

China · Beijing

KV Storage Online Production Application in PayPal Risk

Bruce Li

Paypal Risk Tech Infra



[北京站]

主办方 **Geekbang**  **InfoQ**
极客邦科技



促进软件开发领域知识与创新的传播



关注InfoQ官方微信
及时获取ArchSummit
大会演讲视频信息



全球软件开发大会 [北京站]

2017年4月16-18日 北京·国家会议中心

咨询热线: 010-64738142



全球架构师峰会 2016 [深圳站]

2017年7月7-8日 深圳·华侨城洲际酒店

咨询热线: 010-89880682

Agenda

- 1、Paypal Risk Data Challenges
- 2、Considerations on Risk Storage
- 3、Generic KV Storage Data Access Solution
- 4、Experiences Learnt from Aerospike Migration
- 5、Q&A

Challenges of Paypal Risk Data Access

Business Requirements:

- Provide sub-second level high quality risk decision service
- Support rapid business growth
- Support flexible & fast-evolving data schema changes
- Risk decisions should be offline simulatable

Tech Challenges:

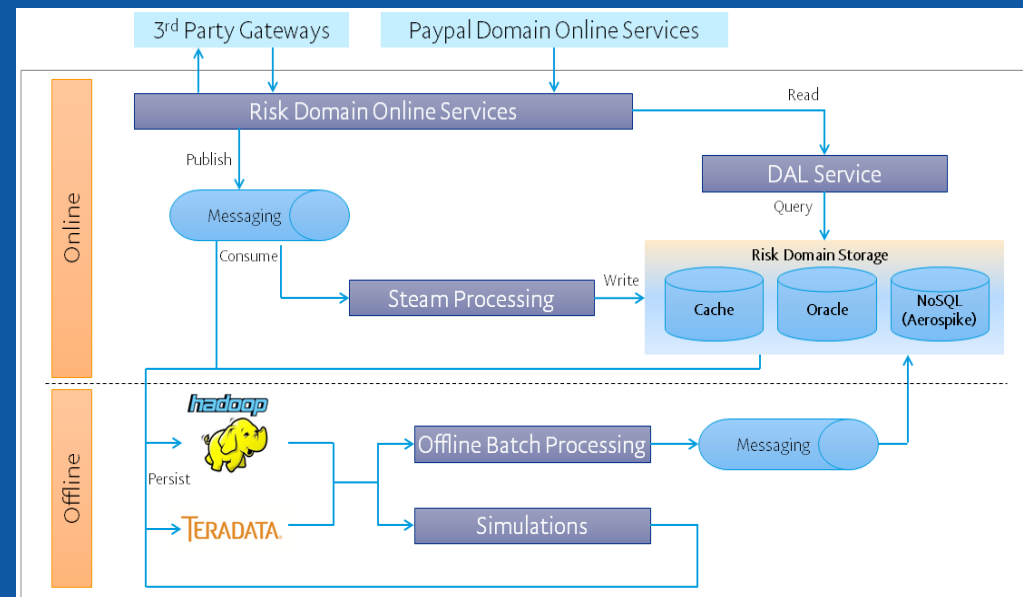
- Low latency & large parallel data loading
- 10X scalability and high availability
- The same schema supports both online & offline



Risk Online Data & Flow

Major Risk Data Set Types

- Real-time events
- Real-time computed data
- Near-real-time computed data
- Offline computed data
- Static data
- Others



Online Risk Data Storage Requirements

	Events	Offline Computed Data	Real-time Computed Data	Static Data
Is KV use case	Storage is not, but event cache can be	Yes	Yes	Not natural KV, but convertible
Join needed	Limited	No	No	Very limited
Row size	Medium-Large	Small-Medium	Medium-Large	Small-Medium
Raw data size	~X00 TB	~X0 TB	~X0 TB	~X0 GB
Column based	Yes	Yes	Yes	Yes
Secondary index needed	No	No	No	No
Need server-side compute on write	Yes (for cache)	No	Yes	No
Need server-side compute on read	Yes	No	Yes	No
Recommended storage	DB + KV cache	KV storage	KV storage/cache	Embedded DB

