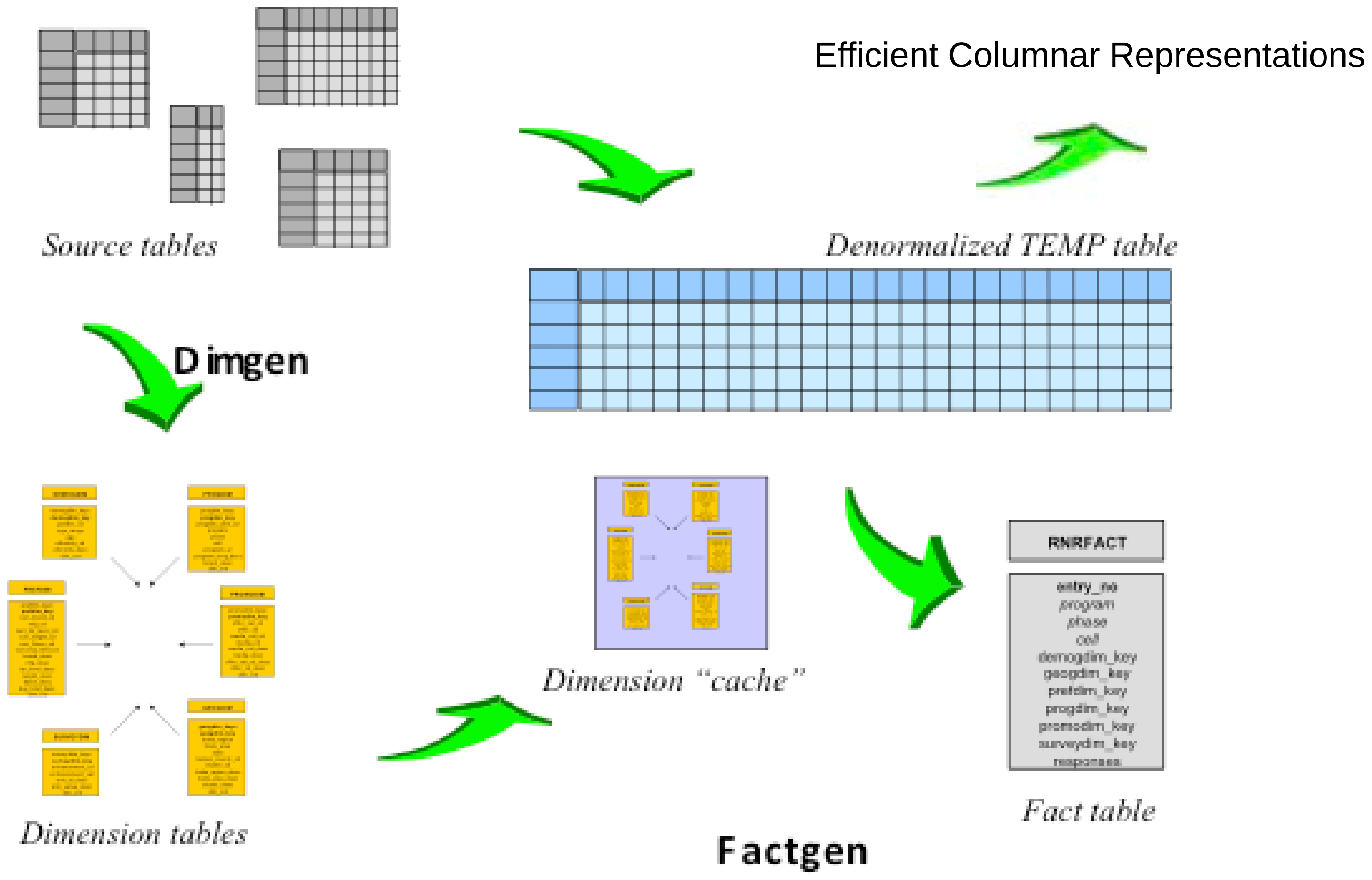
Operational (OLTP)

- **Optimized for tracking and status reporting (e.g. LIMS, sample tracking)**
- **Transactional queries**
  - **Short, UPDATE intensive**
  - **Well defined, indexed**
  - **Few records per query**
  - **High volume**
- **Exploits RI, triggers, stored procedures**
- **No "results"**
- ***Normalized Schema (3NF)***

Analytical (OLAP)

- **Optimized for large-scale analysis and decision support applications**
- **Analytical queries**
  - **Long, SELECT intensive**
  - **Ad hoc, table scans**
  - **Many records (scans), many tables (joins)**
  - **Relatively low volume**
- **scalar functions, aggregation, extended functions**
- **Results used elsewhere (BI reporting, visualization, …)**
- ***De-normalized Schema***

