Serving Billions of Queries in Millisecond Latency

Bloomberg

Engineering

HBaseConAsia 2018
August 17, 2018

Biju Nair
bnair10@bloomberg.net

TechAtBloomberg.com

Agenda

- Need for low latency
- HBase principles
- Modeling
- Implementation
- Monitoring and Tuning

TechAtBloomberg.com

Bloomberg

Engineering

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

TechAtBloomberg.com

Bloomberg

Engineering

Bloomberg Tech

- More than 5,000 software engineers (and growing)
- 100+ engineers 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
- 2M 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

TechAtBloomberg.com

Bloomberg

Engineering

Data Storage and Retrieval

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

Using RDBMS

- 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

TechAtBloomberg.com

Bloomberg

Engineering

Key Value

...

Key-9999	Value-a
Key-9998	Value-b
Key-9997	Value-c
Key-9996	Value-d
Key-9995	Value-e
Key-9994	Value-f

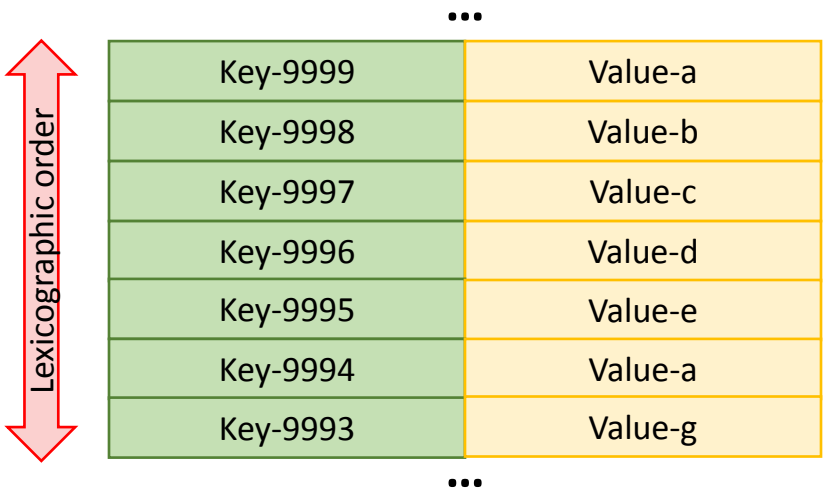
...