Agenda

- 1、Paypal Risk Data Challenges
- 2、Considerations on Risk Storage
- 3、Generic KV Storage Data Access Solution
- 4、Experiences Learnt from Aerospike Migration
- 5、Q&A

KV Storage Selection

NoSQL Feature	AeroSpike	Couchbase	Cassandra	MongoDB
SSD Support?	Yes	Yes	Yes	Yes
Type	Key/Value (& Columnar)	Document (JSON)	Columnar	Document (BSON)
User Defined Functions (UDF) or Scripting?	Yes (Lua)	No	Yes (Java)	Yes (JavaScript)
Partial Read/Write Support?	Yes	No	Yes	No
Server-side Computation?	Yes	No	Yes	Yes
Complex data type Support?	Yes	Yes (JSON)	Yes	Yes (JSON)
Secondary Indexes?	Yes	Yes	Yes	Yes
TTL expiration?	Yes	Yes	Yes	Yes
Cross Data Center Replication?	Yes	Yes	Yes	No
Range Queries?	Yes (not mature enough)	No	Yes	No
Automatic failover & rebalancing?	Yes	Yes	Yes	Yes
Auto-Sharding	Yes	Yes	Yes	Yes (not mature enough)
Tunable Consistency levels?	Yes	No	Yes	Yes
SQL Like Interface?	Yes	No	Yes	No
Aggregation Query Support?	Yes	No	Yes	No
Share-nothing Architecture?	Yes	Yes	Yes	No (Master/slave)
Cross Datacenter Replication (XDCR/XDR)?	Yes	Yes	Yes	Yes
Performance / Scalability Concerns	N/A - Green Light!	N/A	GC Pauses	Master/slave SPOF

KV Storage & Data Access

KV Storage

- Generic KV storage abstraction
- Won't be locked in by a specific solution
- Separate operations and business

Data Access

- Flexible mapping from data set to KV
- Data compaction
- Metadata driven, support fast-evolving schema changes
- Very high throughput but low latency

Data Access Layered Abstraction



Data Set

- Multi-columns but only one key



Data Access

- Generic async/sync API , e.g. get/put/compute/batch/metadata API
- Data set to KV mapping
- Data set name, column name dictionary



