

Gain a MongoDB Advantage with the Percona Memory Engine

Webinar

December 6th, 2017

Vadim Tkachenko

CTO, Percona



Me

- CTO at Percona
- Leading Software Engineering at Percona
- Technology & (database, hardware, filesystems) Performance Enthusiast
- Current areas of interest: MySQL, MongoDB and ClickHouse
- @VadimTk

Memory Engine

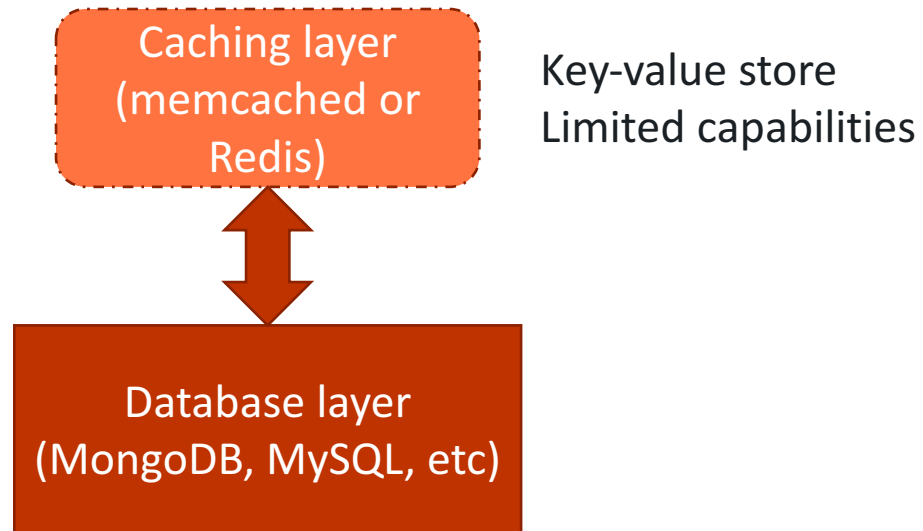
- <https://www.percona.com/software/mongo-database/percona-memory-engine-for-mongodb>
- Storage Engine in Percona Server For MongoDB
- Based on wiredTiger
 - but without writes to the storage
- High Write Throughput
- High Read Throughput
- Predictable, low latencies

Memory Engine use cases

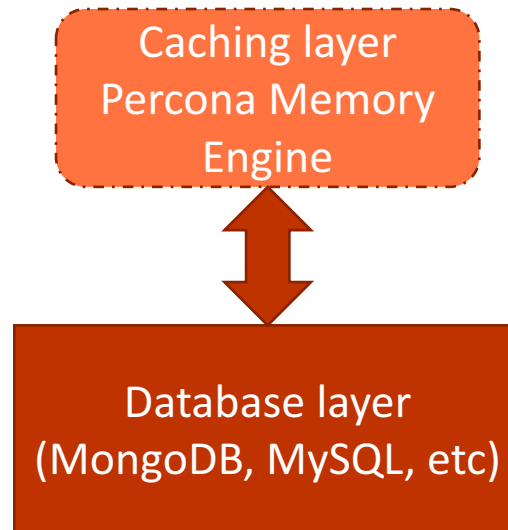
Application cache

Replace services such as memcached and custom application-level data structures with the full power of MongoDB features.

Typical architecture



Better way



- Document store
- Flexible query language
- Replication / sharding
- Familiar MongoDB deployment

Session management

Decrease application response times by keeping active user sessions in memory.

Transient runtime state

Store application stateful runtime data that doesn't require on-disk storage.

Multi-tier object sharing

Facilitate sharing of data in multi-tier/multi-language applications.

Application testing

Reduce turnaround time for automated application tests.

Sophisticated data manipulation

Increase performance for data manipulation operations such as aggregation and map reduction.

Real-time Analytics

Uses in-memory computing in situations where response time is more critical than persistence.

Memory Engine deployments

Standalone



PSMDB Memory

- For testing and development
- Useful for sizing and quick resets
- No Redundancy
- No Persistence
- Limited to server size

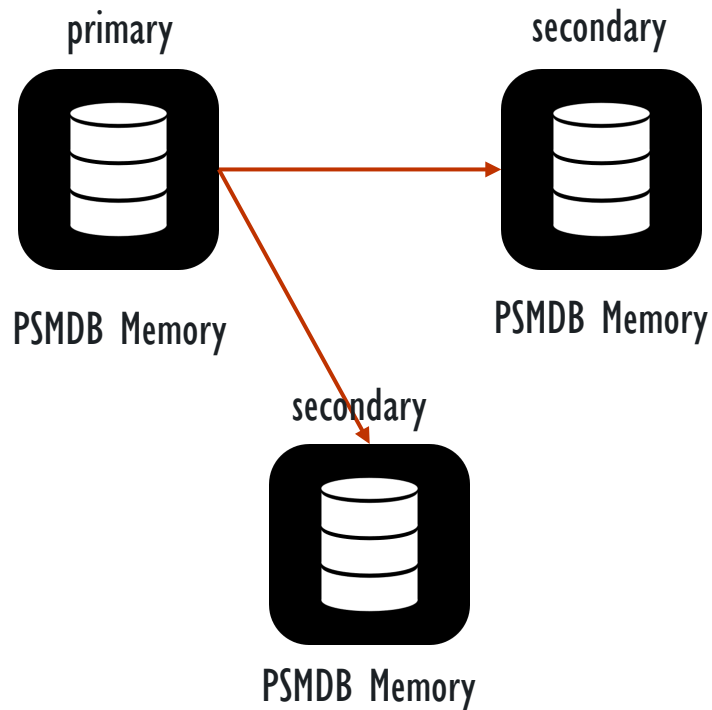
Standalone config

```
./mongod --storageEngine inMemory --dbpath /data/db --inMemorySizeGB 150
```