TechAtBloomberg.com

Bloomberg

Engineering

Ordered Key Value

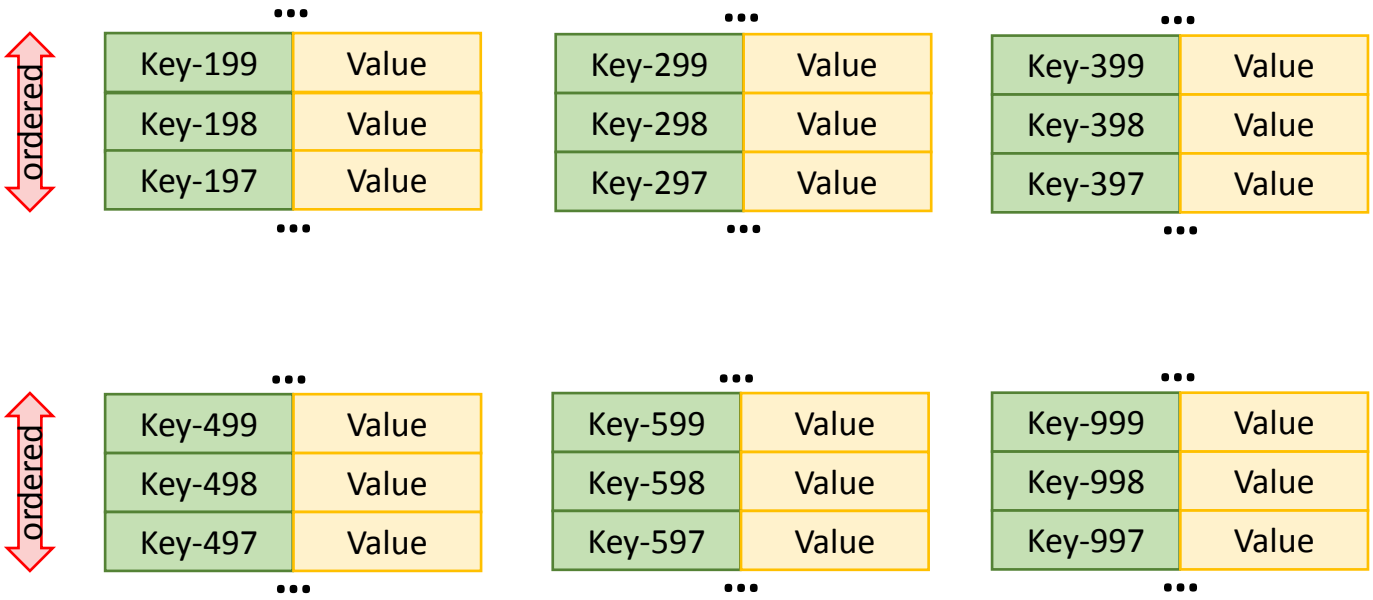


TechAtBloomberg.com

Bloomberg

Engineering

Distributed Order Key Value



TechAtBloomberg.com

Bloomberg

Engineering

Abstraction

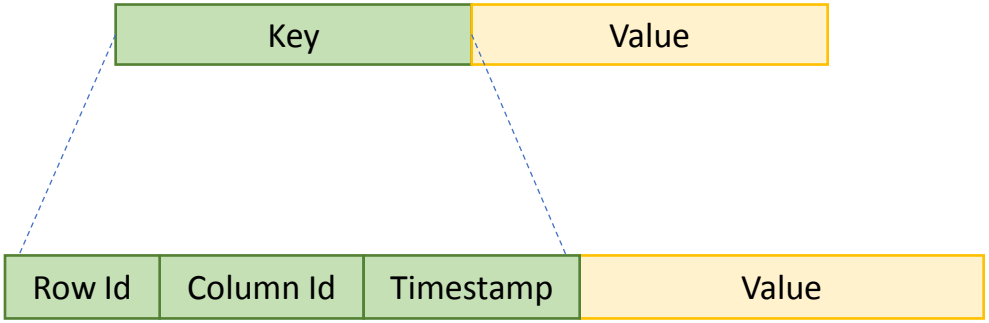
- Table row view
- Versioning
- ACIDity

TechAtBloomberg.com

Bloomberg

Engineering

Table Row View

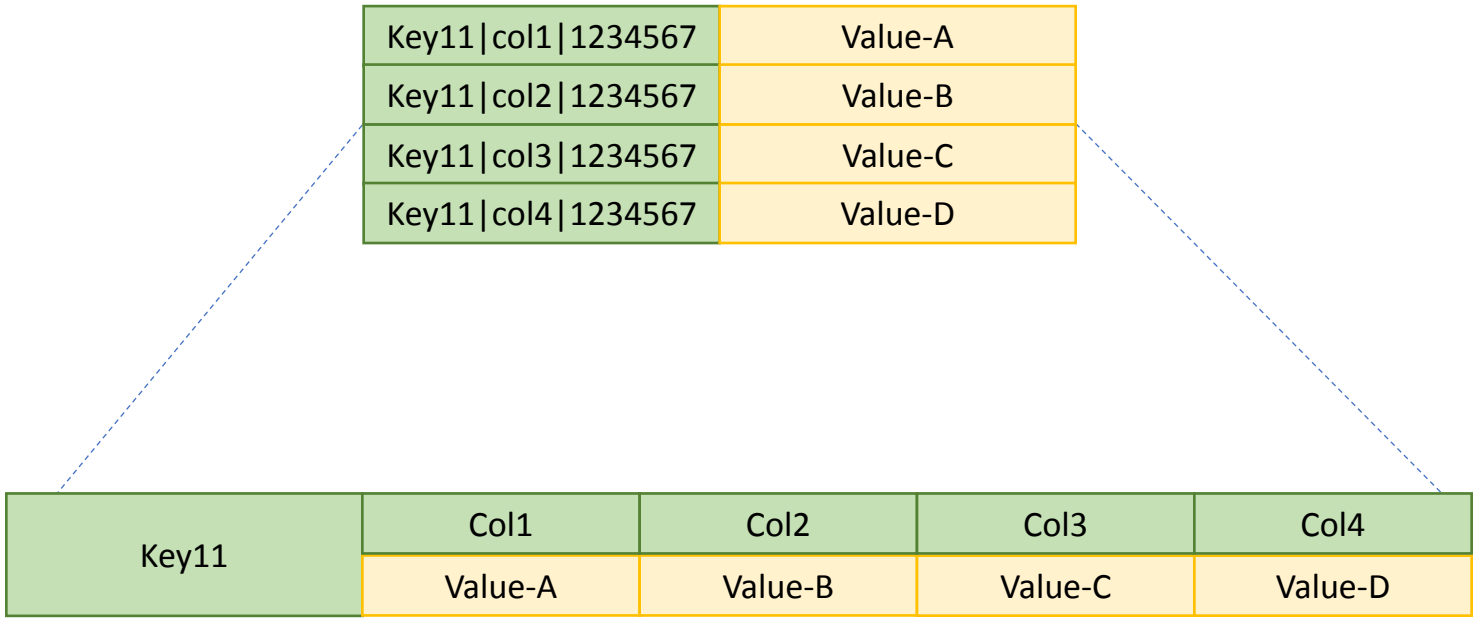


TechAtBloomberg.com

Bloomberg

Engineering

Table Row View

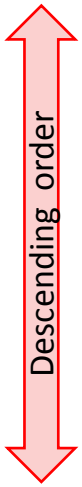


TechAtBloomberg.com

Bloomberg

Engineering

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

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

TechAtBloomberg.com

Bloomberg

Engineering

Data Skew

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-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