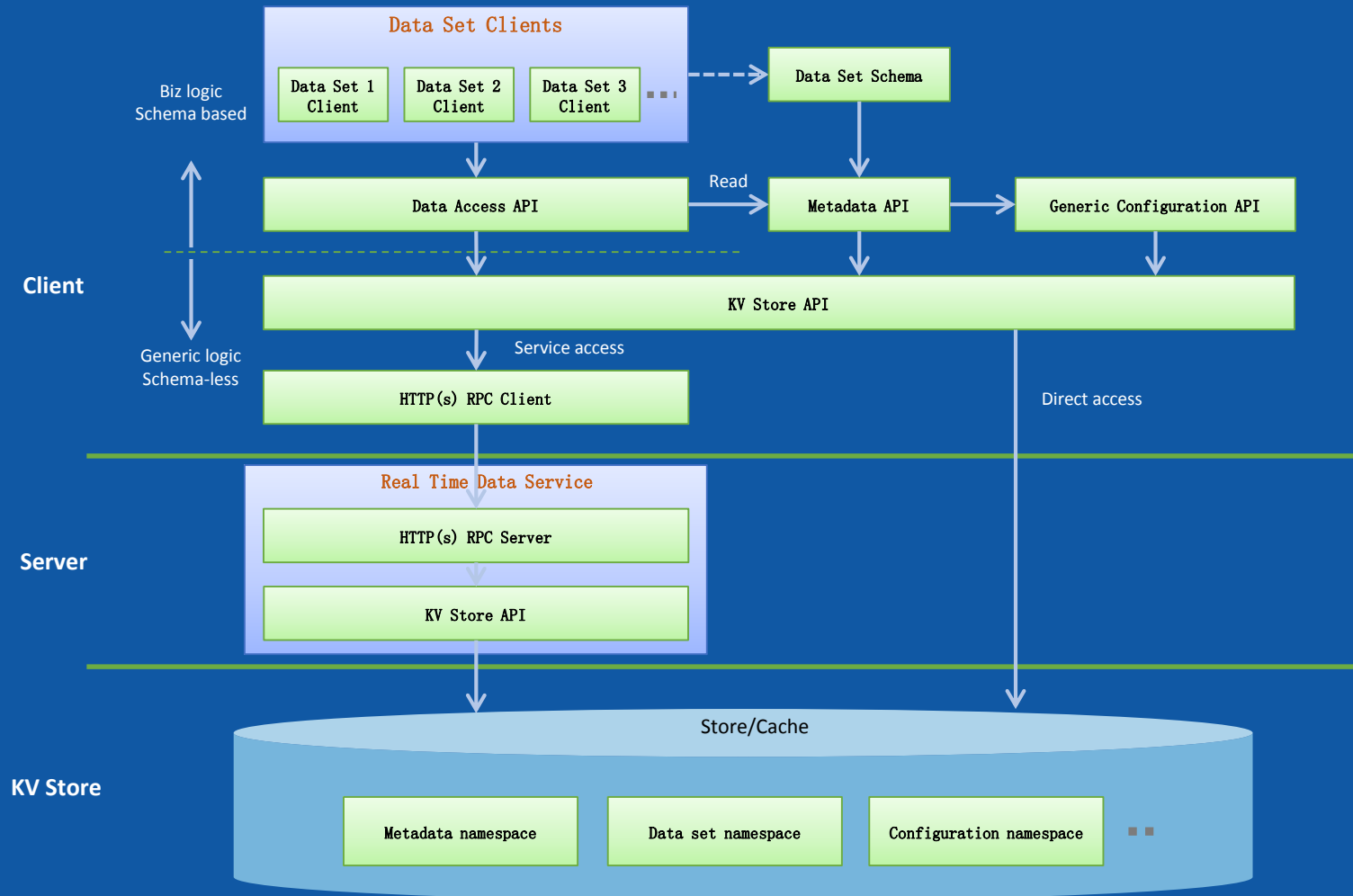
KV Storage/Cache

- Sharding, node add/remove, XDR, data migration...

Agenda

- 1、Paypal Risk Data Challenges
- 2、Considerations on Risk Storage
- 3、Generic KV Storage Data Access Solution
- 4、Experiences Learnt from Aerospike Migration
- 5、Q&A

Generic KV Data Access Design



Generic KV Data Access API

Read API

- Get, exists

Write API

- Insert, update, upsert, delete, CAS based update/insert/...

Server-side Compute API

- Increase/decrease, list/map/set operations, user defined function (UDF)

Batch API

- Read/write/compute/...

KV Data Access Service Considerations

High Throughput

- Customized async RPC based on Netty, lock-free implementation
- Async storage client, always use batch when possible