PSMDB Memory

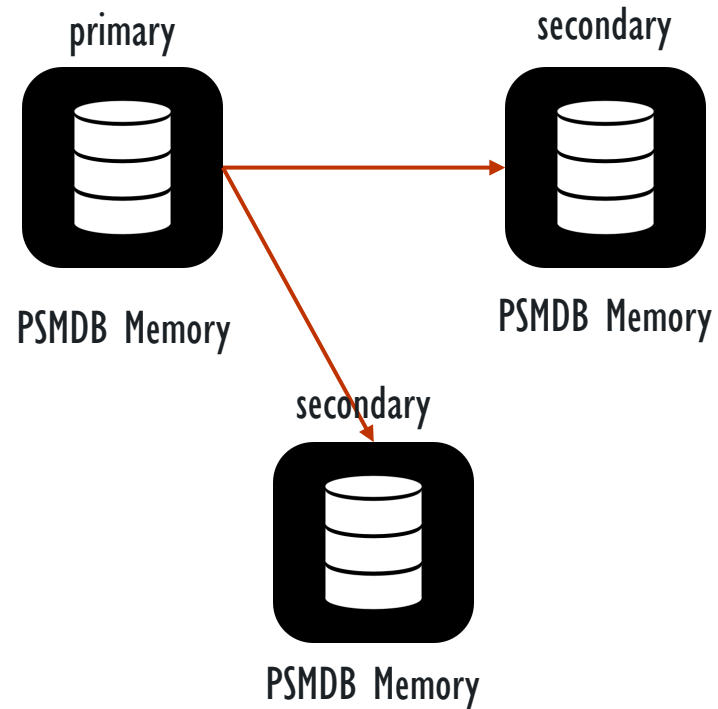
Replica Set - Memory



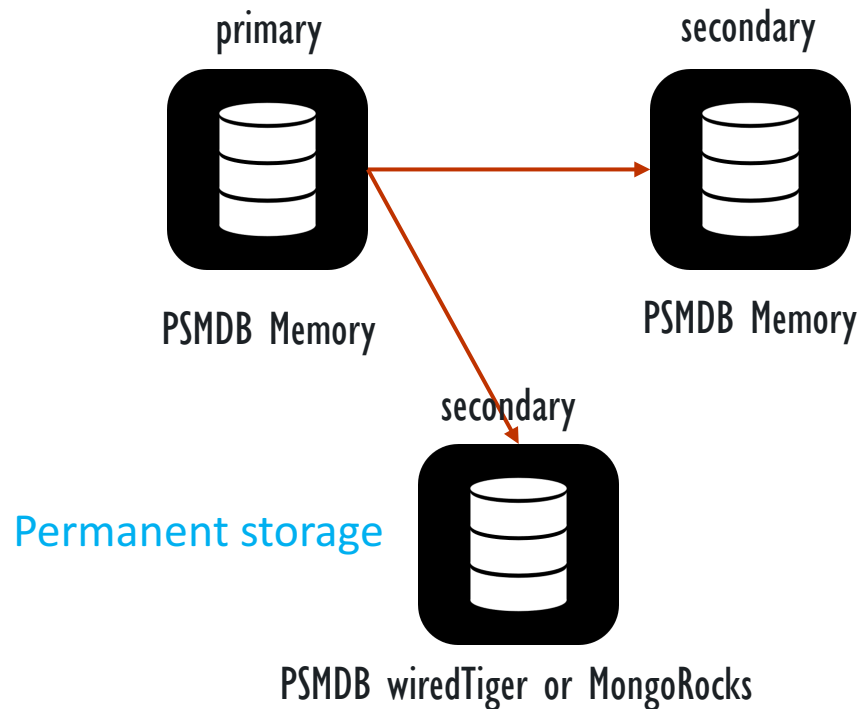
- For production
- Failover capability
- Data redundancy
- No Persistence
- Read scaling

Replica Set - config

`./mongod --storageEngine inMemory --dbpath /data/db --inMemorySizeGB 150 --replSet repSet1`



