AEROSPIKE

Optimizing High Performance Databases for Flash Storage

HPTS, Sept 28, 2015

Brian Bulkowski | CTO & Founder, Aerospike

How is Aerospike used ?

OPERATIONAL KEY VALUE
Session, authentication, account status, cookies, deviceID, IP address, location, segments, trades, debits, billing, prices...

REAL-TIME DECISIONS
Best sellers, top scores, trending tweets

LOAD BALANCER
Simple, stateless

APP SERVERS
Fast, stateless

node.js
NGINX
jetty://

HIGH PERFORMANCE NoSQL

AEROSPIKE
cassandra
mongoDB

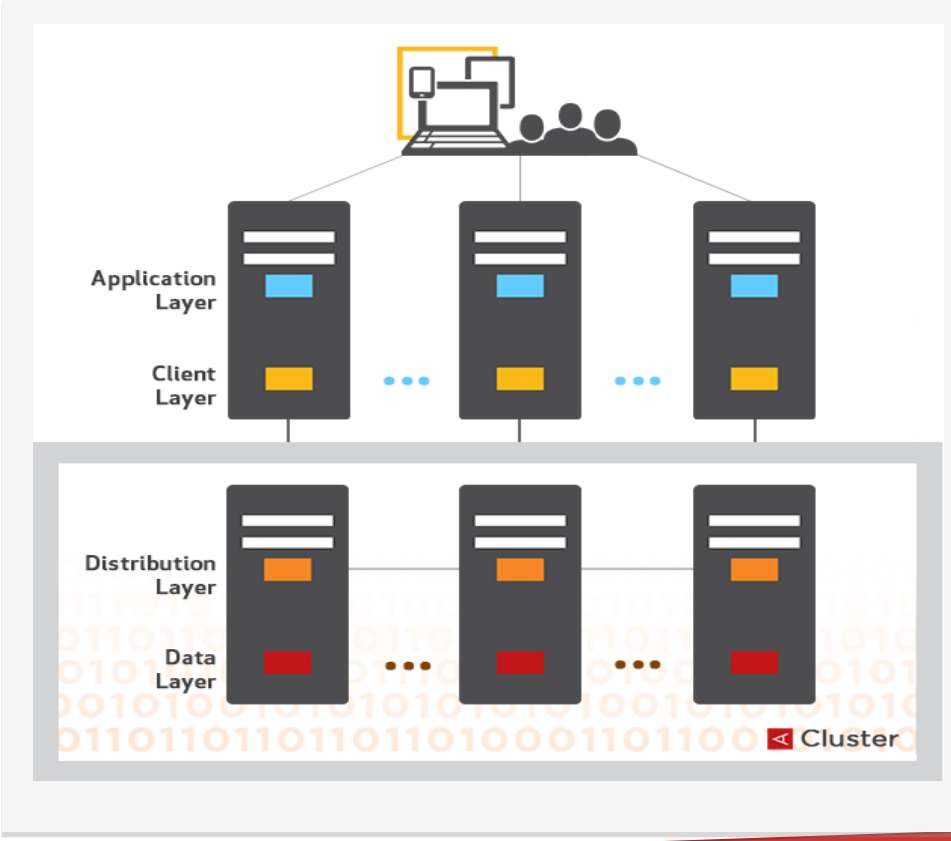
RESEARCH WAREHOUSE
Long-term cold storage

hadoop
APACHE HBASE
HDFS BASED

SHARED-NOTHING SYSTEM: 100% DATA AVAILABILITY

Very Large Data Bases 2011

- Every node in a cluster is **identical**, handles both transactions and long running tasks
- Clients give rich semantics in multiple languages and connect over the network
- Data is replicated **synchronously** with within the local cluster
- Data is replicated **asynchronously** across data centers
- Primary key hash RIPE MD160 (20 byte) for **extreme collision resistance** (DHT), **Red black tree** for records within hash bucket
- Scatter-gather **B+** secondary index



A Flash optimized database

- 2008 - Hadoop, Aster, Greenplum, Vertica, Netezza …
  - Analytics meets clustering with HDD
- Internet-scale databases were in terrible shape
  - Sharded MySQL + memcache → ewwww
  - Oracle seen as unacceptable
  - Big guys write their own
- Flash gives insane IOPs – great for small objects (< 4K)
  Power isn't getting cheaper
- Target scale:
  Billions of objects,
  Terabytes of storage,
  Millions of requests per second,
  "small" clusters (1 to 100)

