Serving Billions of Queries In Millisecond Latency

Biju Nair
HBaseConAsia 2018
August 17, 2018

TechAtBloomberg.com

Agenda

- HBase principles
- Modeling
- Implementation
- Monitoring and Tuning

Bloomberg by the numbers

- Founded in **1981**
- **325,000** subscribers in **170 countries**
- Over **19,000 employees** in 192 locations
- **More News reporters** than The New York Times + Washington Post + Chicago Tribune
- **Over 5,000 Engineers**

Bloomberg Tech

- Over 5,000 software engineers
- 100+ technologists and data scientists devoted to machine learning
- One of the largest private networks in the world
- 100B+ tick messages per day, with a peak of more than 10 million messages/second
- >1.5M news stories ingested / published each day (that's 500 news stories ingested/second)
- News content from 125K+ sources
- More than a billion messages (emails and IB chats) processed each day

TechAtBloomberg.com

Bloomberg
Engineering

Bloomberg in a nutshell

Data Storage and Retrieval

- Files
- VSAM
- Network
- Hierarchical
- Relational
- MPP

TechAtBloomberg.com

Bloomberg
Engineering

RDBMS Application Lifecycle

- Use Case
- Entities and Relations
- Logical data model
- Physical data model
- Implementation and tuning

TechAtBloomberg.com

Bloomberg

Engineering

HBase Principles

- Ordered Key Value Store
- Distributed

Key Value

...

Key-9999	Value
Key-9998	Value
Key-9997	Value
Key-9996	Value
Key-9995	Value
Key-9994	Value

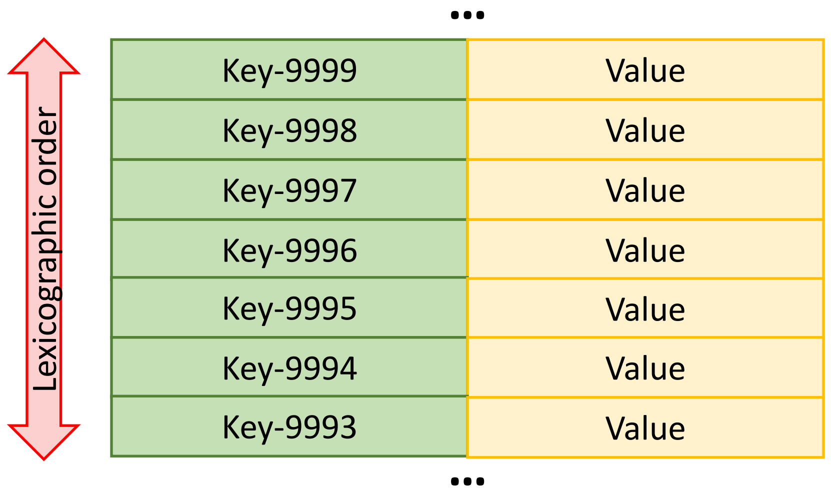
...