Low Latency

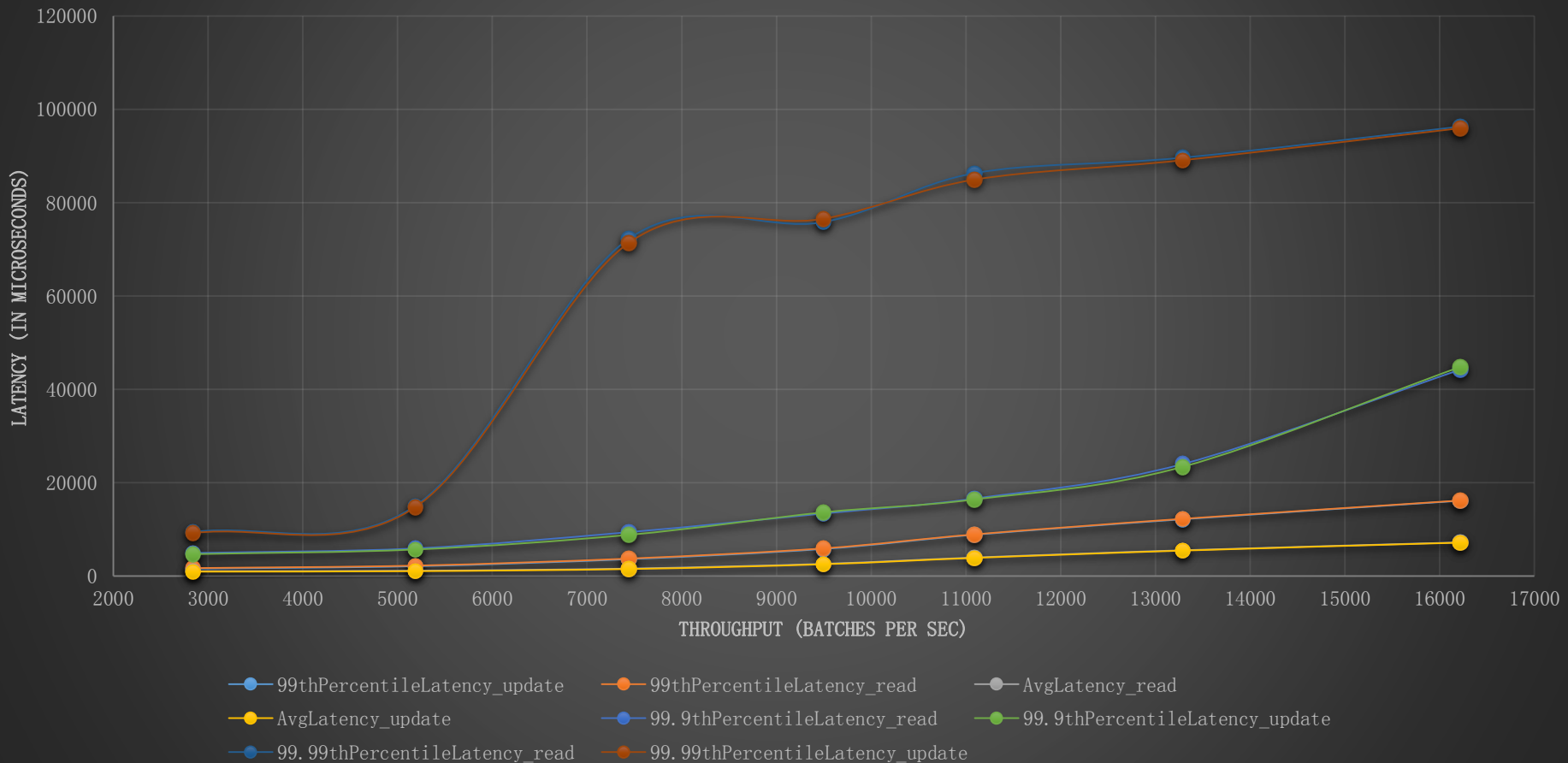
- De/Serialization on direct NIO buffers
- Avoid unnecessary object creation/memory copy, GC optimizations
- Release unused objects as earlier as possible
- Workaround HTTP/1.1 limits, or embrace HTTP/2

Isolation

- Cluster vs. node level
- Connection level
- JVM level

Asynchronous Data Access Service Benchmark

E2E Client-Service-Aerospike Benchmark: Read 50% Write 50%
Latency vs. Throughput (4-core VM)



Agenda

- 1、Paypal Risk Data Challenges
- 2、Considerations on Risk Storage
- 3、Generic KV Storage Data Access Solution
- 4、Experiences Learnt from Aerospike Migration
- 5、Q&A

Why Aerospike

- Share-nothing architecture
- Best performance among the other options
- User defined function support
- Native XDR support

Aerospike Online Performance

Sherlock

Search for pools/correlation id here.