2010 – the year of mainstream Flash

- Intel X-25M released
  - Device itself was --- flawed --- but available
- First Aerospike deployment – Samsung SS805
- OS issues observed in Linux
  - Disk scheduler – good
  - File systems – bad
  - Memmap – terrible
  - Page cache – terrible
  - SATA NCQ – twitchy
  - TCP – fairly fast

Immutable NAND attributes

- Parallelism required
- “Garbage collector” in device
- Controller weaknesses / differences
- Write amplification causes drive failures
- 70 ms read similar to a TCP/IP network round trip

Log-based storage layout

- Log-based layout is EXCEPTIONALLY good
- Copy-on-write for reliability
- Large objects create less garbage collection
  - And thus less write amplification
- DRAM buffers later become hardware supported writes
- …. but create continual defragmentation (compaction)

Indexes in DRAM

- Don’t need to persist
  - Apply k-safety rules
  - Gain reliability with clustering
- Need parallelism of DRAM
  - DRAM memory bandwidth still very important
- “Persistent” DRAM through shared memory
- Support background tasks (eviction, migration)

Improving controllers

- “Sunlight is the best disinfectant”
- http://github.com/aerospike/act
- **DEVIATION** of **LATENCY** measured
- Internet behavioral workload
  - 1.5 KB reads
  - 128 KB writes
  - 50 / 50 read write
  - Include defragmentation workload (large reads + writes)
  - Random, no de-dup, no cache locality
- Easy to run 24 to 48 hours
- Increase workload until 95% latency > 2 ms
  (roughly)

Results

- 2011
  - Intel 320 good
  - Samsung SS805 very good
  - OCZ was “spotty”
  - Fusion OK but price problems
- 2013 – “wide sata”
  - Intel S3700 series
  - Samsung 840 series
- 2013 – Few great PCIe
  - Micron P320h – SLC – 150x
  - Micron P420h – MLC – 24x
- 2015 – NVMe, finally
  - Intel P3700 series
  - Samsung PM1720 series
  - Amazon i2 – 6x per drive

Device	Speed	Reads (1.5KB)	Writes (128KB)	Reads (128KB)	Price
Intel 320	“3x”	6 K (9 MB/s)	18 MB/s	18 MB/s	$3 / G
S3700	“6x”	12 K (18 MB/s)	32 MB/s	32 MB/s	$2 / G
IoDrive2	“3x”	6 K (9 MB/s)	18 MB/s	18 MB/s	$8 / G
P320h	“150x”	300 K (450 MB/s)	900 MB/s	900 MB/s	$8 / G
Crucial M500	“0.5x”	1 K (1.5 MB/s)	3 MB/s	3 MB/s	$0.5 / G
P3700 PM1720	“75x” (**)	150 K (225 MB/s)	450 MB/s	450 MB/s	$2 / G

A word about NAND reliability

- 100+ customers
- 1000+ drives for largest customers
- 100's of drives for most customers

- Drives fail occasionally – a few per month
  Need online "hot standby" (replica)
  Remote hands can hot-swap a new device
  (front-panel NVMe almost here)

- Need ability to "forklift upgrade" without downtime

- Correctness at power-cycle is far better now

Beyond the block interface

- Defragmentation at 2 layers is wasteful
- OpenNVM “kv” interface
  - Aerospike + FusionIO collaberation
  - Too complex
- FTL bypass
  - Do wear leveling in the DB
  - More complex than you would think
  - In discussions with vendors

Future of NAND

- NAND has *at least* 4x density coming soon
  - Sandisk pushing hard this year
- NAND speed increases possible (2x / 4x)
- Fast transaction logging
- In-place writes becoming practical
- Control of garbage collection
- In-device compute

Intel XPoint

- New Intel / Micron memory technology
- “between NAND and DRAM” in latency
  (NAND – 70 ms, DRAM 10 ns)
- Fine grained writes (no write block problem)
- Power only while reading or writing
- Great for indexes
  Option for transactional writes

AEROSPIKE

Questions?

brian@aerospike.com
@bbulkow

@aerospikedb