Replica Set – with Persistent Option

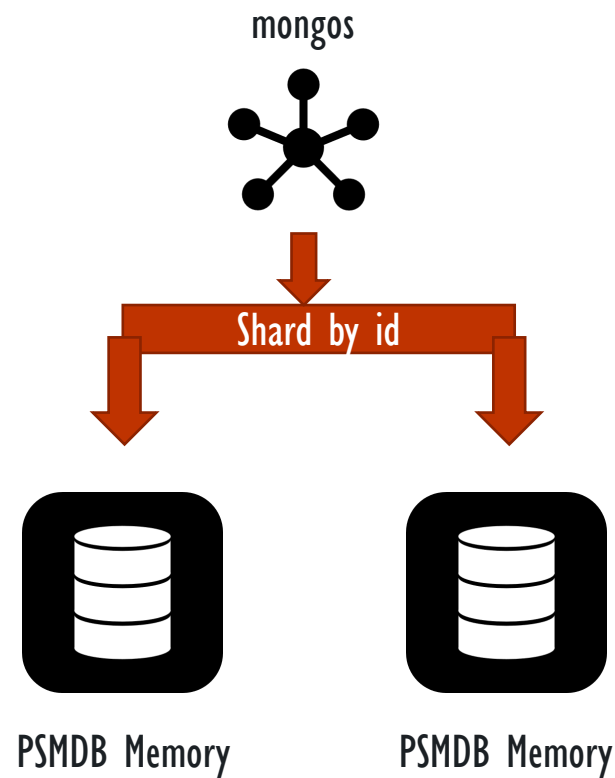


- For production
- Failover capability
- Data redundancy
- Data Persistence
- Read scaling
- Configure permanent member as a hidden member (i.e. `hidden: true` and `priority: 0`)

Replicate set with permanent option

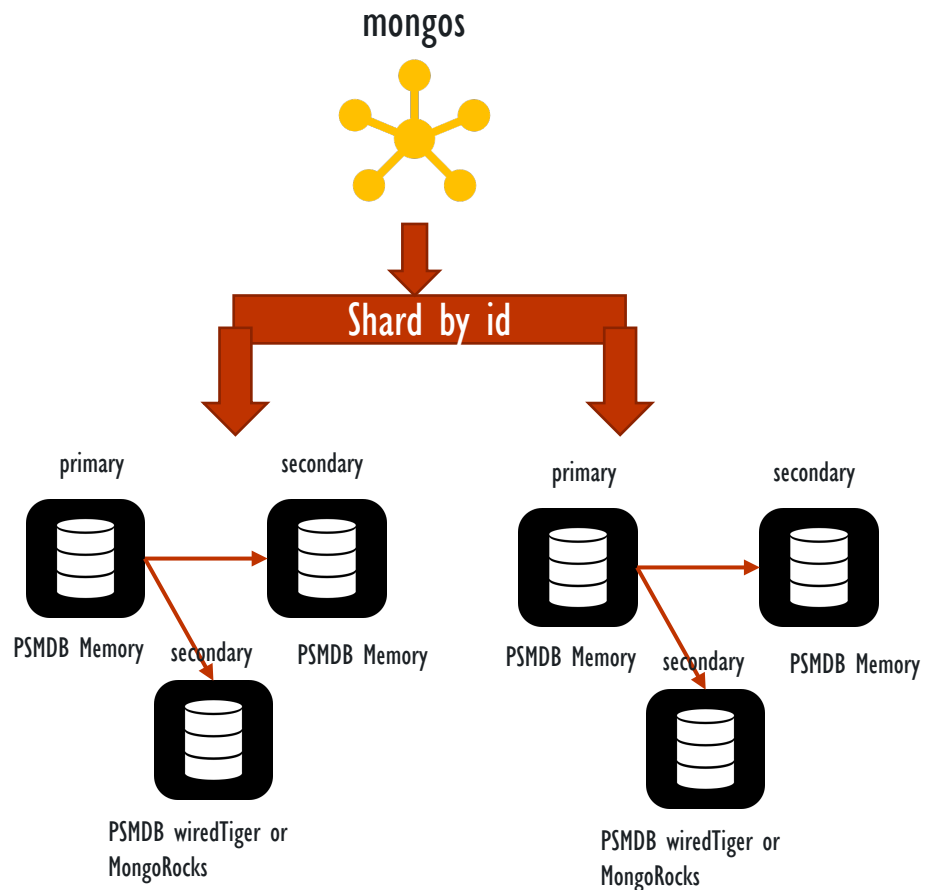
- Only the mongod instances running with the Memory storage engine can become the primary.
- Clients connect only to the Memory storage engine mongod instances.
- Even if both mongod instances running Memory storage engine crash and restart, they can sync from the member running permanent storage engine.
- The hidden mongod instance running with permanent storage persists the data to disk, including the user data, indexes, and replication configuration information.