Time (POT): 1h 3h 8h 1d 3d 1w Absolute Time Compare

Use hour boundary

< Hour >

< Day >

< Week >

2016/10/12 01:15 - 2016/10/13 01:15

Local logs

CAL logs

PoolViewer

Pool

By Alias

all

silc

dcg11

dcg12

sica

phx

dcg01

dcg02

dcg13

raz01

Pool Summary

Machine Data

Inbound Calls

Outbound Calls

Inbound Connections

Outbound Connections

Clients

Table

SQL

Events

Transactions

Build Data

Heartbeat

Transaction Type

nskaero4-storage1

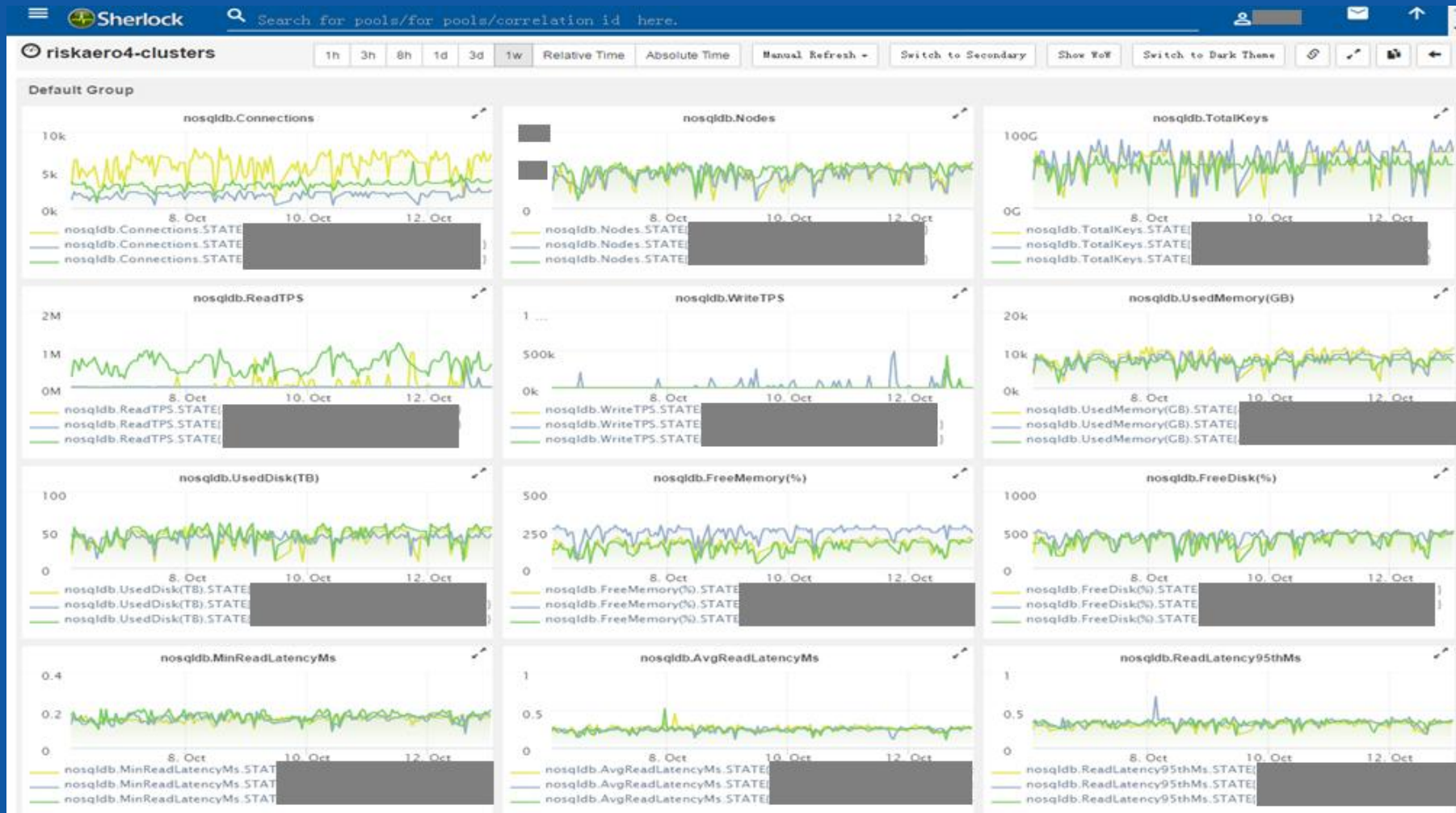
Search

nskaero4-storage1

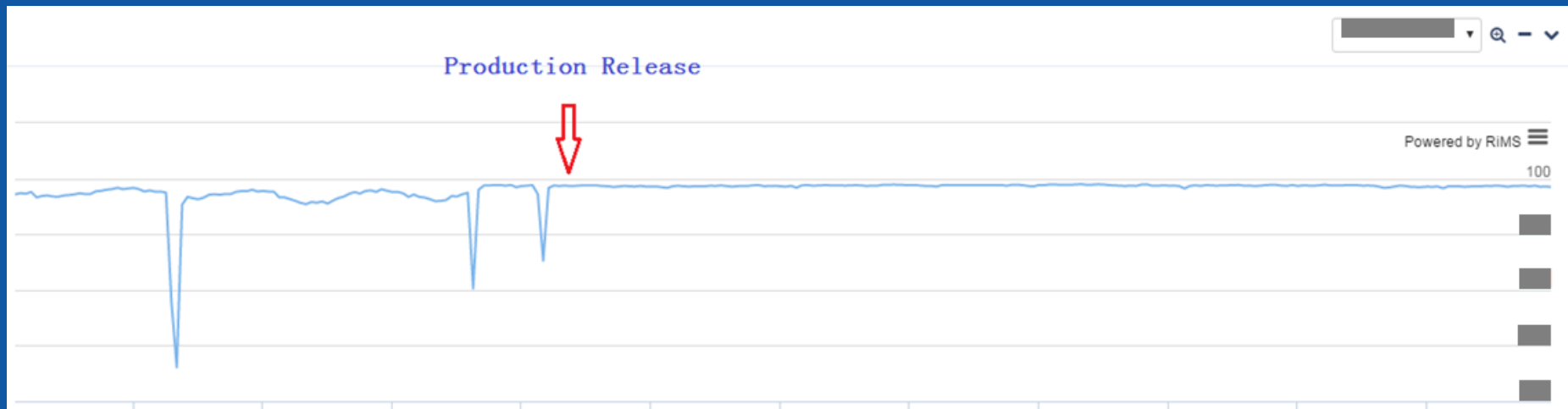
Download As CSV

Name	Colo	Count	Failure Count	Failure %	Min (ms)	Max (ms)	Avg (ms)	Median	StdDev	95.0 %ile(ms)	99.0 %ile(ms)
get	sica	68	0	0	0	0	7	0.25	0	0.95	1.12
get	sica	26	0	0	0	0	4	0.31	0	0.87	1.75
Qaero batch	sica	117,596,709	2,192	0.00	0	585	1.19	1	1.08	2	3.00
Qaero batch	sica	194,673,404	2,214	0.00	0	943	0.94	1	0.99	2	3
get	sica	10,072	0	0	0	29	0.18	0	0.77	1	2.90
get	sica	3,424	8	0.23	0	101	0.45	0	5.56	1	2.79
get	sica	4,030	10	0.25	0	101	0.42	0	5.29	1	2.43
get	sica	202	0	0	0	4	0.15	0	0.52	1	2.33
get	sica	4,238	1	0.02	0	99	0.19	0	1.67	1	2.07
get	sica	35,991	0	0	0	63	0.13	0	0.63	1	2
get	sica	510,543	0	0	0	157	0.19	0	0.64	1	2
get	sica	132,396	0	0	0	62	0.15	0	0.62	1	2
get	sica	1,048,903	7	0.00	0	137	0.13	0	0.64	1	2
get	sica	130,449	0	0	0	56	0.15	0	0.50	1	2