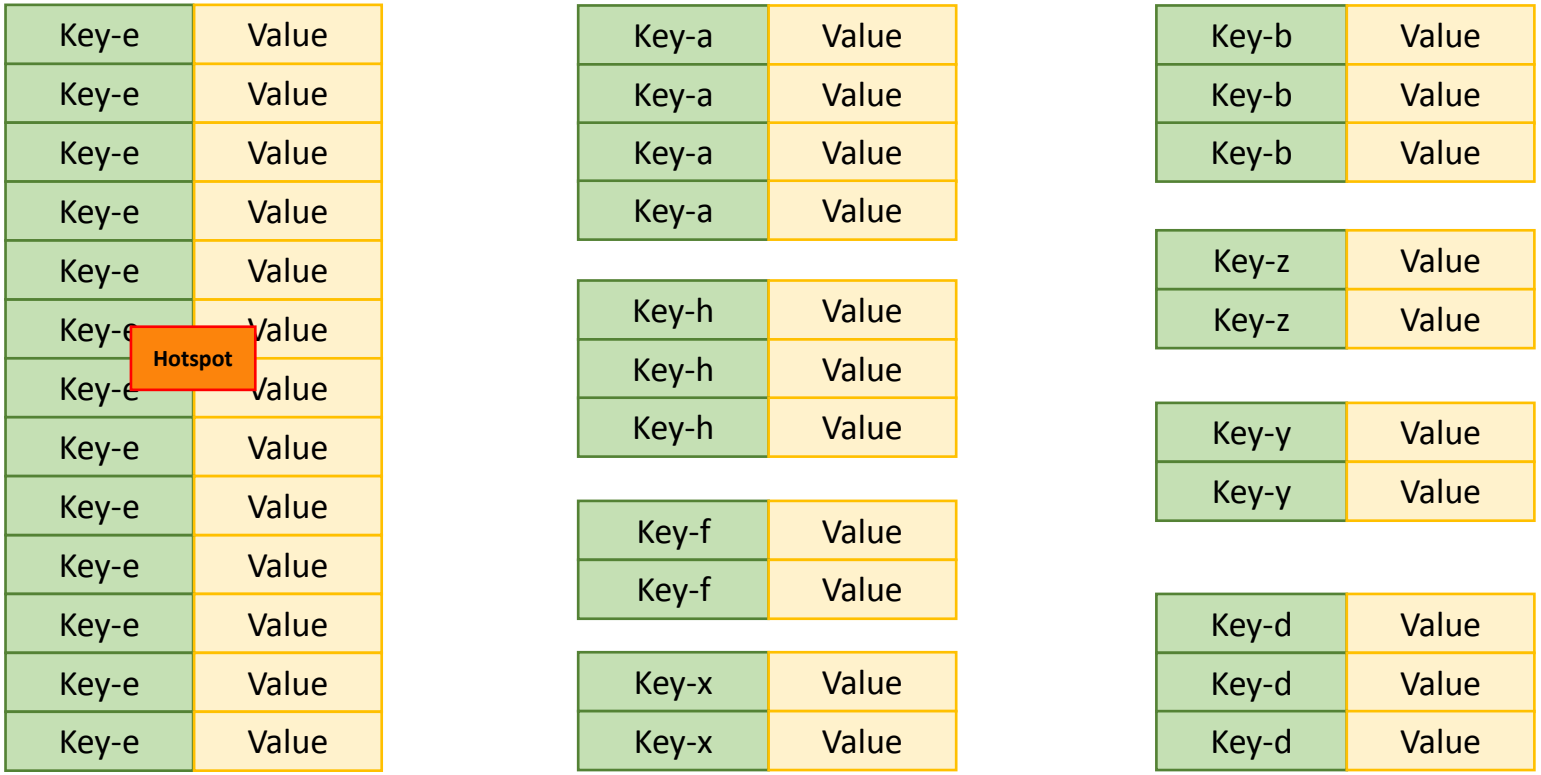
Key-d	Value
Key-d	Value
Key-d	Value

TechAtBloomberg.com

Bloomberg

Engineering

Data Skew

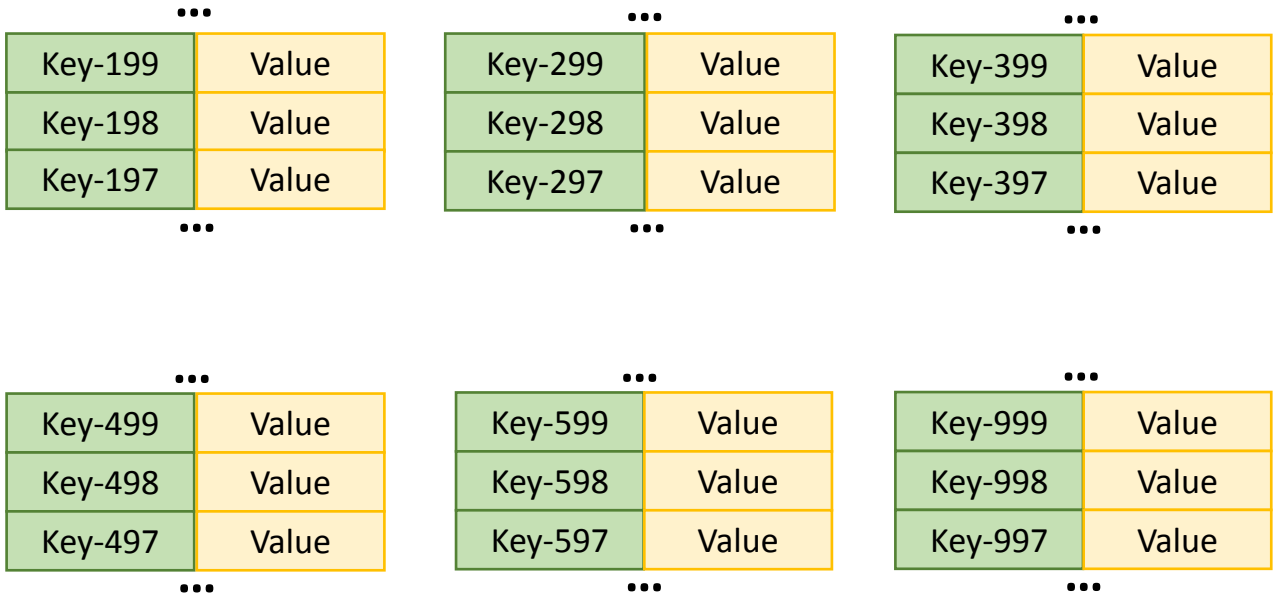


TechAtBloomberg.com

Bloomberg

Engineering

Query Skew

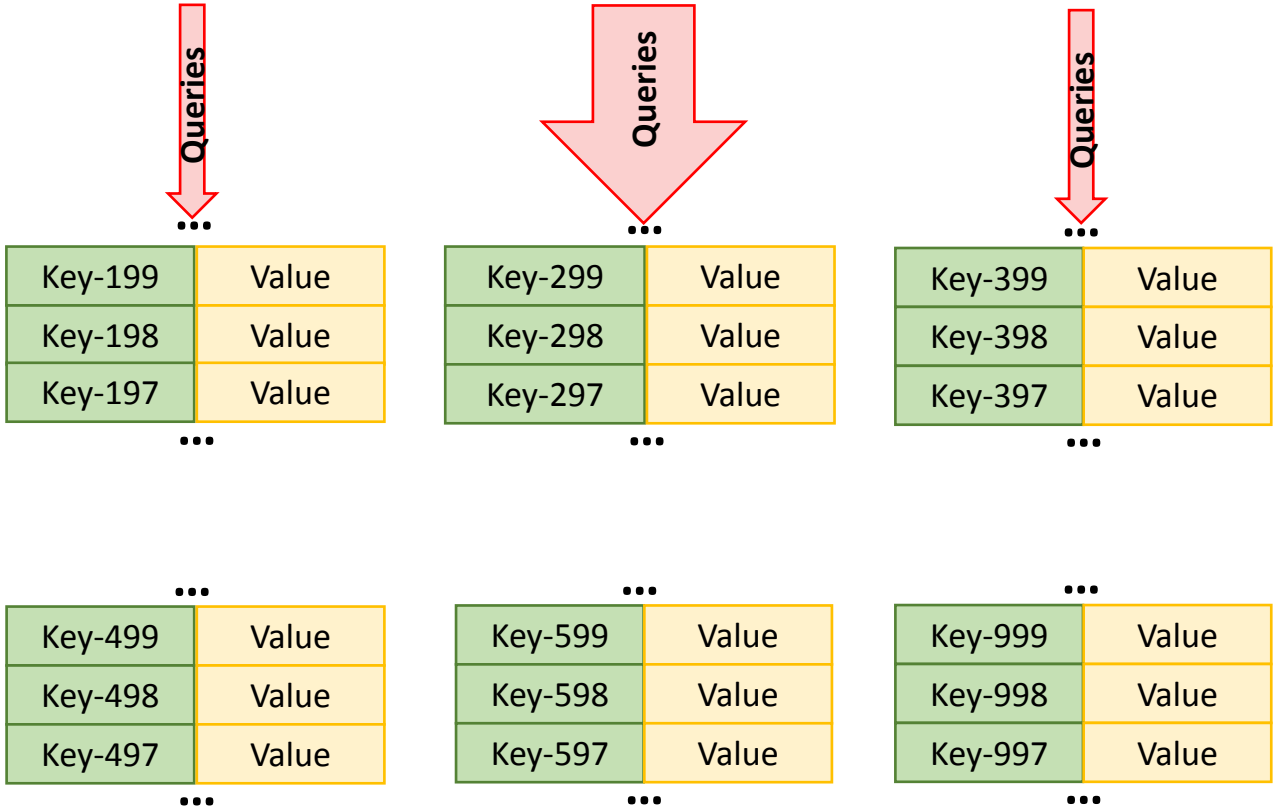


TechAtBloomberg.com

Bloomberg

Engineering

Query Skew

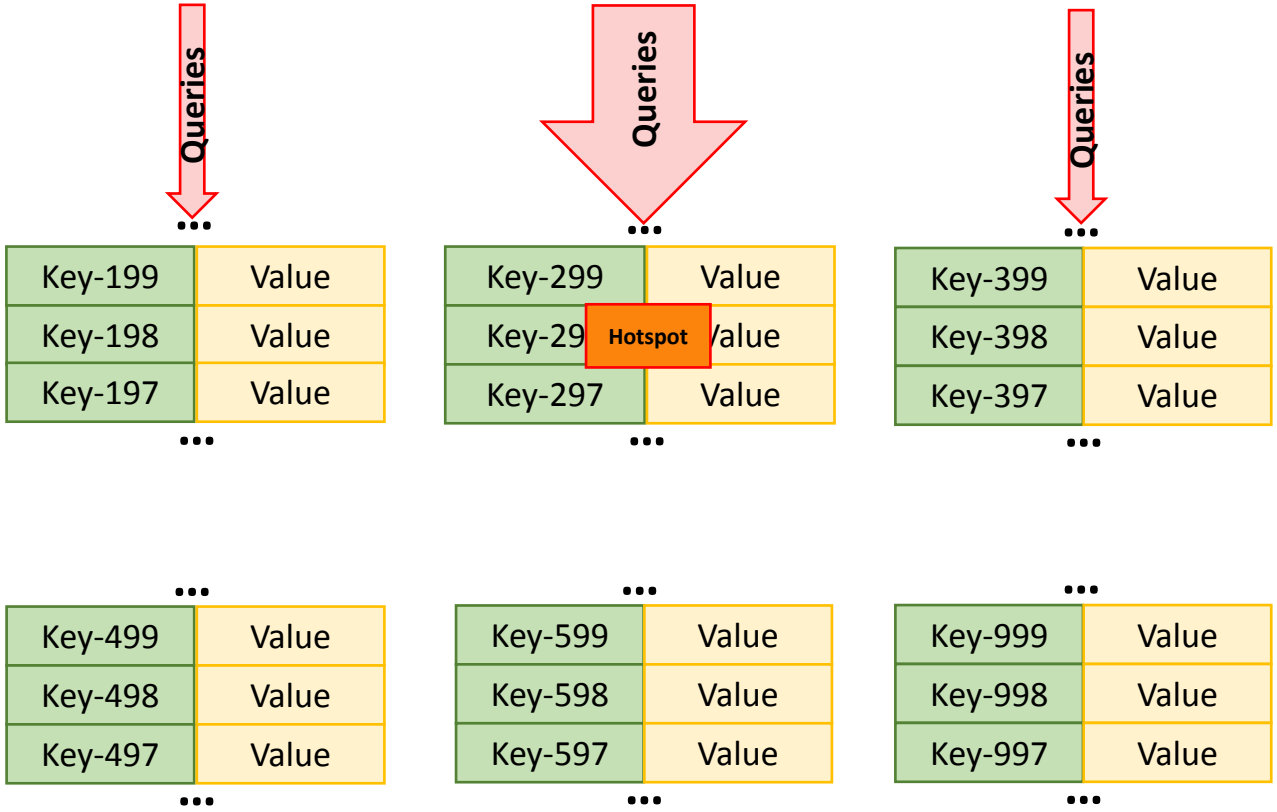


TechAtBloomberg.com

Bloomberg

Engineering

Query Skew