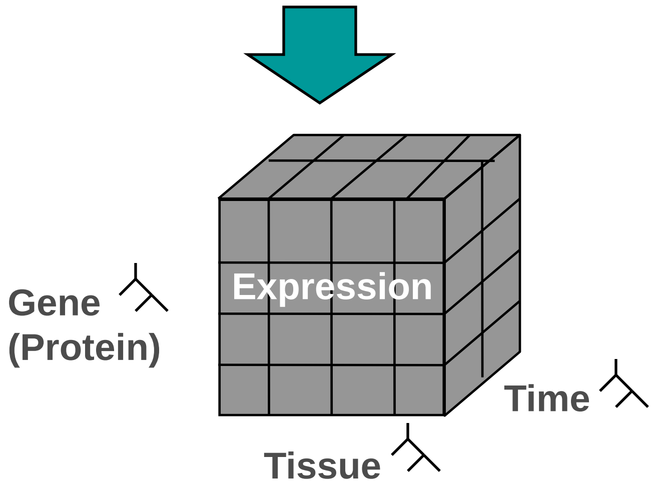
Denormalization, Columnar Stores, Star Schema, and Cubes



OLAP, Multi-dimensional Databases (MDDs), and Cubes

- **An approach to multi-resolution, N-dimensional data analysis (OLAP)**
  - **central fact table(s)**
  - **peripheral dimension tables**
- **Optimized for quick response to "slice and dice" queries, drillup / drilldown, pivot**
  - **pre-aggregations (rollups)**
  - **highly indexed**
  - **special schema (star, snowflake)**
  - **read mostly / batch update model**
  - **browsable dimensions**
  - **filtering of facts by dimension attributes**
- **Efficient data representations**
  - **sparse data, time series**
  - **MDD or RDBMS implementations**

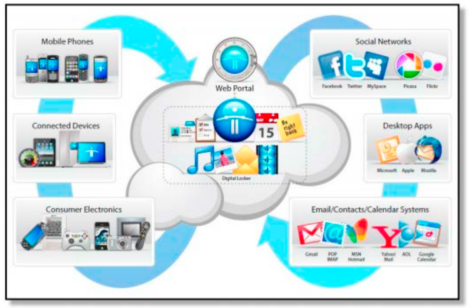
Library	Organ	Donor	Geneid	Pathology	Expr Level
L01833	brain	103391	g121131	normal	0.00214
L01833	brain	103391	g124122	normal	0.00133
L01833	brain	103391	g089913	normal	0.00212
L01011	lung	110122	g113314	cancerous	0.00413



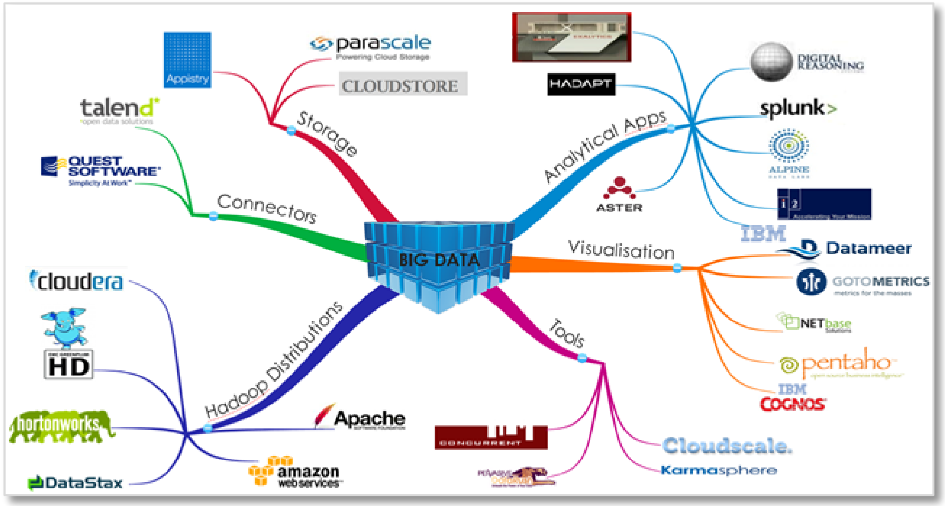
An Evolution of Data Warehousing / Analytics Platforms

Teradata (1983) • First DSS appliance based on MPP architecture	Gen 1 • Scaleout MPP
Netezza (2000); acquired by IBM (2010) • Hardware accelerated “function shipping” (FPGA-based AMPP architecture)	Gen 2 • Custom HW (x)
Greenplum (2003); acquired by EMC (2010) • Shared nothing implementation of Postgres on commodity server MPP architecture • Columnar stores and MapReduce support added	Gen 3 • Commodity servers (scaleout)
Vertica (2005); acquired by HP (2011) • Columnar data stores (scan performance, high compression factors) ParAccel (2005); • Columnar data stores, commodity servers	Gen 3.5 • Columnar stores
Aster Data (2005); acquired by Teradata (2011) • MapReduce-based implementation, SQL-MapReduce NoSQL Data Warehouses • HBase, Cassandra, …	Gen 4 • MapReduce and Derivatives, NoSQL