get	sica	14,119	0	0	0	14	0.19	0	0.49	1	2
get	sica	484,392	2	0.00	0	101	0.13	0	0.59	1	2
get	sica	552,563,383	1,062	0.00	0	821	0.13	0	0.59	1	2
get	sica	34,260	1	0.00	0	101	0.13	0	1.00	1	2
get	sica	31,854	0	0	0	87	0.12	0	0.66	1	2
get	sica	438,010	28	0.01	0	104	0.13	0	1.02	1	2
get	sica	36,116	5	0.01	0	102	0.15	0	1.32	1	2
get	sica	162,242	2	0.00	0	100	0.17	0	0.70	1	2
get	sica	35,039,966	118	0.00	0	436	0.18	0	0.67	1	2
get	sica	896,282	2	0.00	0	108	0.13	0	0.60	1	2
get	sica	17,835,743	68	0.00	0	383	0.16	0	0.62	1	2

Aerospike Online Performance – Cont.



ATB Improvement with Aerospike Adoption



Web Data Query Tool

The screenshot displays the 'Risk Data Tool v1.2' interface. On the left, a sidebar shows a tree view of data sets under 'qa paypal.com', including 'Aerospike1', 'Aerospike2', and 'Aerospike4'. The main area shows a query for 'ACCOUNT_2ND' with a filter 'Filter with Data Set Name...'. Below the filter, there are tabs for 'Metadata', 'Edit', and 'Collapse', and a dropdown for 'All 262 Column(s)'. The results are displayed in a table with columns for 'Key' and 'Value'. The 'Key' column shows keys like 'bin:80', 'bin:101', 'bin:109', and 'bin:142'. The 'Value' column shows JSON data for each key, including fields like 'type', 'slidingwindow', 'timeDecay', 'array', 'entry', 'pos', 'value', and 'lastUpdatedMls'.

Key	Value