TechAtBloomberg.com

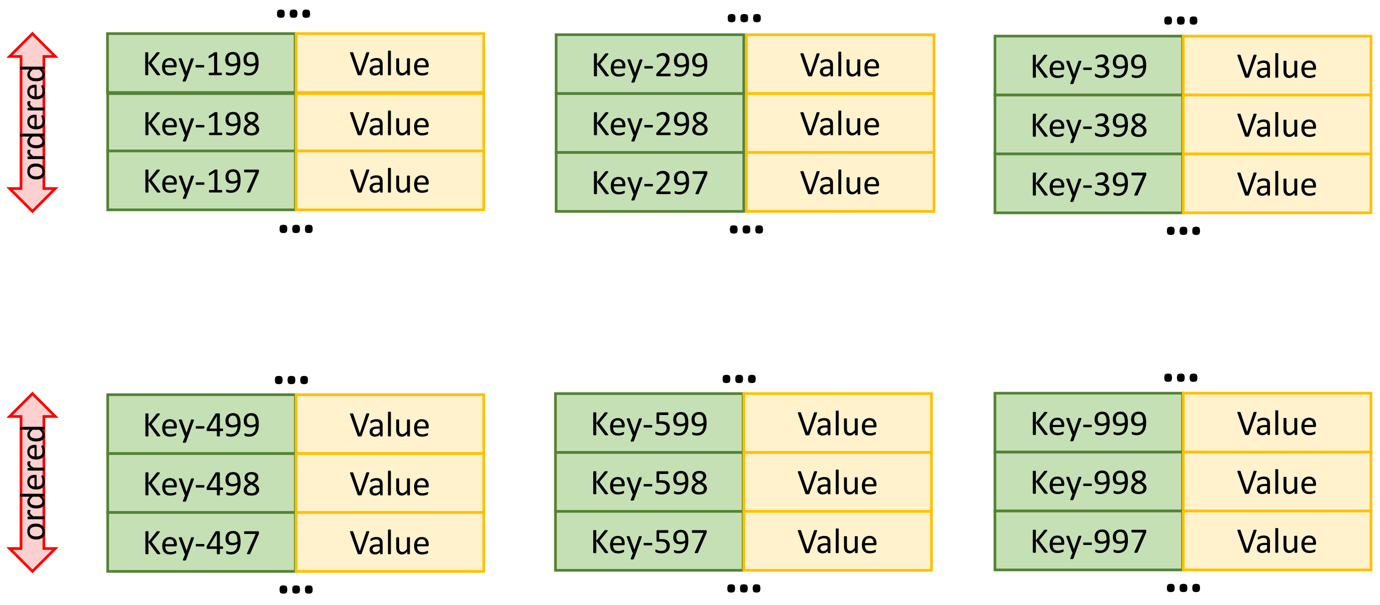
Bloomberg

Engineering

Ordered Key Value



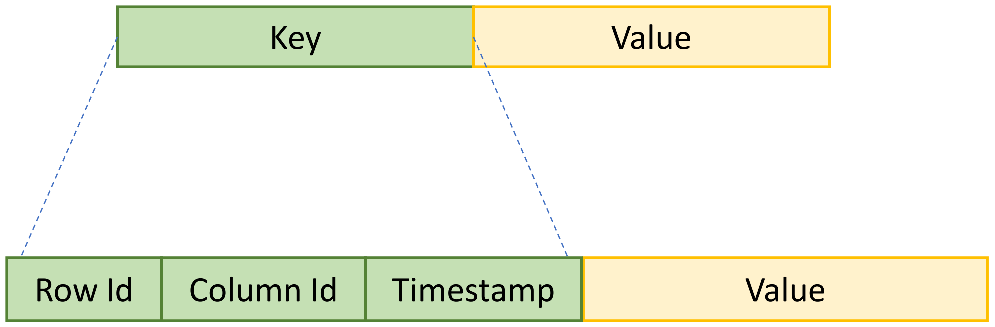
Distributed Order Key Value



Abstraction

- Table row view
- Versioning
- ACIDity

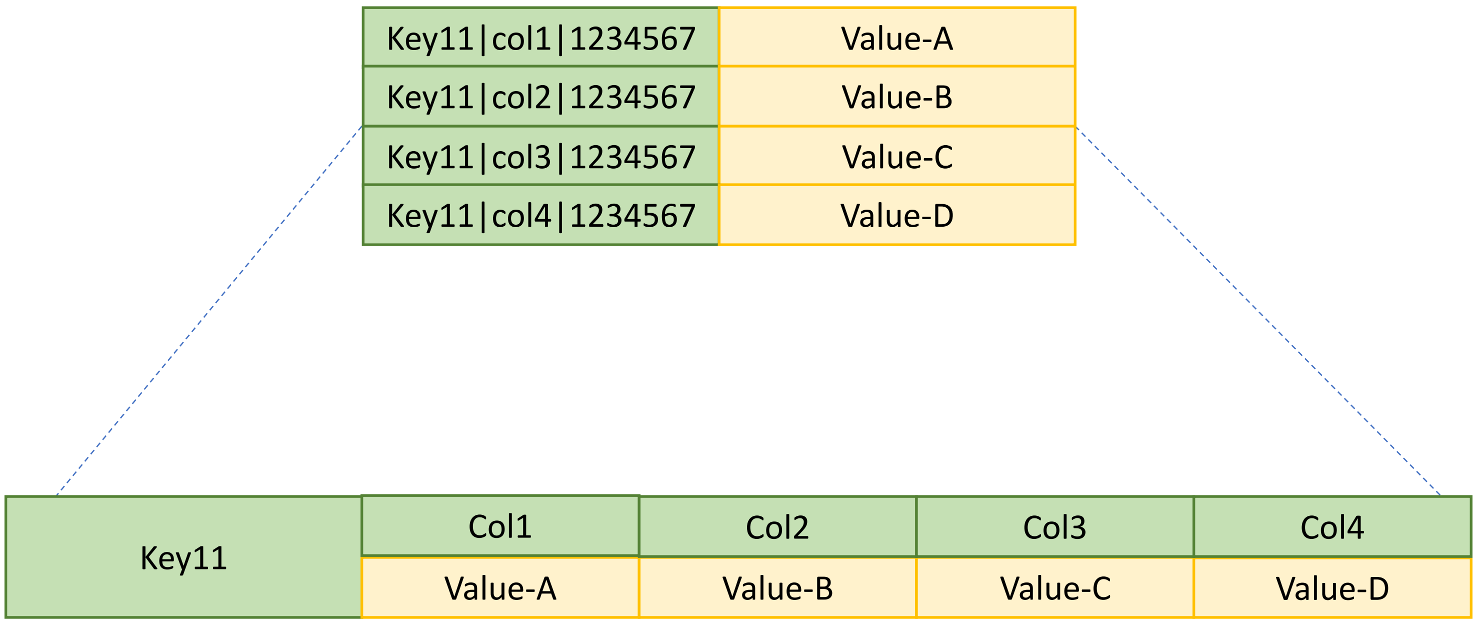
Table Row View



TechAtBloomberg.com

Bloomberg
Engineering

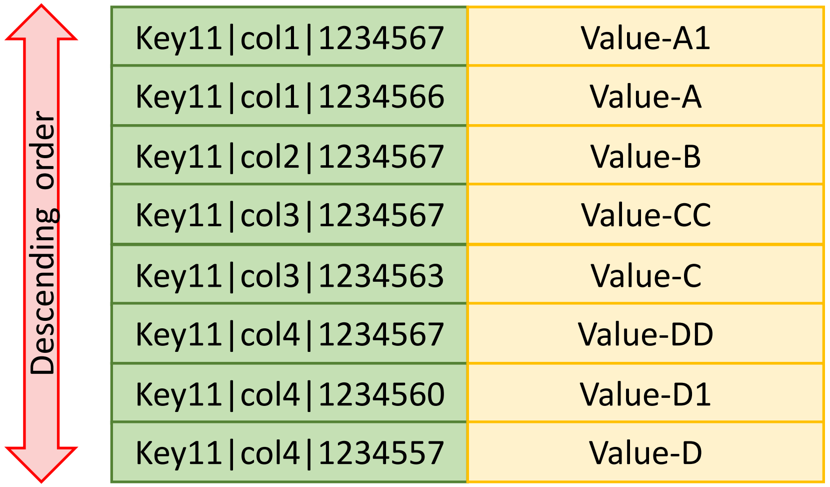
Table Row View



Key11|col1|1234567	Value-A
Key11|col2|1234567	Value-B
Key11|col3|1234567	Value-C
Key11|col4|1234567	Value-D

	Col1	Col2	Col3	Col4
Key11	Value-A	Value-B	Value-C	Value-D

Versioning



Key11|col1|1234567	Value-A1
Key11|col1|1234566	Value-A
Key11|col2|1234567	Value-B
Key11|col3|1234567	Value-CC
Key11|col3|1234563	Value-C
Key11|col4|1234567	Value-DD
Key11|col4|1234560	Value-D1
Key11|col4|1234557	Value-D

TechAtBloomberg.com

Bloomberg Engineering

ACIDity

- **A**tomic at row level
- **C**onsistent to a point in time before the request
- **I**solation through MVCC (reads) and row locks (mutations)
- **D**urability is guaranteed for all successful mutations

TechAtBloomberg.com

Bloomberg

Engineering

Data Modeling

- Fitness for key value store
  - — Can’t build relations
  - — No secondary indexes
  - — De-normalization

- Understand queries to design key
  - — Data Skew
  - — Query Skew

Data Skew

Hot

Key-e	Value
Key-e	Value
Key-e	Value
Key-e	Value
Key-e	Value
Key-e	Value
Key-e	Value
Key-e	Value
Key-e	Value
Key-e	Value
Key-e	Value
Key-e	Value
Key-e	Value