The New Analytics Infrastructure



The Internet of Things



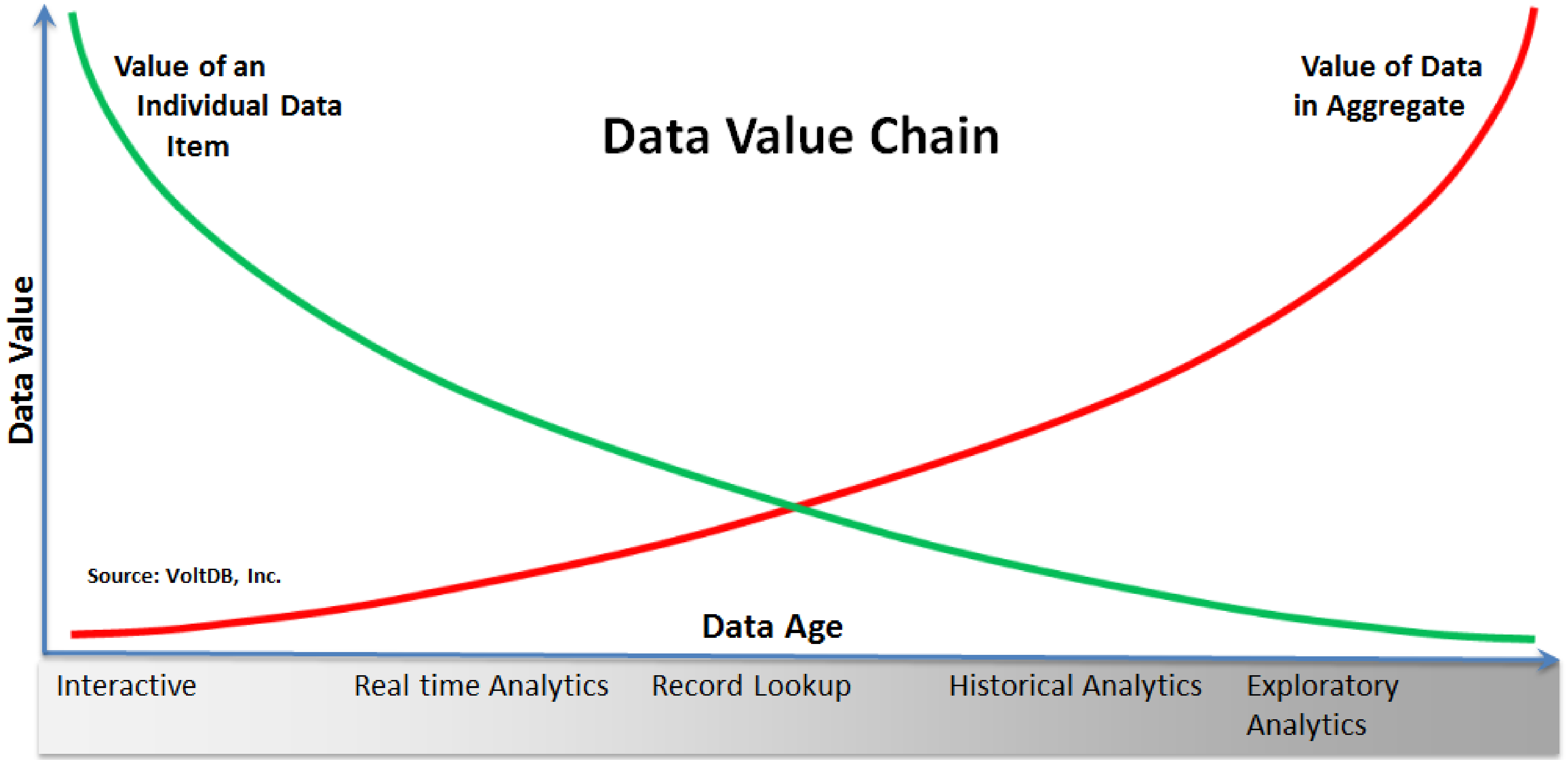
Big Data Ecosystem

Key Implications

- Data Warehouse, Operational Data Store, Data Mart distinction blurring
- Historically batch-oriented processes (ETL, summary/aggregation) are becoming more real-time
    - Indexing, Query (SQL)
    - Background MapReduce

http://www.funambol.com/documents/Funambol_InternetofThings_May10.pdf
http://stevenimmons.org/wp-content/uploads/2012/02/bigdataecosystem.png