Sharding – scaling for more memory



Sharding – scaling

- For production
- Failover capability
- Data redundancy
- Data Persistence
- **Write** scaling



PSMDB extras

PSMDB – secondary indexes

```
{  
  "_id" : "Vadim",  
  "visits" : 45,  
  "created" : ISODate("2016-01-07T15:46:32.085Z")  
}
```

```
> db.coll.ensureIndex({visits:1});  
> db.coll.find({visits : {$gte :40}});
```

- Data retrieval by document fields

PSMDB – TTL indexes

```
{  
  "_id" : "Vadim",  
  "visits" : 45,  
  "created" : ISODate("2016-01-07T15:46:32.085Z")  
}
```

```
> db.coll.createIndex({"created": 1},  
                      {expireAfterSeconds: 7200 });
```

- To remove data automatically

More features

- **Authentication**
 - Including External SASL Authentication
 - Role-Based Access Control
- **Audit logging**
- **Geospatial Indexes**
- **Text Search**

PSMDB Memory Engine - summary

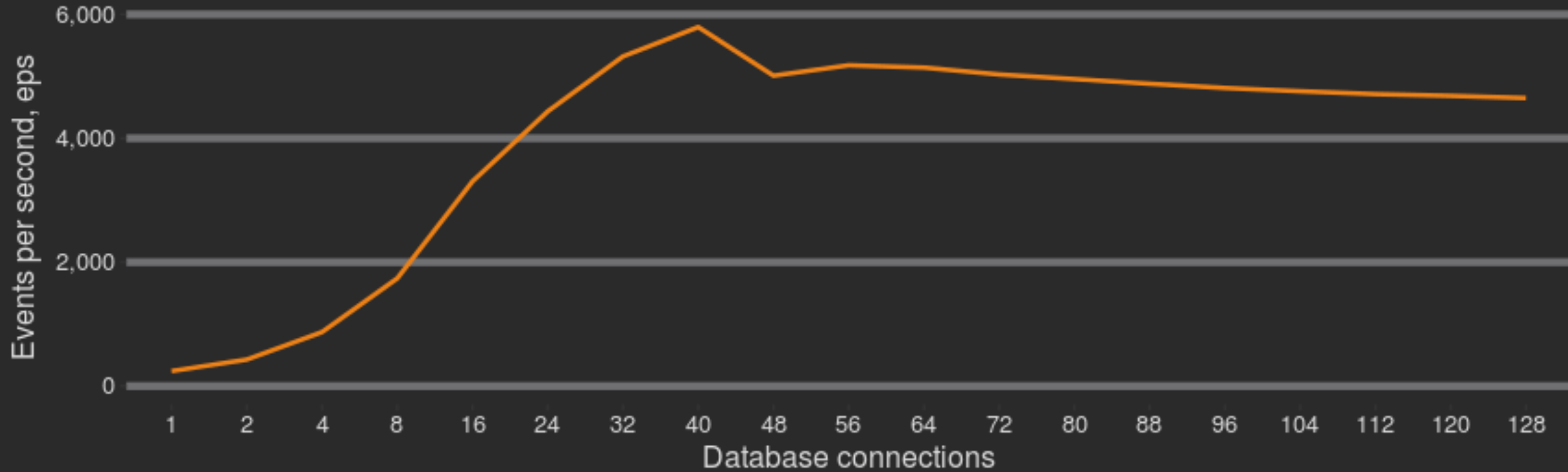
Store Type	Document Store
Failover / HA	Replica Set
Scaling	Sharding
Persistence	Extra Replica with wiredTiger or MongoRocks
CRUD	Insert, find, update, remove
Expire data	TTL
Concurrency	Multi-threading

Scalability – Multi Threading

- Intel(R) Xeon(R) CPU E5-2683 v3 @ 2.00GHz with 28 cores / 56 threads
- 256GB RAM
- Sysbench 1.0 with oltp_mongo.lua
 - Available at <https://github.com/Percona-Lab/sysbench-mongodb-lua>
- 100GB dataset
- User threads from 1 to 128
- Operation: OLTP event

Sysbench for MongoDB: dataset 32tables/12M rows/pareto

Throughput, events per second (higher is better)



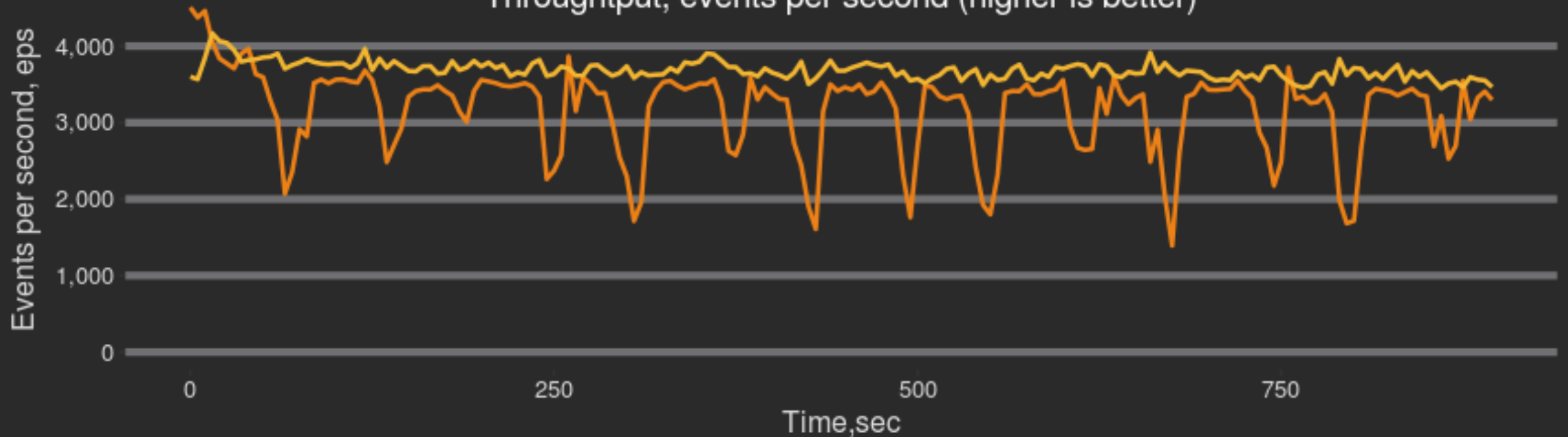
Percona Server for MongoDB 3.4.9: — InMemory