Key-a	Value
Key-a	Value
Key-a	Value
Key-a	Value

Key-h	Value
Key-h	Value
Key-h	Value

Key-f	Value
Key-f	Value

Key-x	Value
Key-x	Value

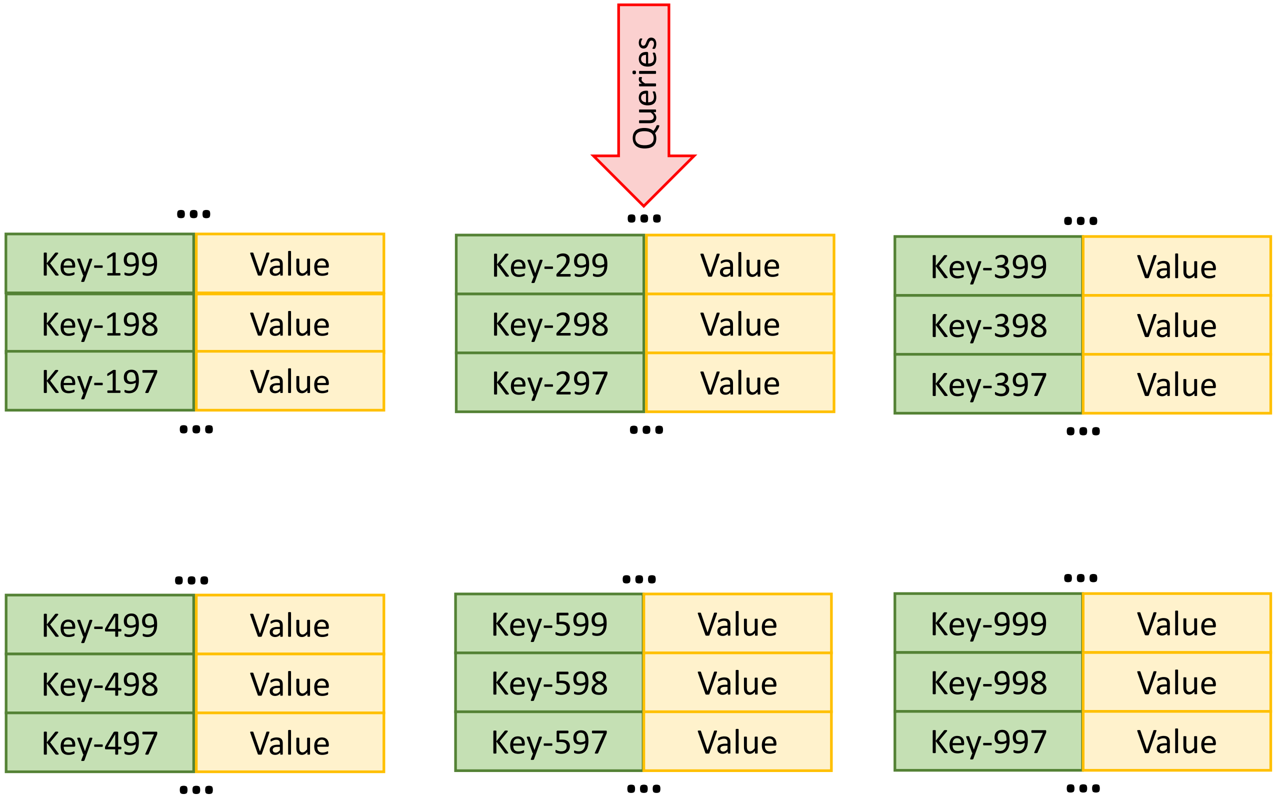
Key-b	Value
Key-b	Value
Key-b	Value

Key-z	Value
Key-z	Value

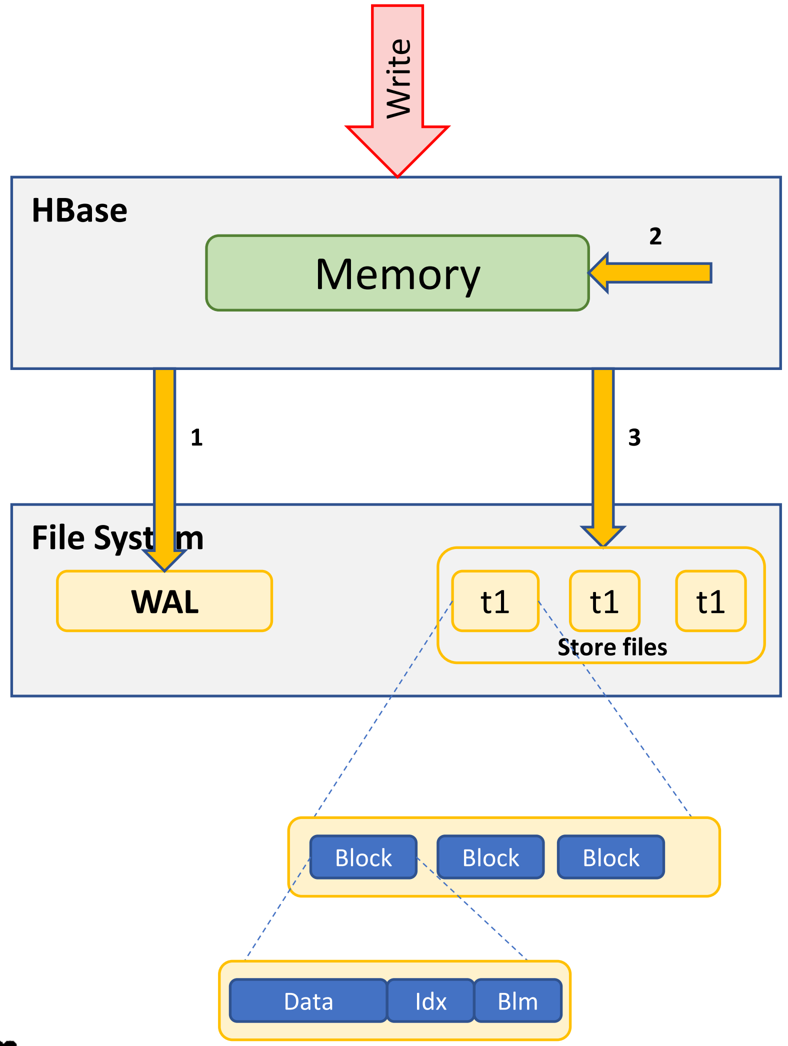
Key-y	Value
Key-y	Value

Key-d	Value
Key-d	Value
Key-d	Value

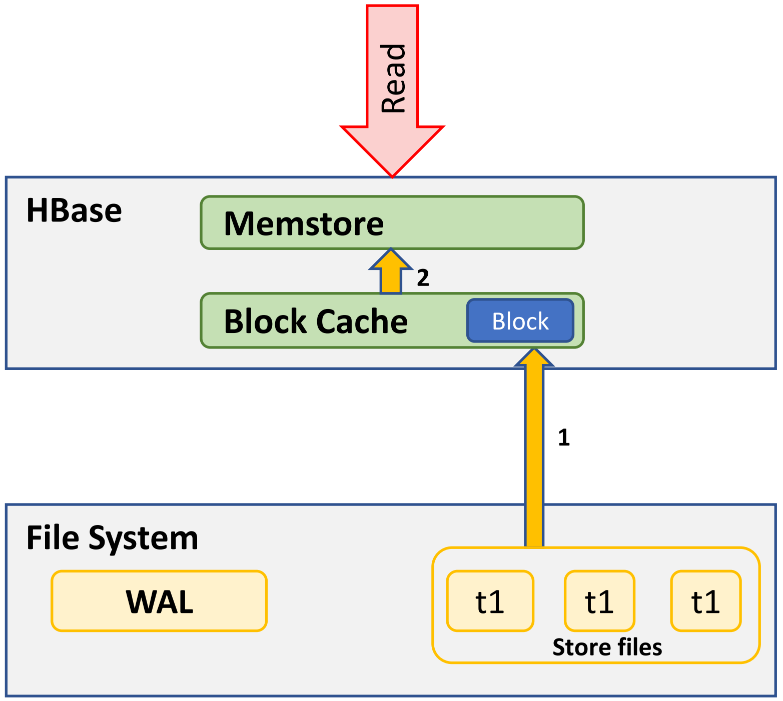
Query Skew


Queries
...
Key-199 Value
Key-198 Value
Key-197 Value
Key-299 Value
Key-298 Value
Key-297 Value
Key-399 Value
Key-398 Value
Key-397 Value
Key-499 Value
Key-498 Value
Key-497 Value
Key-599 Value
Key-598 Value
Key-597 Value
Key-999 Value
Key-998 Value
Key-997 Value

Data Write


Write
HBase
Memory
2
1
3
File System
WAL
t1
t1
t1
Store files
Block
Block
Block
Data
Idx
Blm

Data Read


Read
HBase
Memstore
2
Block Cache
Block
1
File System
WAL
t1
t1
t1
Store files

Cache

- Pack more data into cache
  - Block size
  - Column Family
- Large cache

Block Size vs Read Latency

Get Performance (ms) – 64 K Block