Time-value of Data



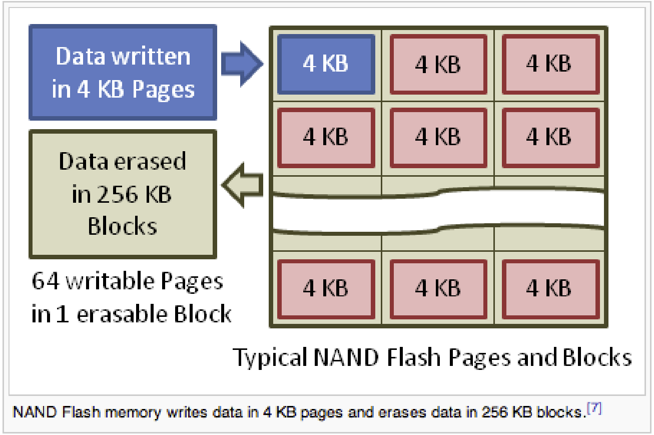
http://blog.voltdb.com/big-data-value-continuum/

NoSQL and NewSQL

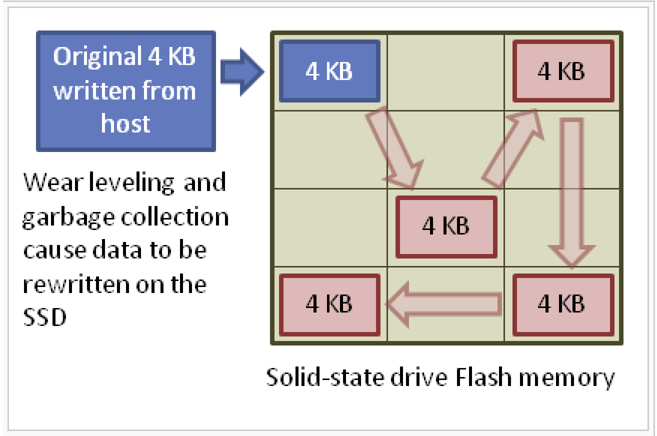
- NoSQL
  - Highly scalable, (generally) looser consistency models than traditional RDBMSs (CAP Theorem; MVCC, ACID, and BASE)
    - Column: HBase, Cassandra, Accumulo
    - Document: MongoDB, Couchbase, MarkLogic
    - Key-value: Dynamo, Riak, Redis, LevelDB, BerkeleyDB, MemcacheDB
    - Graph: Neo4J, Objectivity Infinite Graph
- NewSQL
  - New class of high performance databases supporting the relational model and SQL (transactional, realtime ETL and analytics)
    - NuoDB, VoltDB, Memsql, Pivotal SQLFire

http://en.wikipedia.org/wiki/NoSQL
http://nosql-database.org/
http://www.marklogic.com/blog/acid-base-and-nosql/

Solid State Storage Fundamentals


Data written in 4 KB Pages
Data erased in 256 KB Blocks
64 writable Pages in 1 erasable Block
4 KB
Typical NAND Flash Pages and Blocks

NAND Flash memory writes data in 4 KB pages and erases data in 256 KB blocks.[7]


Original 4 KB written from host
Wear leveling and garbage collection cause data to be rewritten on the SSD
4 KB
Solid-state drive Flash memory

An SSD experiences write amplification as a result of garbage collection and wear leveling, thereby increasing writes on the drive and reducing its life.[1]

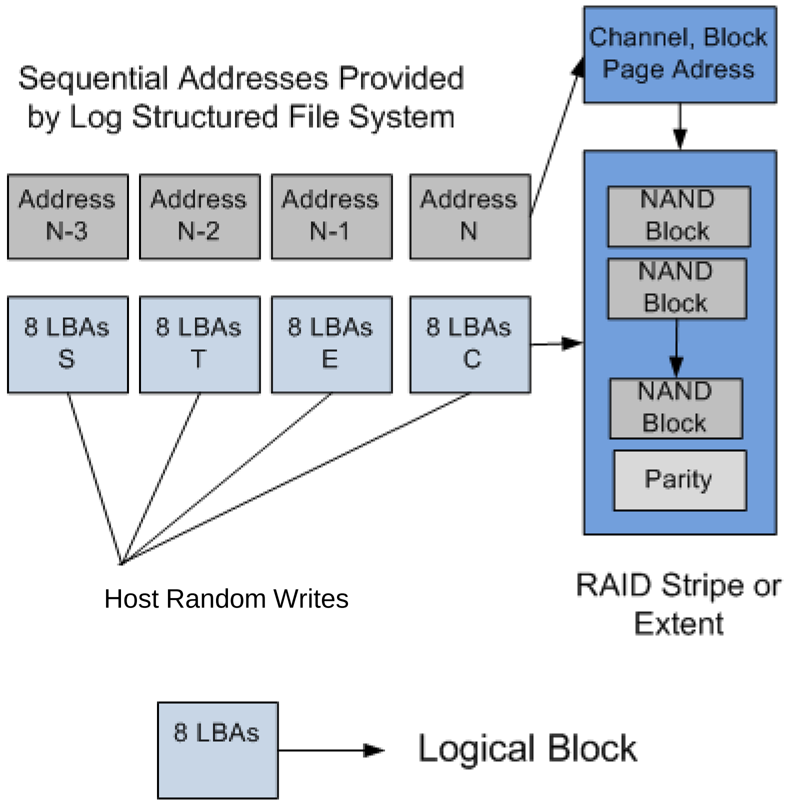
http://en.wikipedia.org/wiki/Write_amplification#Basic_SSD_operation