Stability – Multi Threading

- Intel(R) Xeon(R) CPU E5-2683 v3 @ 2.00GHz with 28 cores / 56 threads
- 256GB RAM
- Sysbench 1.0 with oltp_mongo.lua
 - Available at github.com/percona-labs
- 100GB dataset
- User threads 256
- Operation: OLTP event

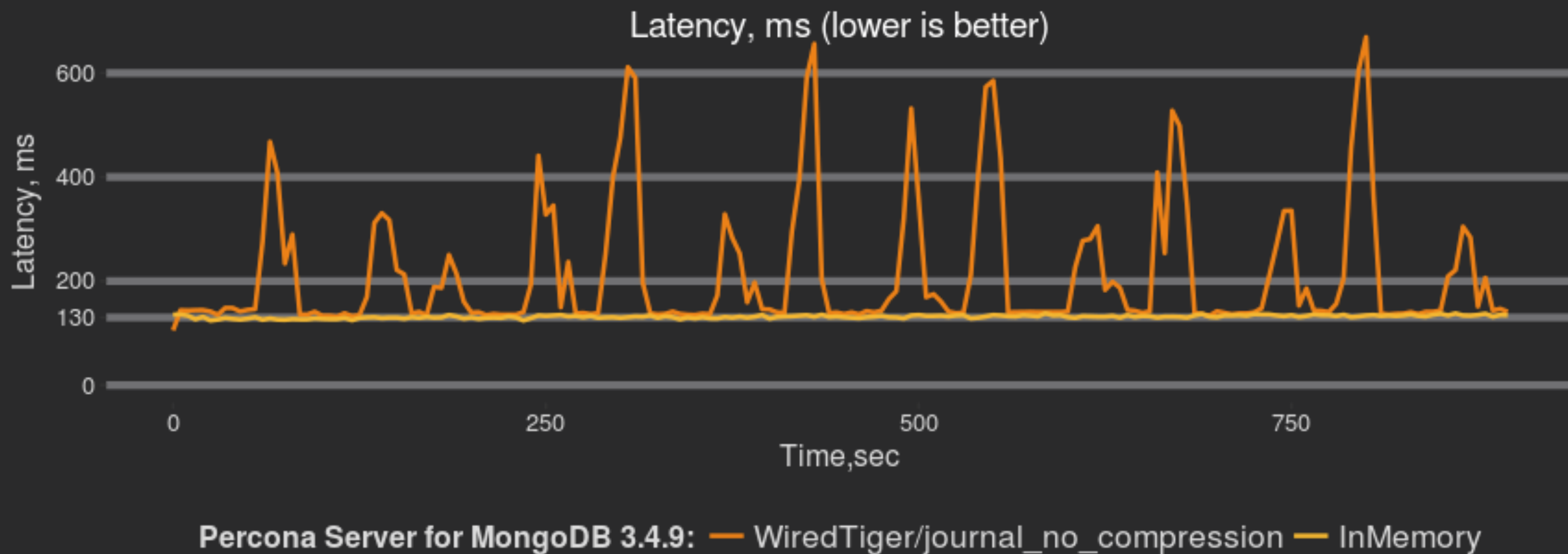
Sysbench for MongoDB: dataset 32tables/12M rows/pareto
Test run with 256 threads

Throughput, events per second (higher is better)



Percona Server for MongoDB 3.4.9: — WiredTiger/journal_no_compression — InMemory

Sysbench for MongoDB: dataset 32tables/12M rows/pareto
Test run with 256 threads



Predictable response time

130ms (with practically no variation)

256 user threads

Monitoring with Percona Monitoring and Management (PMM)