BAvg	16.731	16.728	16.761	16.763	16.418	16.371	16.37	16.431	16.152	16.14	16.169	16.158	16.308	16.29	16.325	16.307	16.34	16.381	16.391	16.352
BMedian	14	14	14	14	13	13	13	13	15	15	15	15	13	13	13	13	13	13	13	13
B95%	41	41	41	41	41	41	41	41	43	43	43	43	40	40	40	40	41	41	41	41
B99%	55	55	55	55	54	54	54	54	55	55	55	55	54	54	54	54	54	54	55	54
B99.9%	71	71	71	71	70	70	70	70	67	67	67	67	71	70	70	71	71	71	71	70
BMax	545	1062	559	567	1075	1027	561	567	564	541	558	1062	1062	561	1075	1072	1067	563	1035	1032

Get Performance (ms) – 16 K Block

Avg	3.002	5.362	5.361	5.357	6.419	6.369	6.405	6.383	6.188	6.196	6.182	6.174	6.246	6.264	6.268	6.253	5.194	5.207	5.219	3.031
Median	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
95%	10	15	15	15	18	18	18	18	18	18	17	17	18	18	18	18	15	15	15	10
99%	15	26	26	26	30	30	30	30	28	28	28	28	29	29	29	29	25	24	25	15
99.90%	26	41	41	41	45	45	45	45	43	43	43	43	44	44	44	44	41	41	41	26
Max	2261	127	185	102	90	106	92	102	93	106	119	114	89	140	132	82	81	150	93	1910

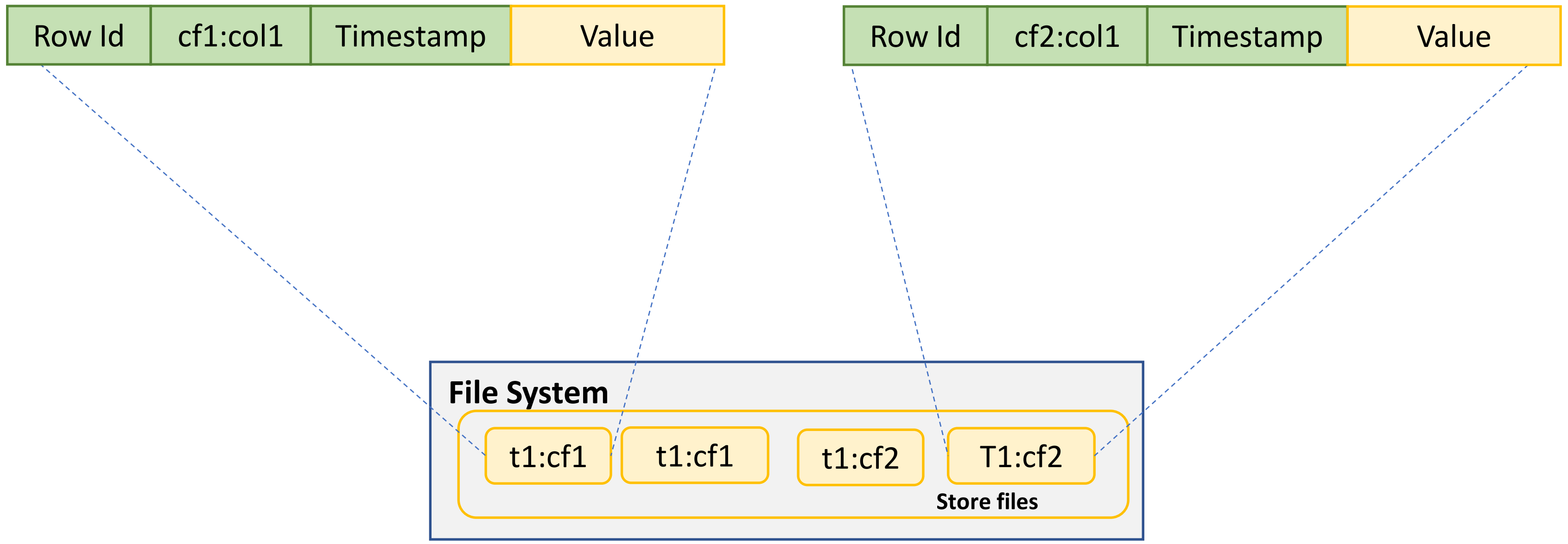
Note: Smaller block size increases the overhead of increased index blocks