Everything I Needed to Know I learned at FMS …

- Data is read/written in pages (typically 4-8 KB)
- Data is *erased* in multi-page blocks (e.g., 128)
- Data can only be written (programmed) into a previously erased block (no "overwrite")
- Background garbage collection (GC) copies valid pages to "squeeze out" deleted pages and make room for new data
  - OS/FS integration (TRIM)
- Additional *write amplification* can occur in support of wear leveling and GC
- Flash memory can only be programmed/erased (P/E) a limited number of times (Endurance)
  - *Performance degrades over device lifetime*
- Overprovisioning is usually employed to afford more flexibility in background tasks such as wear leveling and garbage collection. It reduces write amplification, in turn extending the life of the device
  - Background tasks affect performance, especially at "peak" device speeds (latency)
- Implementations vary widely

Log Structured File Systems

- First Described in a paper by
  - Mendel Rosenblum and John K. Ousterhout
    - http://www.cs.berkeley.edu/~brewer/cs262/LFS.pdf
- Basic principles (Solid State implementations)
  - Extra space or over-provisioning is added to the capacity of the drive
  - Writes are always written to new logical blocks (LBs); Updates invalidate old LBs
  - Garbage Collection periodically purges the invalidated LBs by re-writing still valid LBs to fresh Erase Blocks and erasing the cleared Erase Blocks
  - Effectively transforms random writes into sequential writes
    - The percentage of transformation is dependent on the amount of over-provisioning
  - FTL embodies the virtual-to-physical (V2P) mapping



Solid State Performance Characteristics (General)

HDD (SAS)	Sequential	Random
Read	200 MB/s	200 IOPS
Write	200 MB/s	200 IOPS

3-8x ↓ ↓ 100-1000x

SSD / PCIe	Sequential	4K Random
Read	.5 – 1.5 GB/s	60-200K IOPS
Write	.5 – 1.3 GB/s	50-250K IOPS

Rand Response	Read	Write
HDD	8 ms	0.5 ms*
SSD	60 us	20 us

General Performance Characteristics. YMMV, depending on

- Device architecture
- Interface (SATA, SAS, PCIe)
  - 2-5x performance range
- Transfer sizes
- QDs (concurrency)

Cost Differential

- $0.50 - $1.50 / GB SAS HDD
- $2 - $12 / GB SSD/PCIe (< $1)

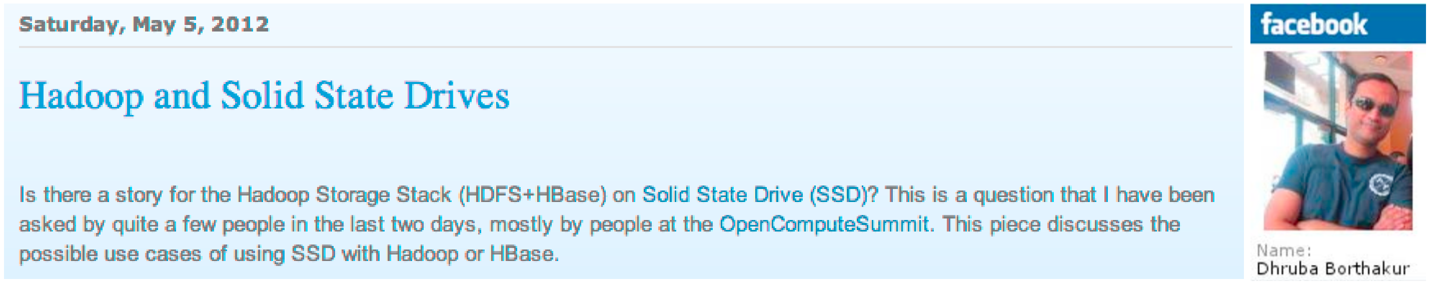
Sweet Spot

- High Random IOPS (esp. Read)
- Low Latency

Key Solid State Storage Characteristics

- Outstanding random IO relative to HDD
  - Random write performance through log-structured filesystem approaches; corresponding amplification / endurance considerations
- Moderate increase in sequential IO relative to HDD
  - $-per-MB/s parity; Capacity premium
- Significant latency advantage (10x or more)
- Much higher concurrency support (64-128+ per device) without significant latency impact

Flash for MapReduce (?)