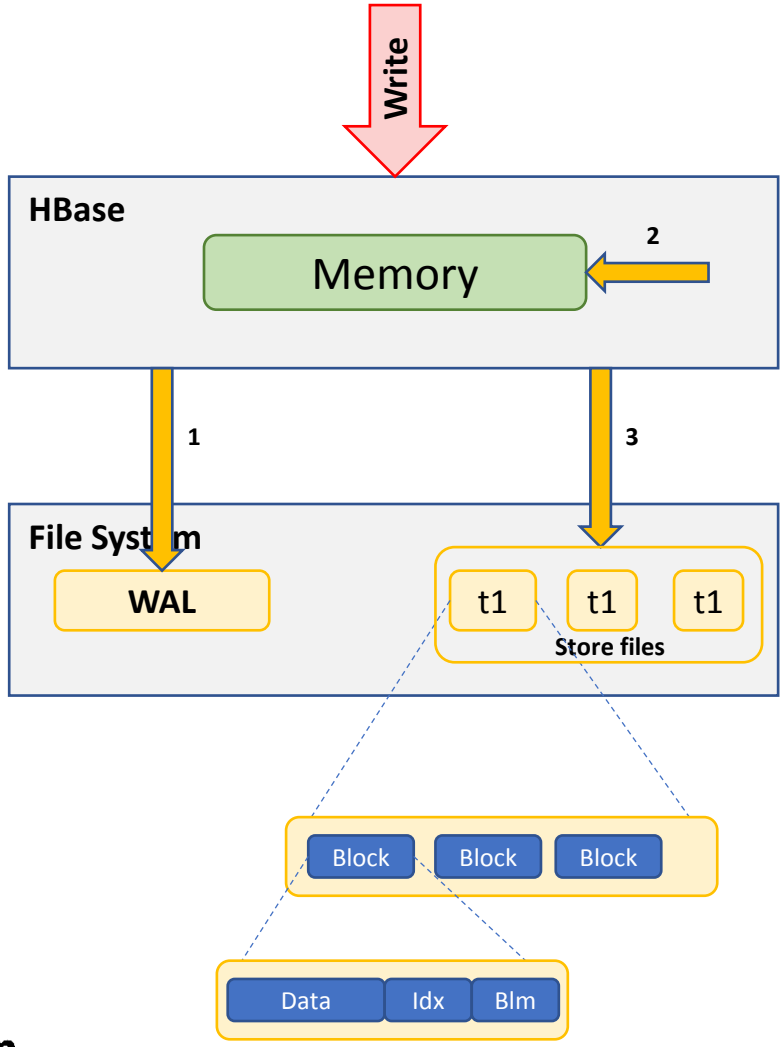


TechAtBloomberg.com

Bloomberg

Engineering

Data Write

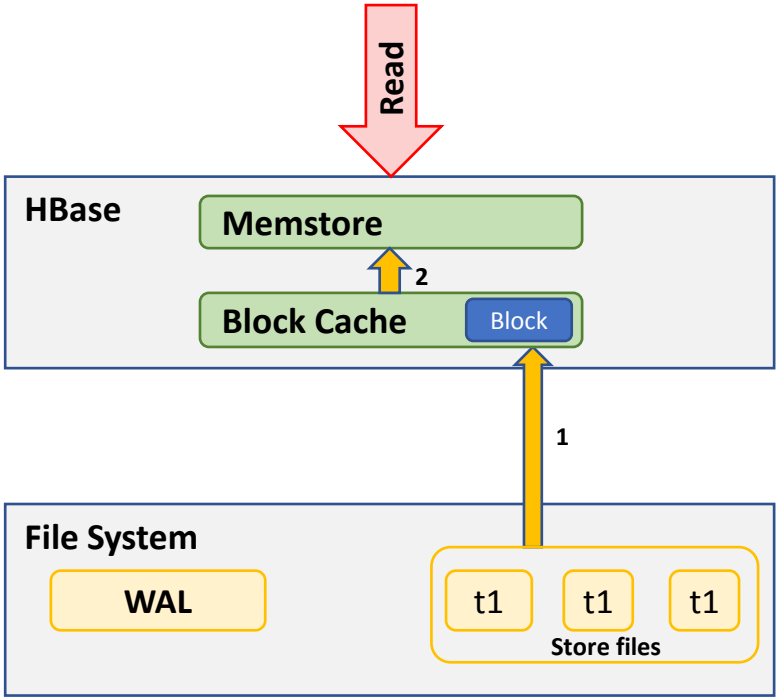


TechAtBloomberg.com

Bloomberg

Engineering

Data Read



TechAtBloomberg.com

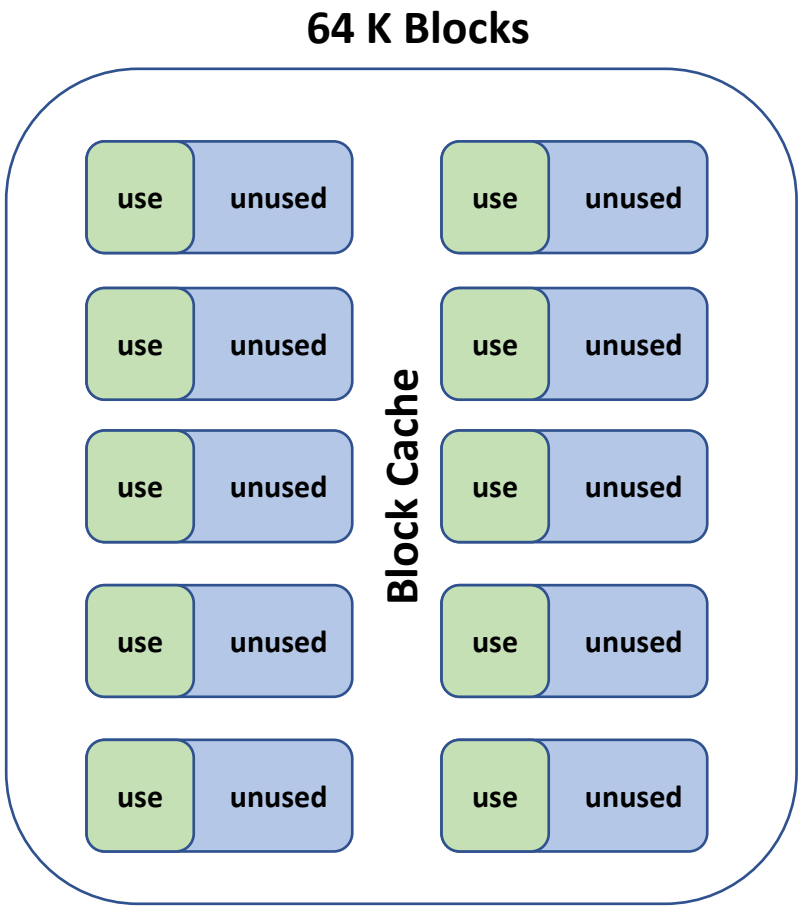
Bloomberg

Engineering

Cache

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

Block Size and Cache Use

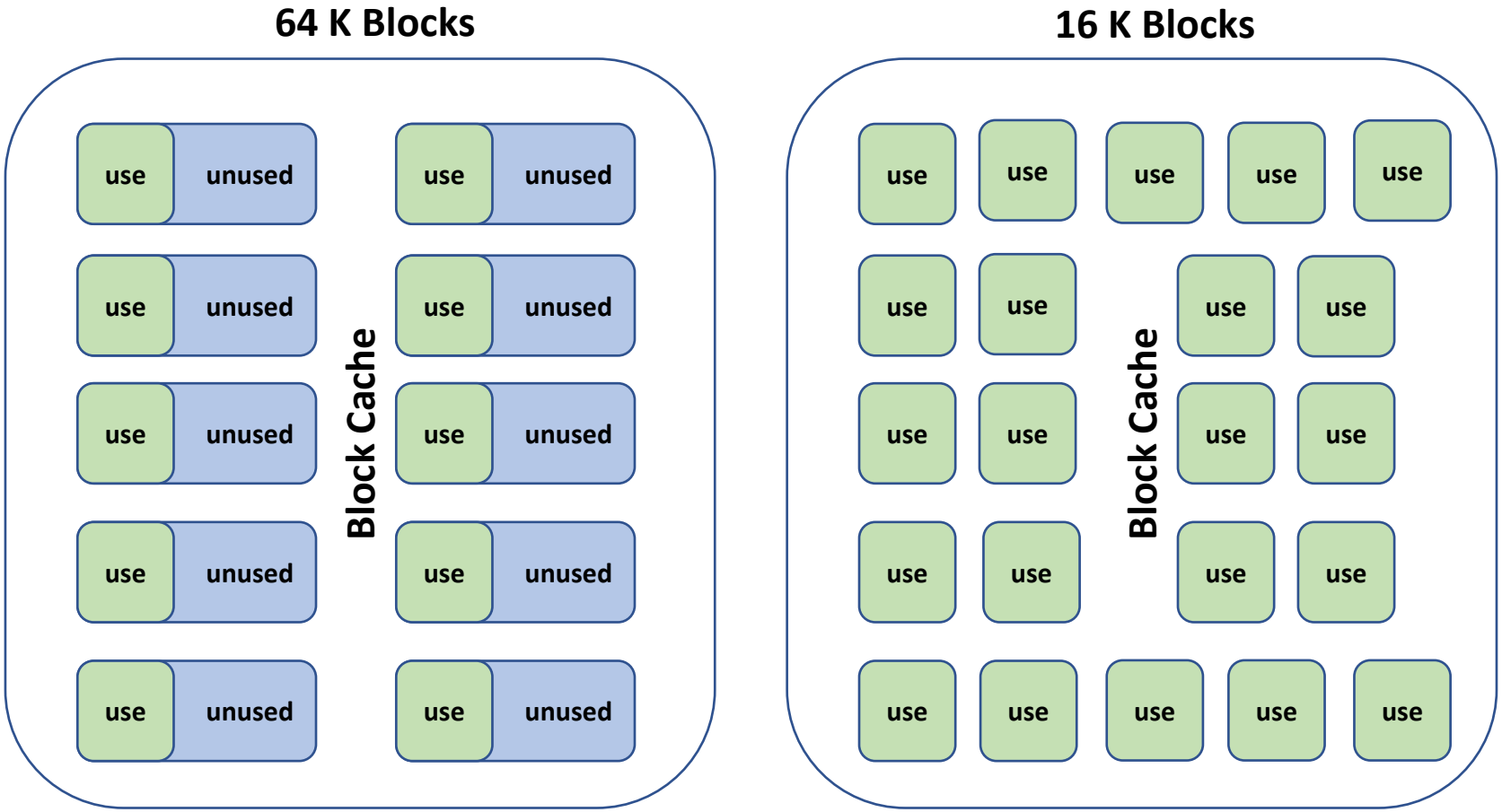


TechAtBloomberg.com

Bloomberg

Engineering