Block Size Vs Index Size

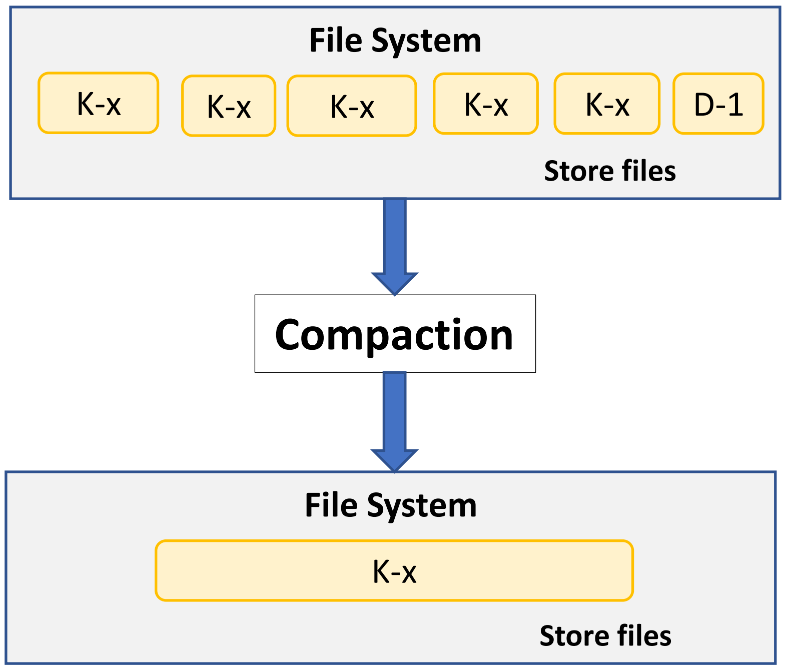
16 K Blocks	
Idx Sz K	Bloom K
266346	2368
247895	2240
225561	2096
253633	2368
224862	2016
225685	2096

8 K Blocks	
Idx Sz K	Bloom K
472058	2432
574239	2944
331899	1792
471362	2304
517272	2560
469543	2432

Column Family



Compaction



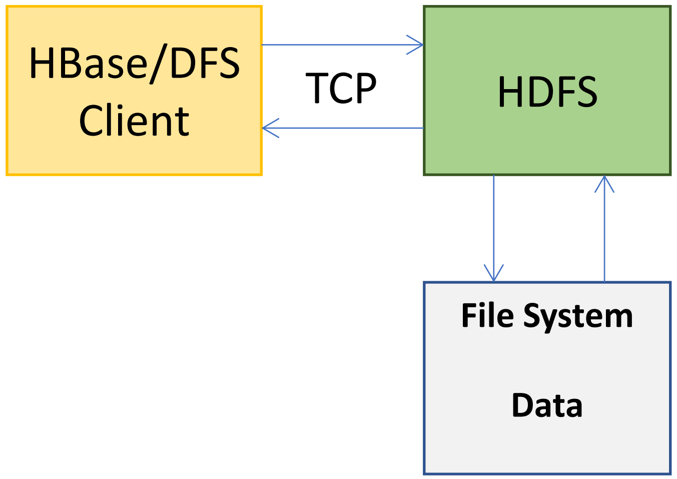
TechAtBloomberg.com

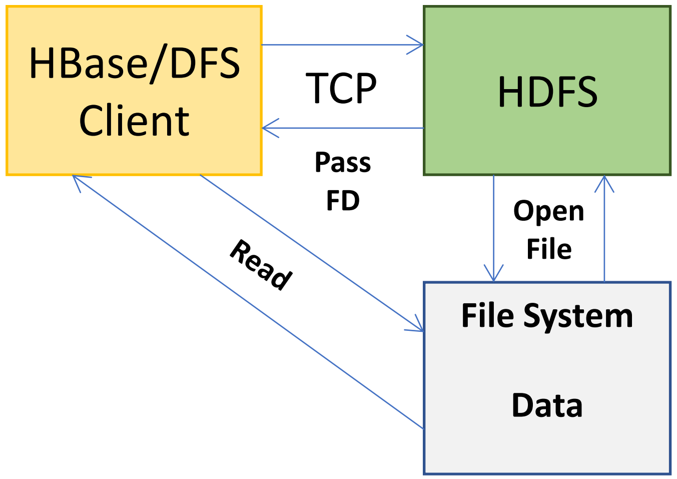
Bloomberg Engineering

Compaction

- Part of regular HBase operations
- Minor Compaction
- Major Compaction
- Utilizes server and HBase resources
- Major compaction can be scheduled