Saturday, May 5, 2012

Hadoop and Solid State Drives

Is there a story for the Hadoop Storage Stack (HDFS+HBase) on Solid State Drive (SSD)? This is a question that I have been asked by quite a few people in the last two days, mostly by people at the OpenComputeSummit. This piece discusses the possible use cases of using SSD with Hadoop or HBase.

facebook

Name:
Dhruba Borthakur

From http://hadoopblog.blogspot.com/2012/05/hadoop-and-solid-state-drives.html

Investigations and Conclusions

- Basic map-reduce is large block sequential and doesn't benefit much (not economically) from SSDs
- HDFS+HBase
  - Tested HDFS with DRAM-based data (~ infinitely fast disk) to determine basic HDFS capability: 92K IOPS and 95% CPU
  - Tested HBase with DRAM-based table (~ infintely fast disk) to determine basic HBase capability: 35K ops/s at 45% CPU; heavy lock contention and context switching

"My conclusion is that database and storage technologies would need to be developed from scratch if we want to utilize the full potential of Solid State Devices"

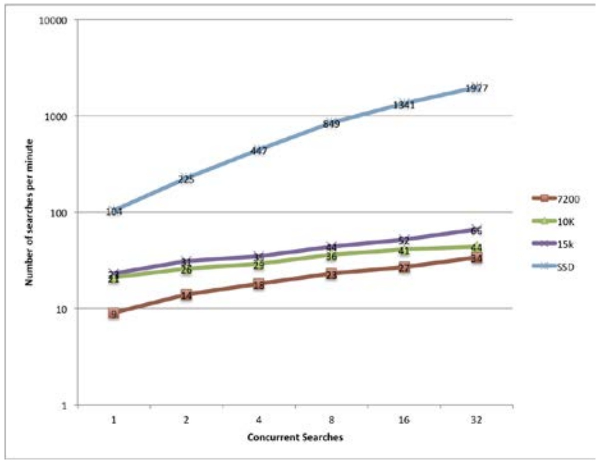
Ongoing Work: BucketCache victim cache (HBASE-7404); Tiered HFile Storage (HBASE-6572); Heterogeneous Storage (HBASE-2832)

Flash for Splunk

Blogs: Tips & Tricks

Quantifying the Benefits of Splunk with SSDs

From http://blogs.splunk.com/2012/05/10/quantifying-the-benefits-of-splunk-with-ssds/

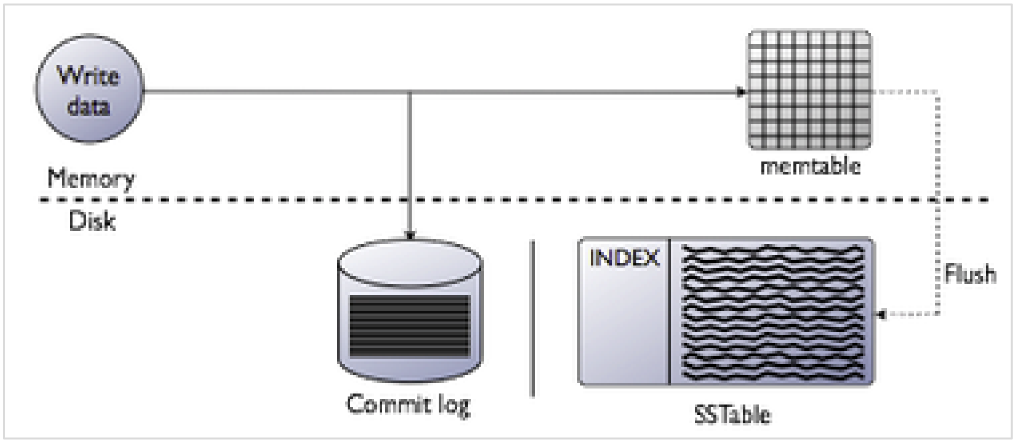


Guidance

- Ingest and indexing are (sequential)-write-intensive activities; these will benefit only modestly from use of SSDs
- Searches:
  - Dense searches – typically used for aggregation – will mostly use large, sequential reads (“scan” operations or “range queries”)
  - Sparse searches – including cross-series queries – depend on random reads (both indexes and data) and will make good use of SSD’s (200x demonstrated for pure “point” queries)
    - Bloom Filters may also benefit (Splunk 4.3)