Block Size and Cache Use



TechAtBloomberg.com

Bloomberg

Engineering

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 index block size

TechAtBloomberg.com

Bloomberg

Engineering

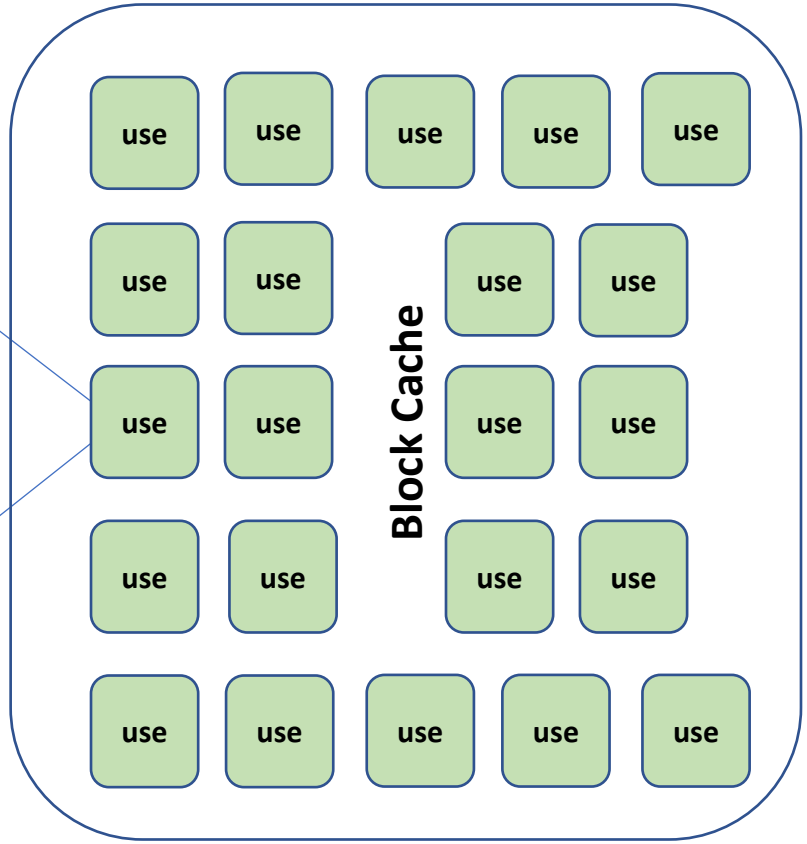
Block Size vs. Index Size

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

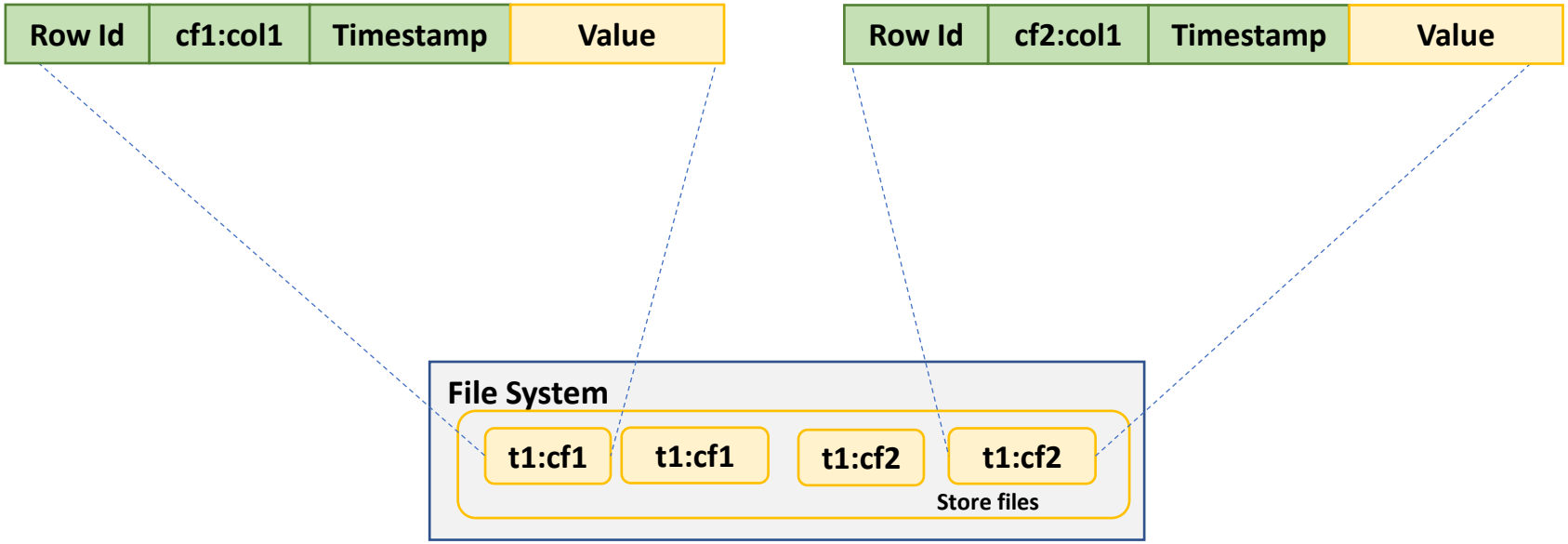
8 K Blocks	
Idx Sz K	Bloom K
472058	2432
574239	2944
331899	1792
471362	2304
517272	2560
469543	2432