Short Circuit Read

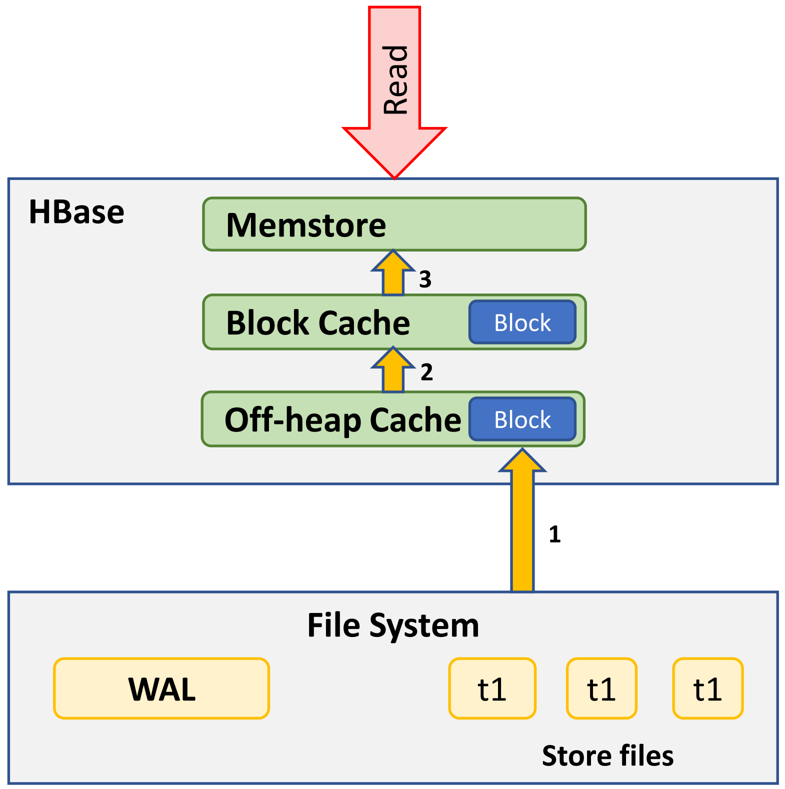




TechAtBloomberg.com

Bloomberg
Engineering

Garbage Collection



Large Cache

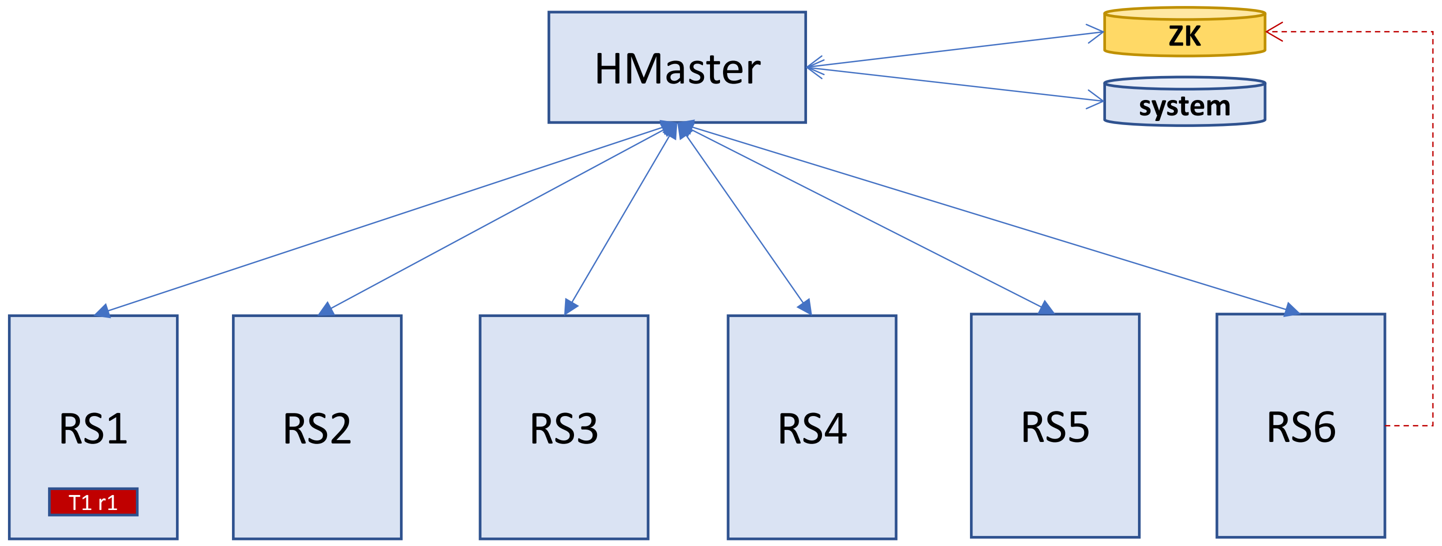
61 GB of Cache					
Avg	2.693	2.814	2.836	2.842	2.812
Median	1	1	1	1	1
95%	8	8	8	8	8
99%	14	14	14	14	15
99.90%	20	20	20	20	20
99.99%	32	31	32	32	33
100.00%	313	319	315	376	341
Max latency	1049	1046	1048	1044	1235