Flash for Cassandra

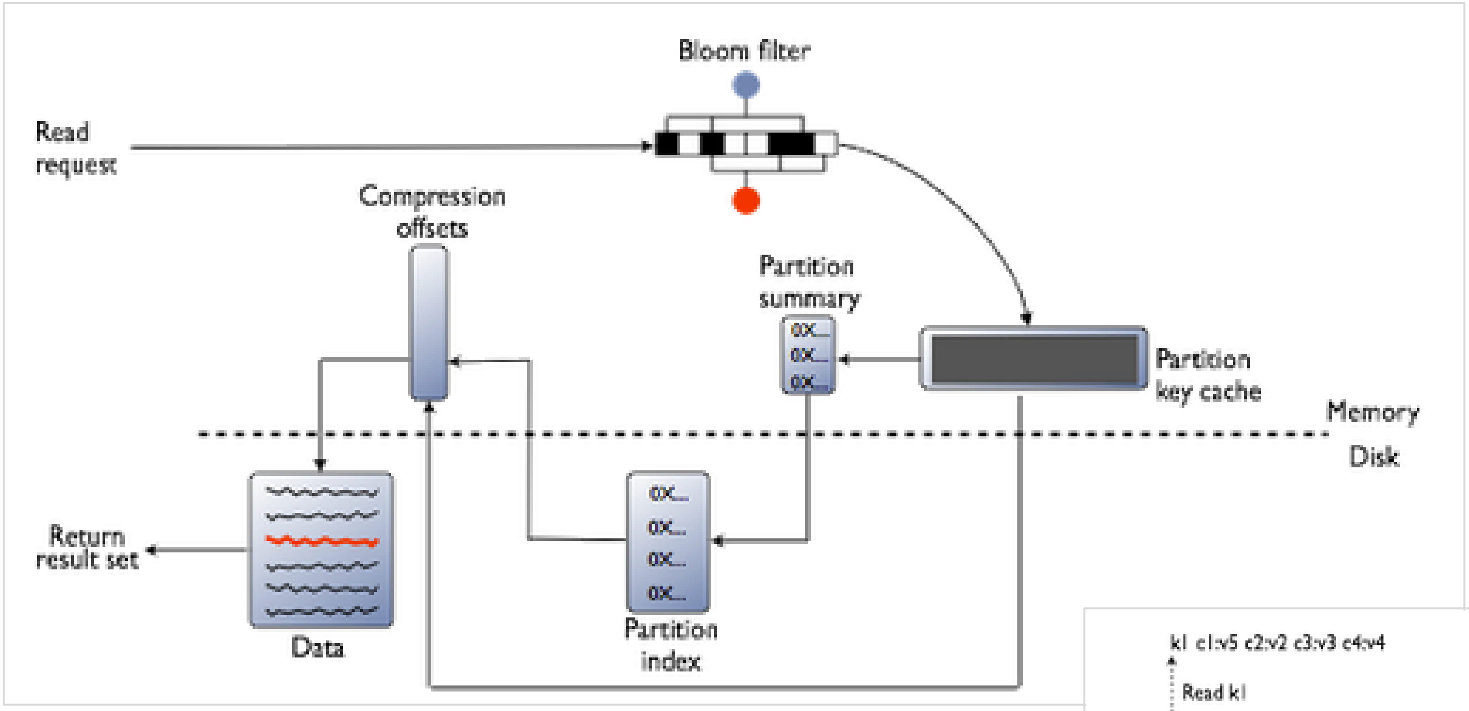
Cassandra Writes (1.2)

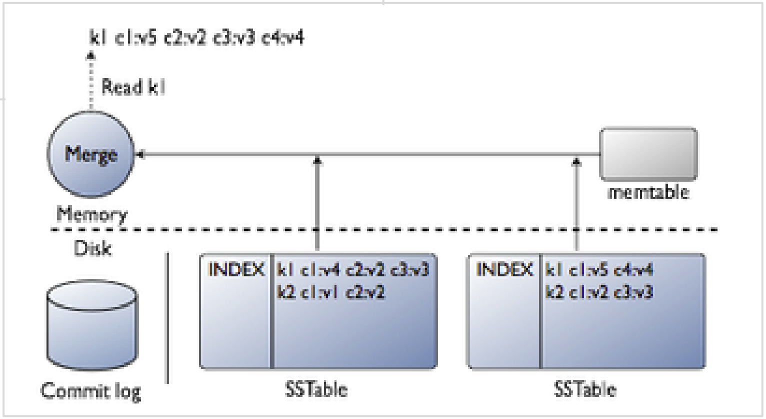


Flash for Cassandra

Cassandra Reads (1.2)

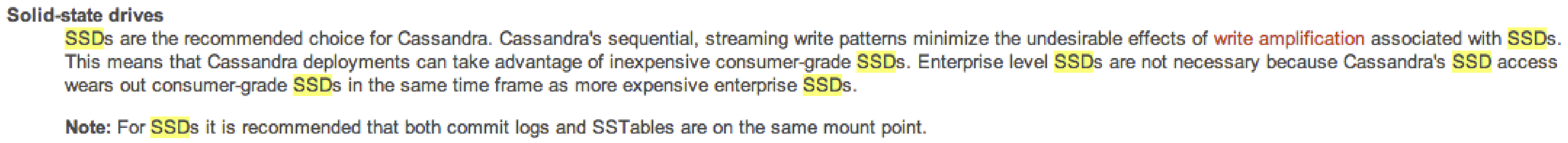
1.2: Compression offsets and Bloom filter off-heap (capacity scaling)





Flash for Cassandra

http://datastax.com/documentation/cassandra/1.2/webhelp/index.html?pagename=docs&version=1.2&file=dml#cassandra/architecture/architecturePlanningHardware_c.html



Solid-state drives

SSDs are the recommended choice for Cassandra. Cassandra's sequential, streaming write patterns minimize the undesirable effects of write amplification associated with SSDs. This means that Cassandra deployments can take advantage of inexpensive consumer-grade SSDs. Enterprise level SSDs are not necessary because Cassandra's SSD access wears out consumer-grade SSDs in the same time frame as more expensive enterprise SSDs.