bin:80	{ "type": "Slidingwindow", "slidingwindowValue": { "array": { "entry": { "pos": 5, "value": { "value": 1 } } }, "entry": { "pos": 26, "value": { "value": 1 } }, "entry": { "pos": 34, "value": { "value": 1 } } }, "lastUpdatedMls": 1457098097 }
bin:101	{ "type": "TimeDecay", "timeDecayValue": { "array": { "value": { "value": 1 } }, "value": { "value": 1 } }, "value": { "value": 1 }, "lastUpdatedMls": 1457099315 }
bin:109	{ "type": "TimeDecay", "timeDecayValue": { "array": { "value": { "value": 1 } }, "value": { "value": 1 } }, "lastUpdatedMls": 1457099297 }
bin:142	{ "type": "Slidingwindow", "slidingwindowValue": { "array": { "entry": { "pos": 6, "value": { "value": -1952 } } }, "entry": { "pos": 27, "value": { "value": -1952 } }, "entry": { "pos": 34, "value": { "value": -1952 } } }, "lastUpdatedMls": 1457098339 }

Experiences Learnt from Aerospike Migration

- Control the size for a single Aerospike cluster
- Balance the data density for each node
- UDF should be simple & fast enough
- Leverage asynchronous client
- Provide different tools for easier user adoption
- Build benchmark tool & automate at the beginning

Agenda

- 1、Paypal Risk Data Challenges
- 2、Considerations on Risk Storage
- 3、Generic KV Storage Data Access Solution
- 4、Experiences Learnt from Aerospike Migration
- 5、Q&A

THANKS



[北京站]

主办方 **Geekbang** > **InfoQ**
极客邦科技