93 GB of Cache					
Avg	3.872	3.995	3.936	4.007	4.052
Median	1	1	1	1	1
95%	14	14	14	15	15
99%	20	20	20	20	20
99.90%	27	27	27	28	28
99.99%	36	36	36	37	37
100.00%	208	310	332	207	232
Max latency	1360	1906	1736	1359	1363

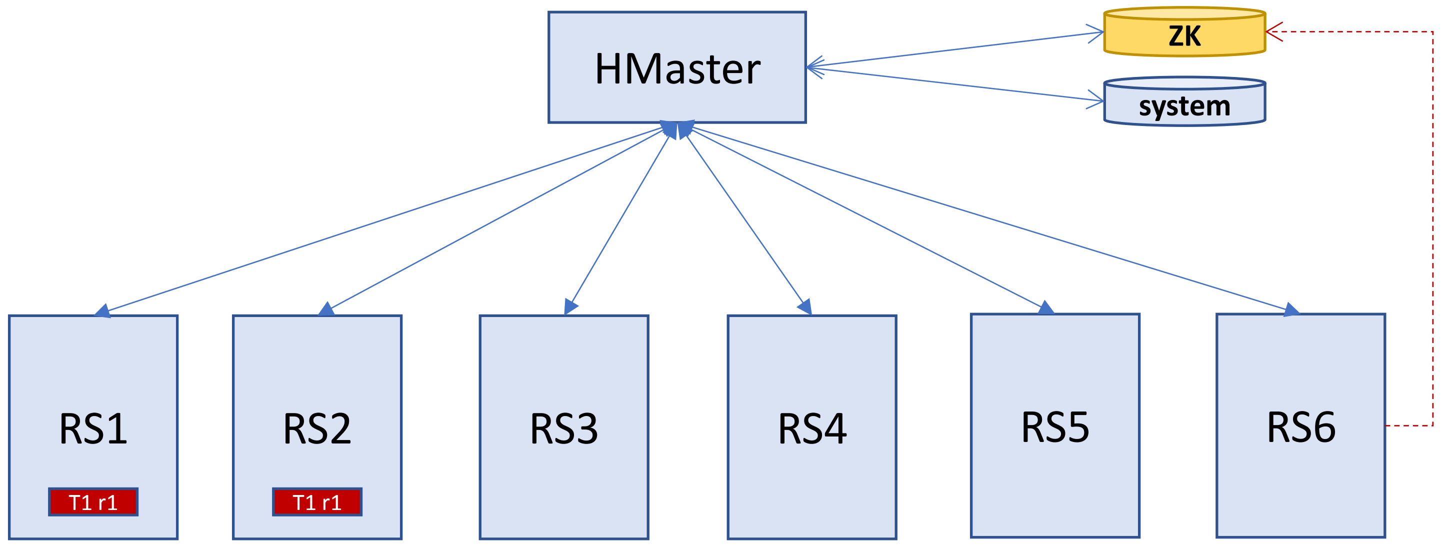
Garbage Collection

- Fine tune Garbage Collector
- For e.g., some CMS GC options to look at
  - ExplicitGCInvokesConcurrent
  - CMSInitiatingOccupancyFraction
  - UseCMSInitiatingOccupancyOnly
  - ParallelGCThreads
  - UseParNewGC
- Log GC info which will help with tuning
  - PrintGCDetails
  - Loggc
  - PrintTenuringDistribution
  - …

Region Replication



Region Replication



TechAtBloomberg.com

Bloomberg

Engineering

Region Replication

- Requires changes to cluster configuration
  - hbase.region.replica.replication.enabled
  - hbase.regionserver.storefile.refresh.period (not the complete list)

- Need to specify region replication in table definition
  - create 't1', 'f1', {REGION_REPLICATION => 2}

- Client need to specify when to read secondary
  - get1.setConsistency(Consistency.TIMELINE);
  - hbase.client.primaryCallTimeout.get
  - hbase.client.primaryCallTimeout.multiget

Region Replication

PrimaryCall Timeout Vs Stale Calls				
Time (ms)	readers	totalQuery	totalStale	%age
3,000	512	1,520,207	0	0.00%
3,000	512	1,520,207	0	0.00%
3,000	512	1,520,207	0	0.00%
1,000	512	1,520,207	0	0.00%
1,000	512	1,520,207	0	0.00%
1,000	512	1,520,207	0	0.00%
100	512	1,520,207	5,101	0.34%
100	512	1,520,207	1,476	0.10%
100	512	1,520,207	74	0.00%
50	512	1,520,207	6,173	0.41%
50	512	1,520,207	4,785	0.31%
50	512	1,520,207	5,263	0.35%
10	512	1,520,207	22,518	1.48%
10	512	1,520,207	16,818	1.11%
10	512	1,520,207	19,050	1.25%

TechAtBloomberg.com

Bloomberg

Engineering

Application Code

- Code on server
  - Co-processor
  - Filters
- Connection reuse
- Batching
- Bulk load instead of Put/BatchMutate
- Scanner caching

Monitoring

- Cache hit ratio
- Data locality
- GC pause
- Compactions
- Call queue
- Read latencies

TechAtBloomberg.com

Bloomberg Engineering

Thank You

Reference: http://hbase.apache.org
Connect with Hadoop Team: hadoop@bloomberg.net

Bloomberg Engineering

TechAtBloomberg.com