PMM

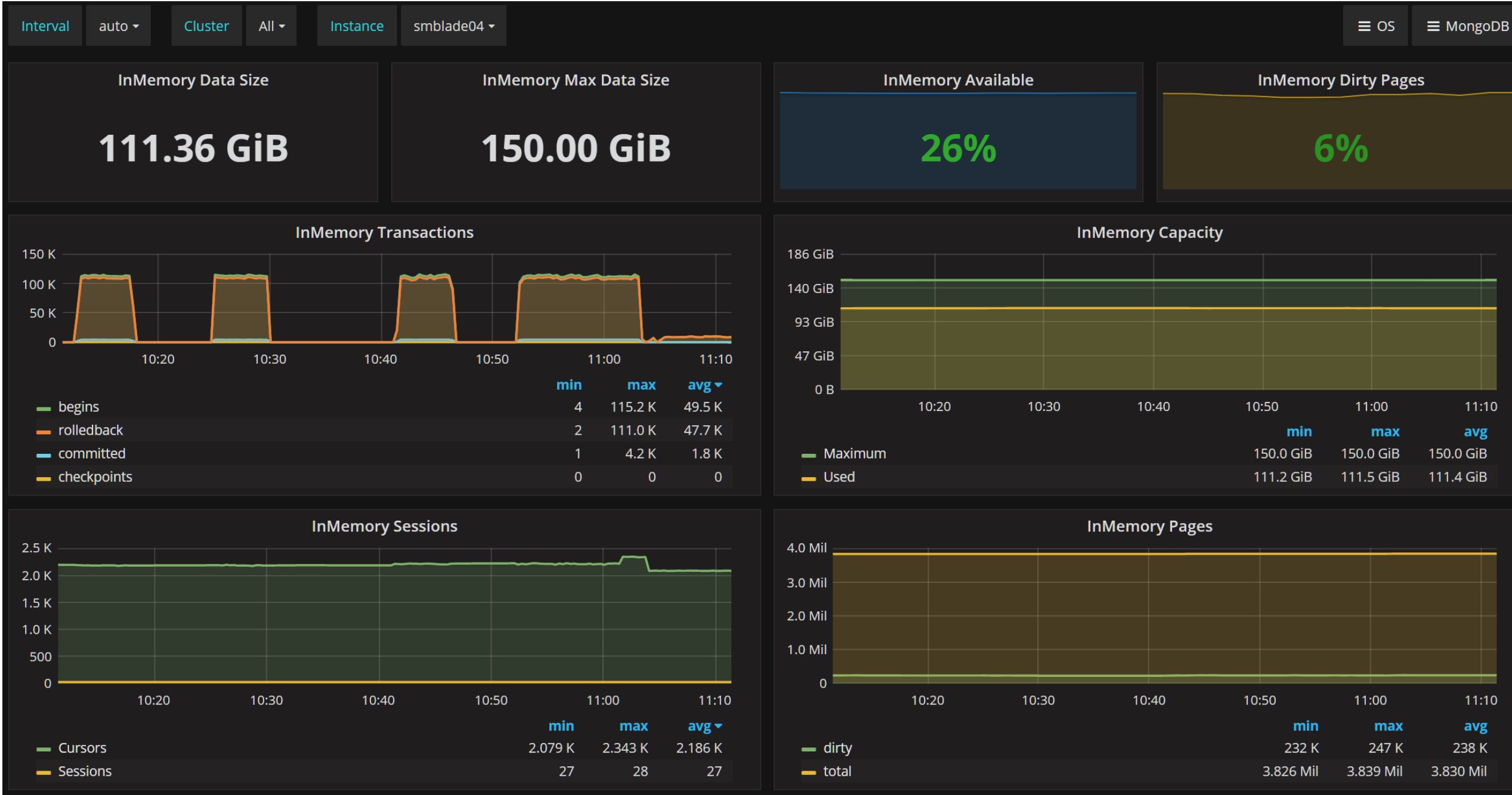
- New software from Percona
- Free and open-source platform for managing and monitoring MySQL[®] and MongoDB[®] performance.
- You can run PMM in your own environment
- Provides thorough time-based analysis for MySQL, MariaDB[®] and MongoDB servers to ensure that your data works as efficiently as possible.

PMM benefits

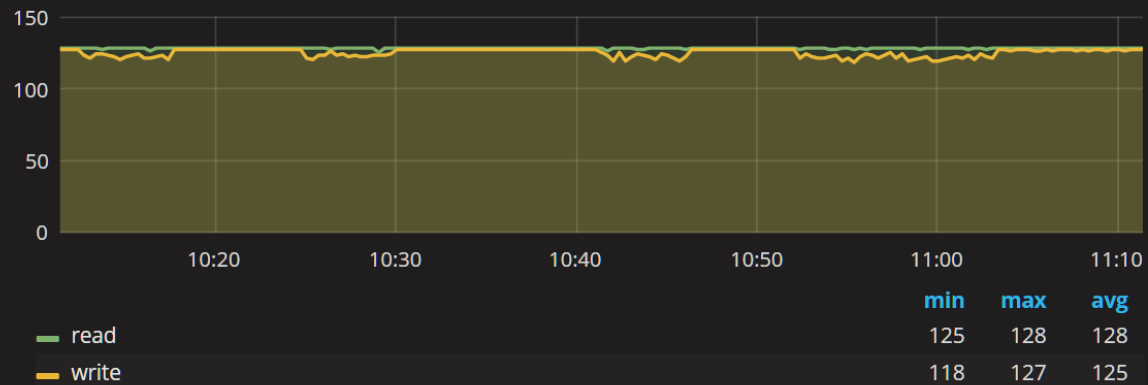
- Provides query and metric information that enables administrators to optimize database performance
- Displays current queries and highlights potential query issues to enable faster issue resolution
- Maps queries against metrics to help make informed decisions about crucial database resources: platform needs, system growth, team focus and the most important database activities