Note: For SSDs it is recommended that both commit logs and SSTables are on the same mount point.

There are a number of cassandra.yaml settings specifically for Cassandra tuning for SSDs, mostly taking into consideration the increased concurrency and performance afforded by SSDs. The .yaml file includes suggested values. YMMV ☺

- concurrent_reads and concurrent_writes
- multithreaded_compaction and compaction_throughput_mb_per_sec (disable throttling)
- commitlog_sync_period_in_ms

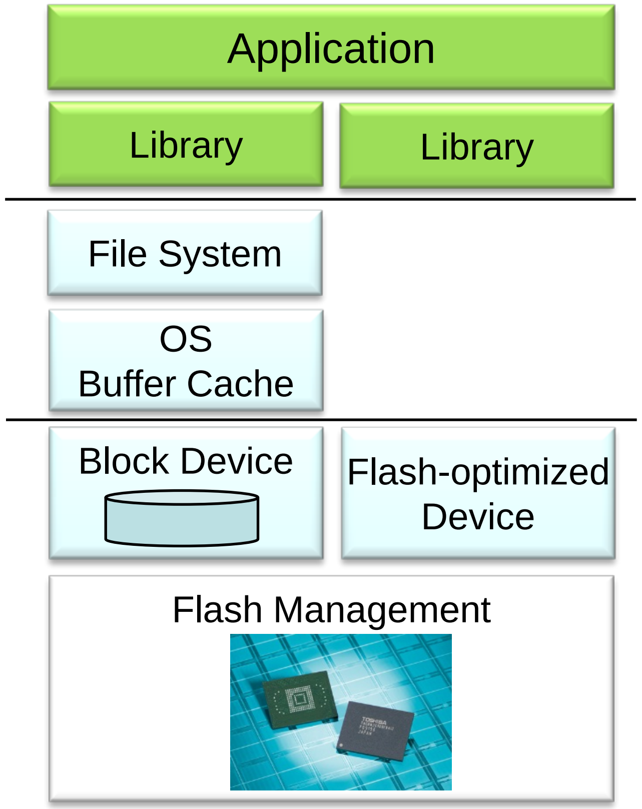
There may be other not-so-obvious settings depending on workload (ingest rate, amount of indexing, query types and concurrency, …). E.g. compaction frequency/type, which can generate more SSTables (which SSD's should handle well), more or less aggressive Bloom Filters, etc. With HDD's, the tendency is to defer flushes (so there may be less data to write); with SSD's, more frequent flushes and more SSTables files may yield better performance (higher concurrencies, which SSD's accommodate).

Flash and Big Data: Future

- There are some things on the horizon that can fundamentally change the evolving architectures
  - High thread count processors (Intel Xeon Phi)
  - PCIe and NVMe
  - Non-spinning disk abstractions on Flash
    - Fusion-io ioMemory SDK
    - sTec FOS/kvs
    - NVM Programming TWG
  - Storage Class Memory (MRAM, PCM, resistive RAM, …)

"Flash 3.0"

- Flash technology presents new opportunities for *application optimizations* in the data path
  - Random access media
  - Near-memory latency at near-disk cost
- These will be optimally delivered through new device abstractions better-matched to device characteristics
  - Memory abstractions
  - Object abstractions
    - Flash Object Store™ (FOS/KVS)
- Natural extension to *Storage Class Memory*



Summary

- A range of Big Data stores and tools are emerging based on scale-out commodity server implementations; many are open source
- These tools provide new approaches – and extensions – to traditional Data Warehouse/Business Intelligence problems
- Many of these stores/tools optimize for sequential writes for ingest and indexing (and logging); this approach is “Flash friendly” (limiting write amplification), but doesn’t drastically improve ingest performance
- Focused queries (point queries) can benefit greatly from the increased random read performance and higher concurrency support of Flash-based devices (both index and data reads)
- Next generation NVMe devices promise significant sequential IO improvement to boost sequential read/write performance (ingest and “scan” queries)
- Architectural enhancements (e.g. victim caches) and new IO architectures are needed to fully leverage Flash
- Storage Class Memories will present the next opportunity – with a new set of architectural opportunities and challenges