Column Family

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



Column Family

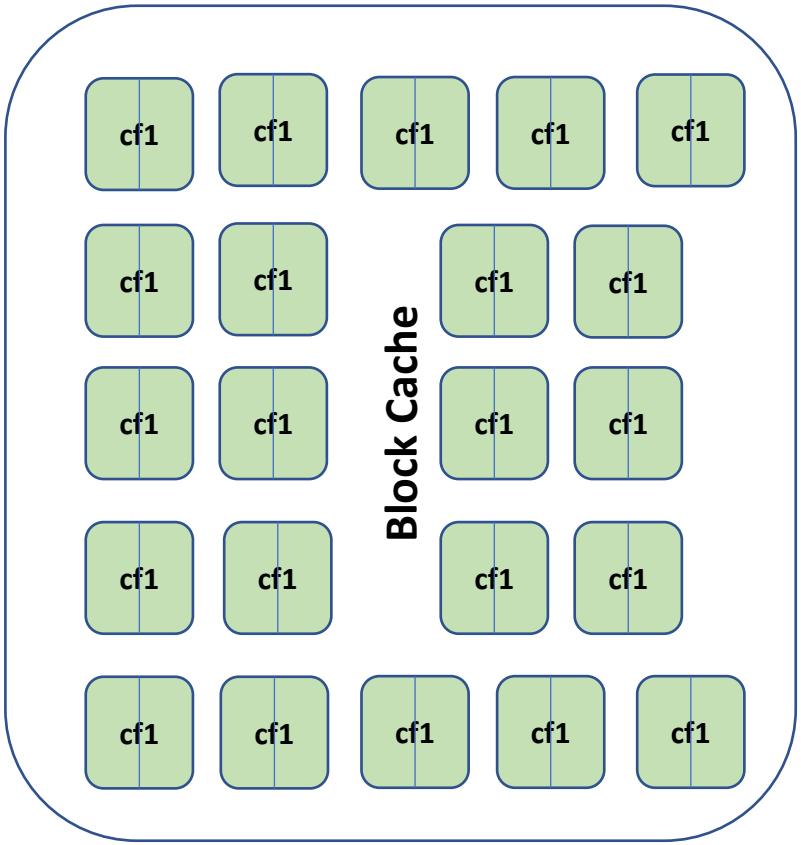


TechAtBloomberg.com

Bloomberg

Engineering

Column Family

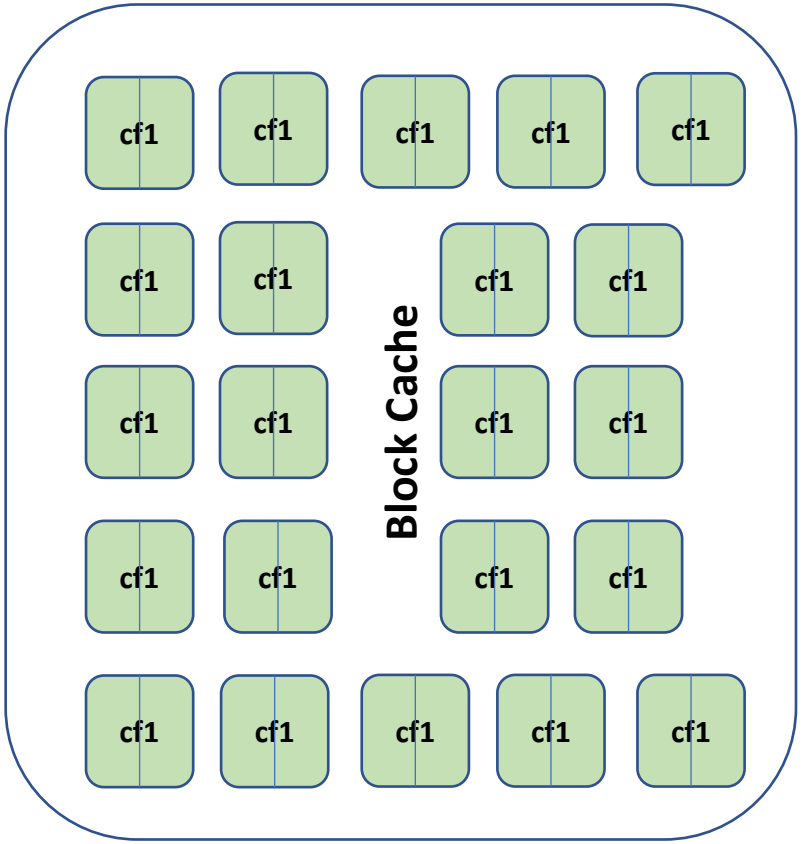


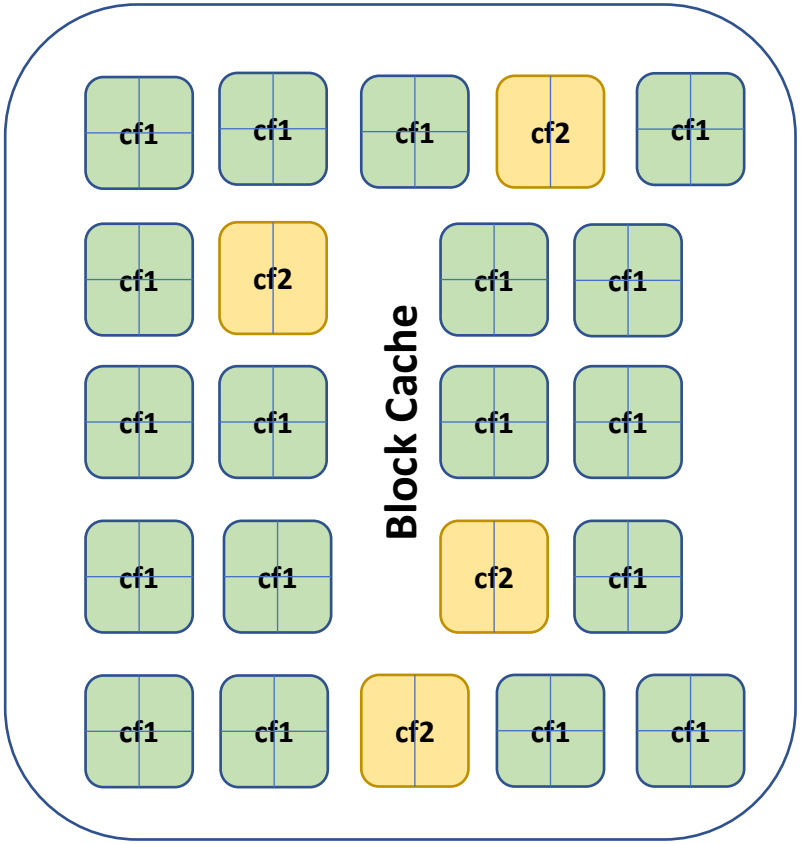
TechAtBloomberg.com

Bloomberg

Engineering

Column Family



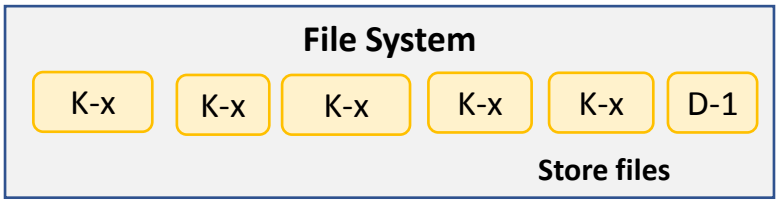


TechAtBloomberg.com

Bloomberg

Engineering

Compaction

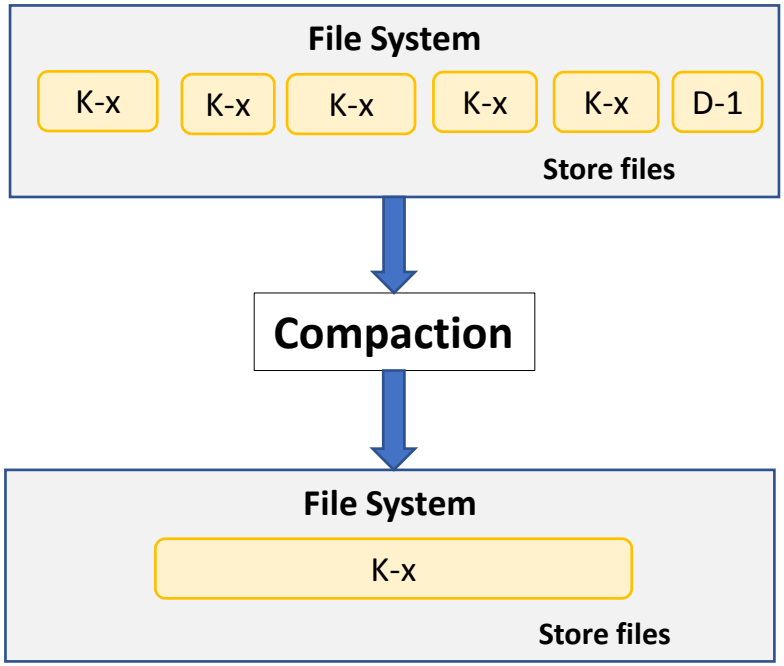


TechAtBloomberg.com

Bloomberg

Engineering

Compaction



TechAtBloomberg.com

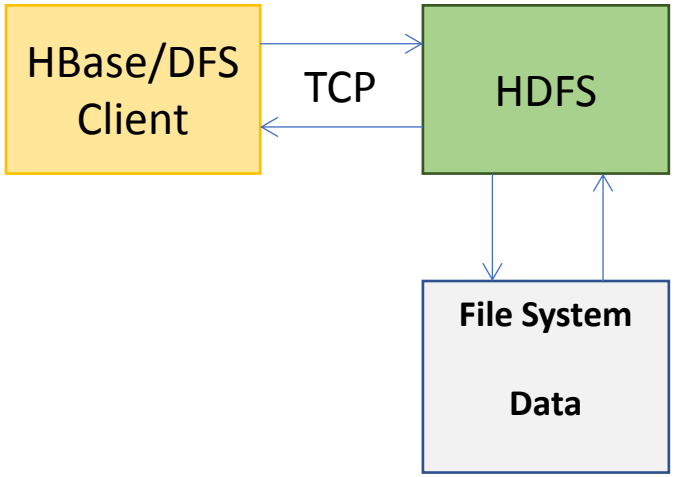
Bloomberg

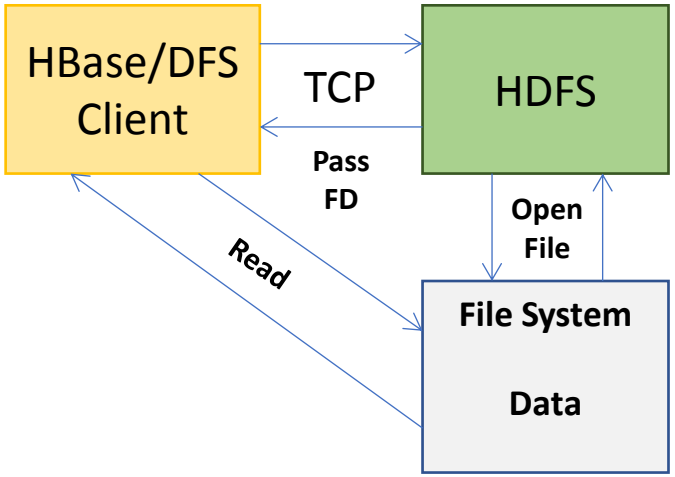
Engineering

Compaction

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

Short Circuit Read