PMM features

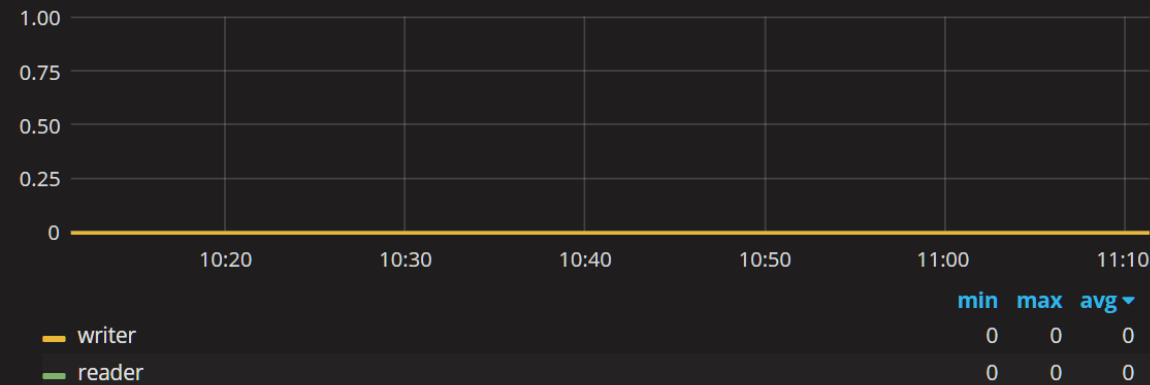
- Point-in-time visibility and historical trending of database performance
- Data from the MongoDB query profiler
- Specialized MongoDB dashboards for graphing MMAPv1, InMemory, MongoRocks
- A historical view of metrics that are critical to a database server
- Query metrics, including bytes sent, lock time, rows sent, and more
- Best-of-breed tools, including Grafana, Prometheus, and Consul, as well as Percona-developed query analytics, administration, API, agent and exporter components
- A single, easy to manage virtual appliance