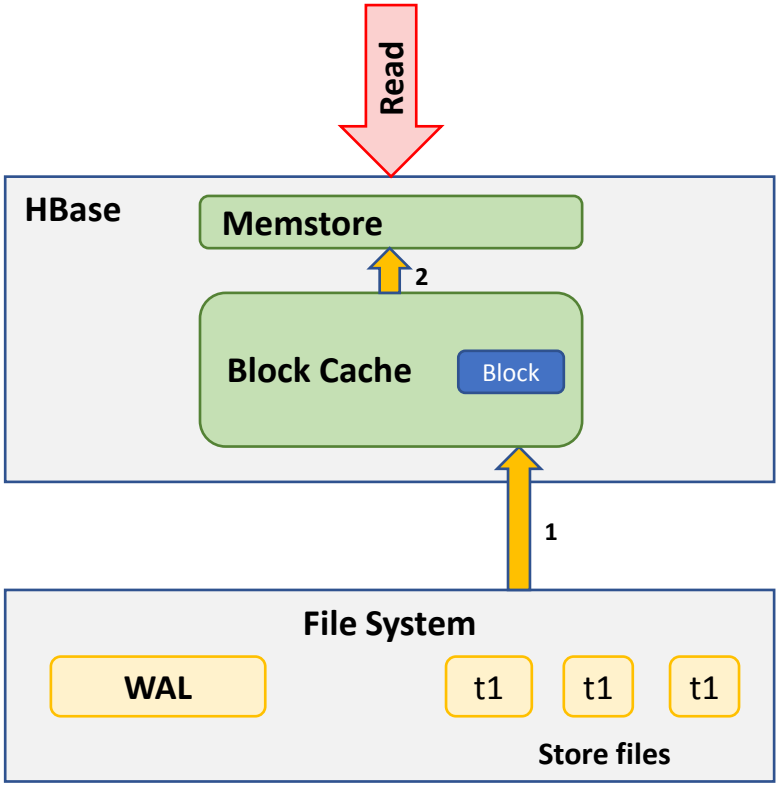


TechAtBloomberg.com

Bloomberg

Engineering

Garbage Collection

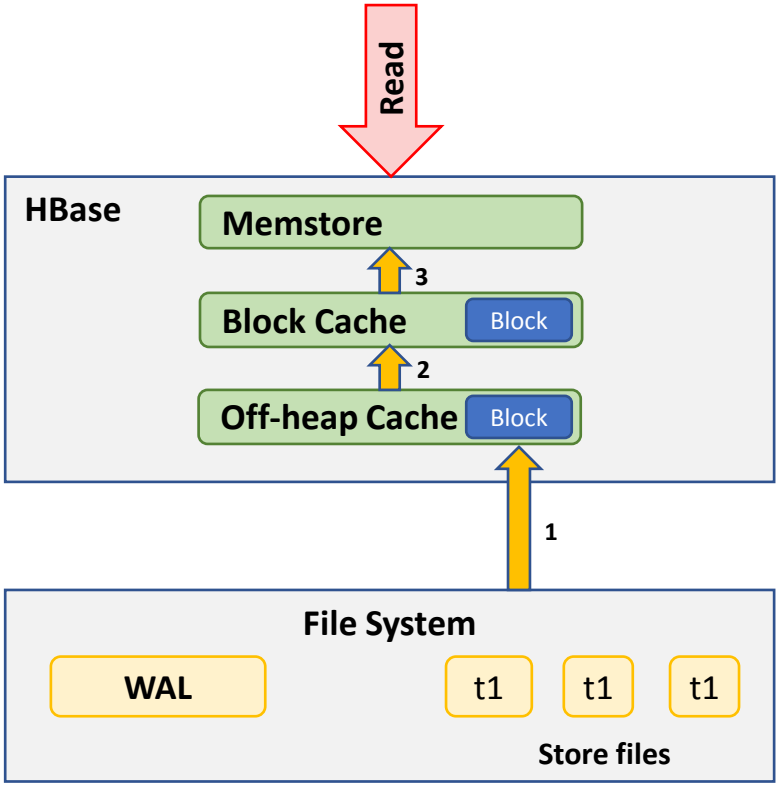


TechAtBloomberg.com

Bloomberg

Engineering

Garbage Collection



Large Cache

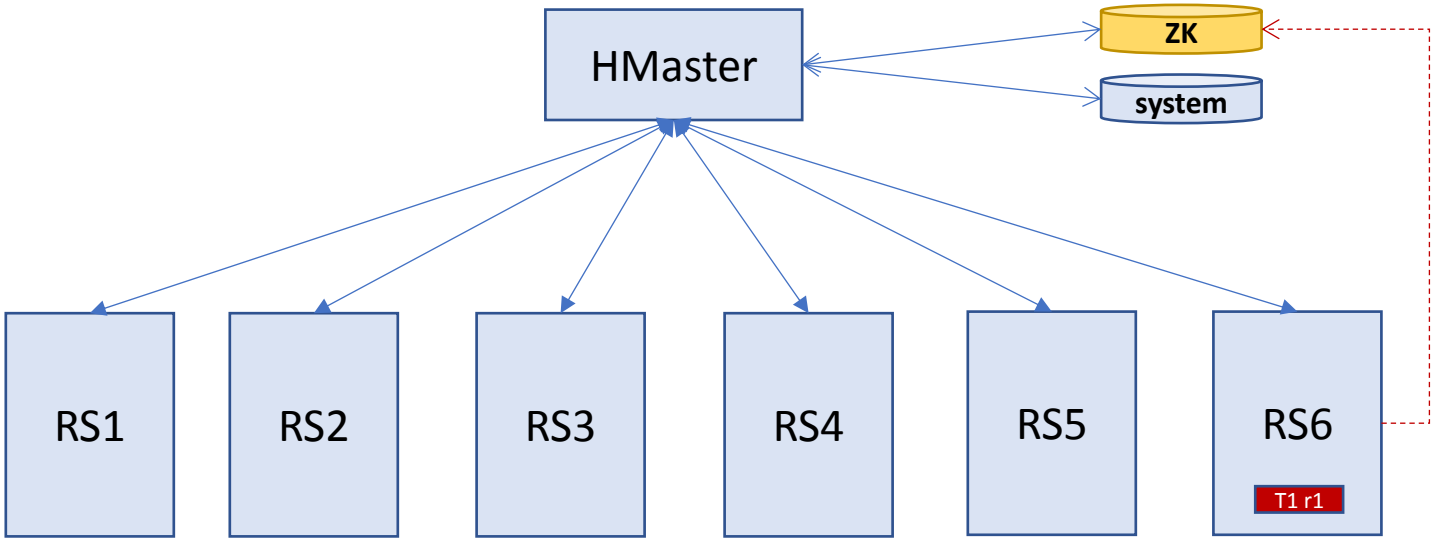
61 GB of Cache					
Avg	2.693	2.814	2.836	2.842	2.812
Median	1	1	1	1	1
95%	8	8	8	8	8
99%	14	14	14	14	15
99.90%	20	20	20	20	20
99.99%	32	31	32	32	33
100.00%	313	319	315	376	341
Max latency	1049	1046	1048	1044	1235

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

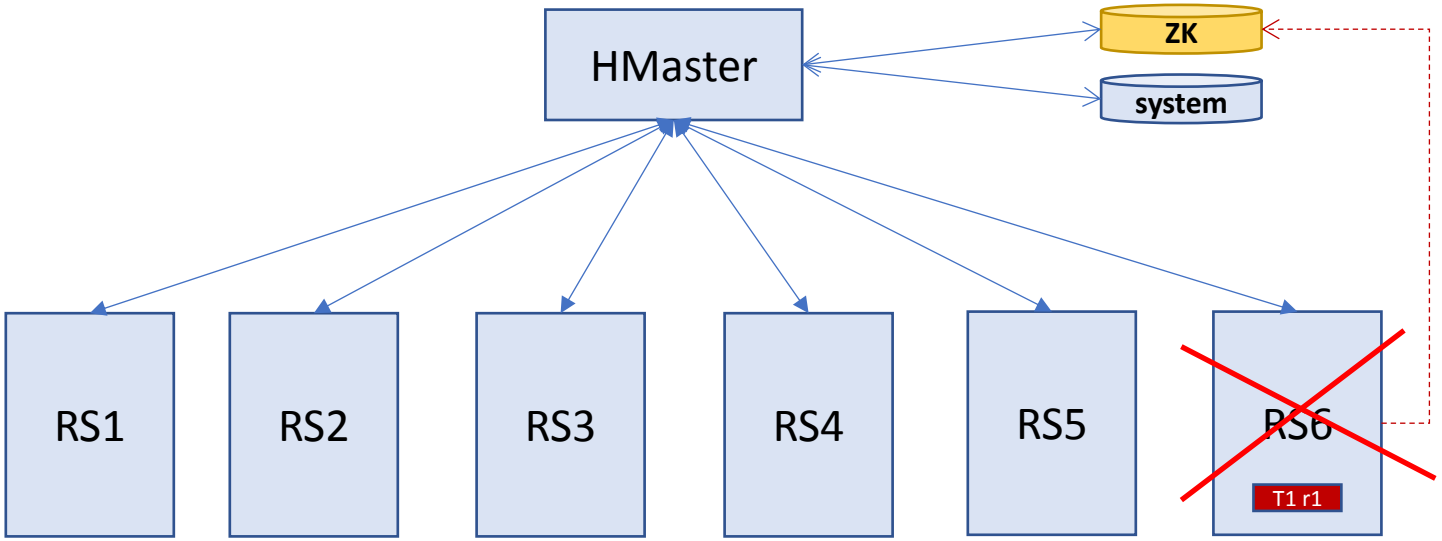
Garbage Collection

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

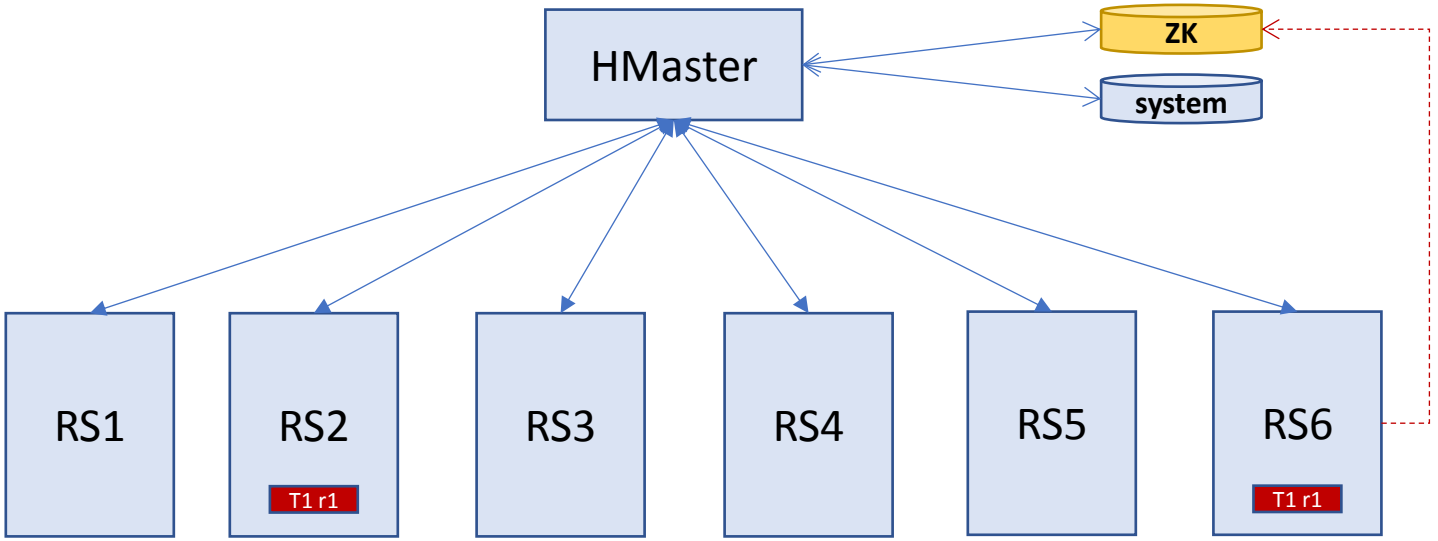
Region Replication



Region Replication



Region Replication



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 needs 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%

Application

- Table pre-split
- Connection reuse
- No read cache/scan cache/batch requests
- Bulk load instead of Put/Batch Mutate
- Column Identifiers
- Code on server
  - Co-processor
  - Filters

TechAtBloomberg.com

Bloomberg

Engineering

Monitoring

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

Summary

- Familiarize system principles
- Use DB development lifecycle
- Ascertain natural fitness for HBase

TechAtBloomberg.com

Bloomberg

Engineering

Thank You!

Bloomberg

Engineering

Reference: http://hbase.apache.org

Connect with Bloomberg's Hadoop Team:
hadoop@bloomberg.net

TechAtBloomberg.com