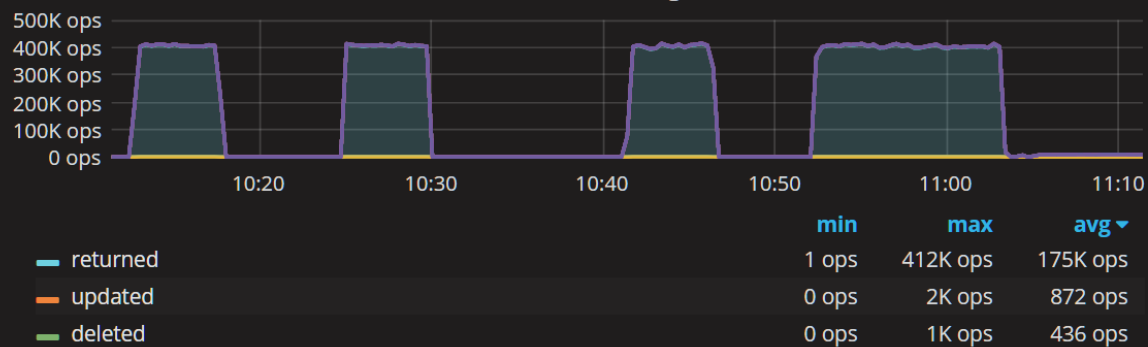
InMemory Concurrency Tickets



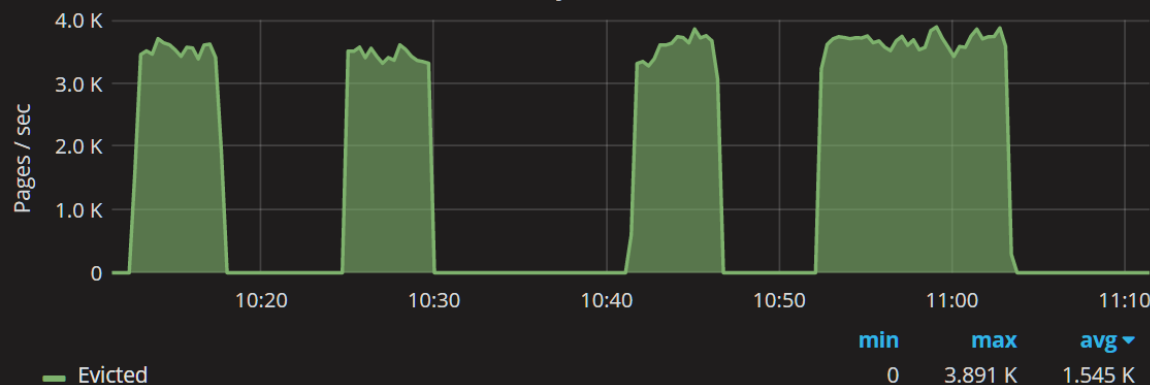
Queued Operations



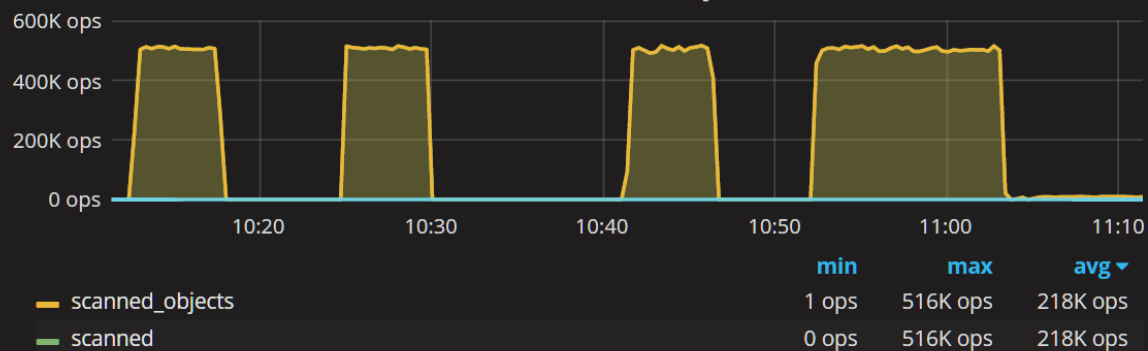
Document Changes



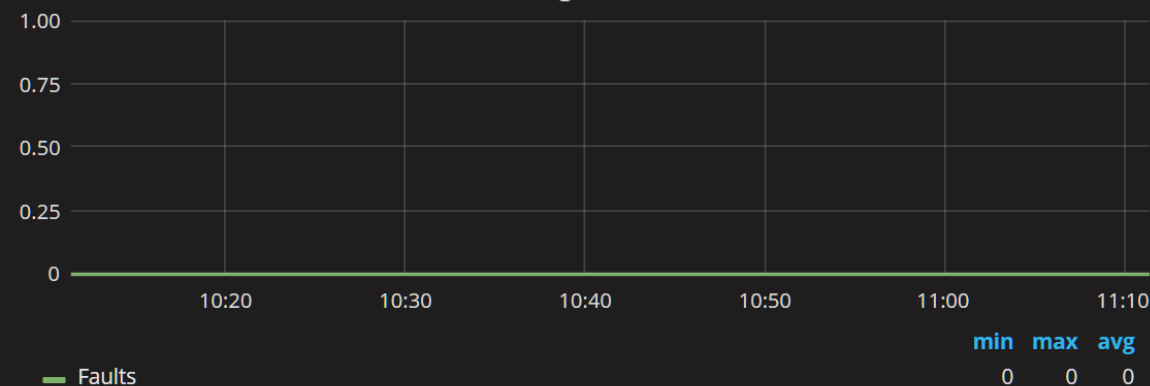
InMemory Cache Eviction



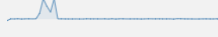
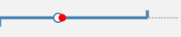
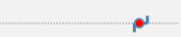
Scanned and Moved Objects



Page Faults



PMM – Query Analytics

Top 10 of 49 Queries by % Grand Total Time (%GTT)									
Search by Query Abstract, Fingerprint or ID 									
#	Query Abstract	ID	Load	Count	Latency				
	TOTAL		 7.08 (100.00%)	727.51 QPS	 2.62 m (100.00%)	 9.73 ms avg			
1	UPDATE sbtest	D3...	 5.09 (71.92%)	289.09 QPS	 1.04 m (39.74%)	 17.61 ms avg			
2	LOCK sbtest	0B...	 0.85 (11.95%)	2.85 QPS	 10.26 k (0.39%)	 296.69 ms avg			
3	SELECT sbtest	558...	 0.39 (5.46%)	226.49 QPS	 815.38 k (31.13%)	 1.71 ms avg			
4	SELECT sbtest	C9...	 0.22 (3.08%)	<0.01 QPS	 6.00 (0.00%)	 0:02:11 avg			
5	UPDATE sbtest	E96...	 0.10 (1.43%)	22.76 QPS	 81.95 k (3.13%)	 4.43 ms avg			
6	SELECT sbtest	737...	 0.09 (1.27%)	22.43 QPS	 80.74 k (3.08%)	 4.02 ms avg			
7	DELETE sbtest	EA...	 0.08 (1.16%)	22.60 QPS	 81.35 k (3.11%)	 3.64 ms avg			
8	SELECT sbtest	6EE...	 0.07 (0.97%)	22.56 QPS	 81.23 k (3.10%)	 3.06 ms avg			
9	SELECT sbtest	84...	 0.07 (0.97%)	22.57 QPS	 81.25 k (3.10%)	 3.05 ms avg			
10	SELECT sbtest	382...	 0.06 (0.88%)	22.42 QPS	 80.70 k (3.08%)	 2.79 ms avg			
⌵ Load next 10 queries ⌵									

Questions?
