Barking Mad

With DalmatinerDB

https://dalmatiner.io

Once upon a time...

Alfresco

Nagios

ZABBIX

graphite

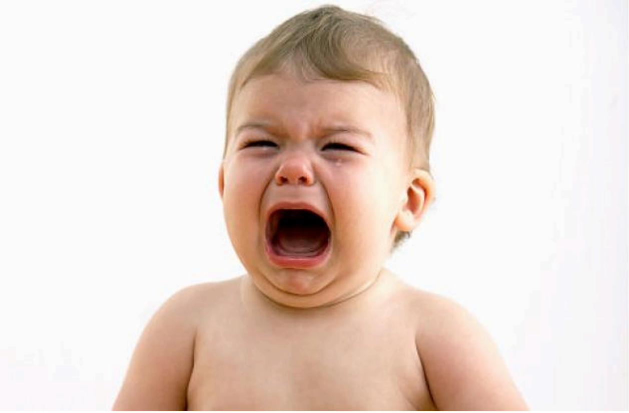
collectd
The system statistics collection daemon

sensu

StatsD

OPSVIEW

Kibana



End of 2013

Wealth
Startup
Happiness

Dataloop.IO

What is Dataloop?

Up / Down

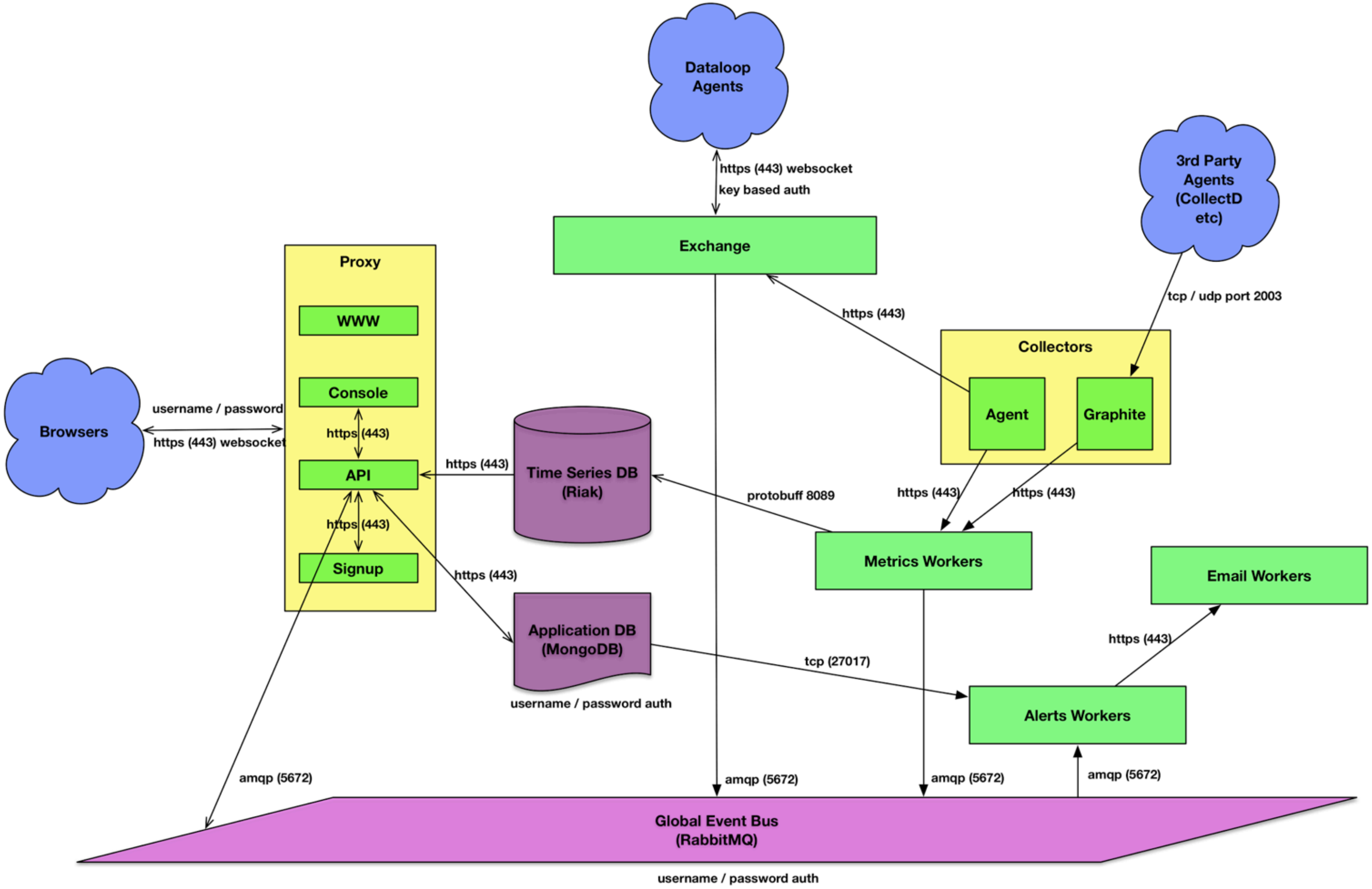
Performance

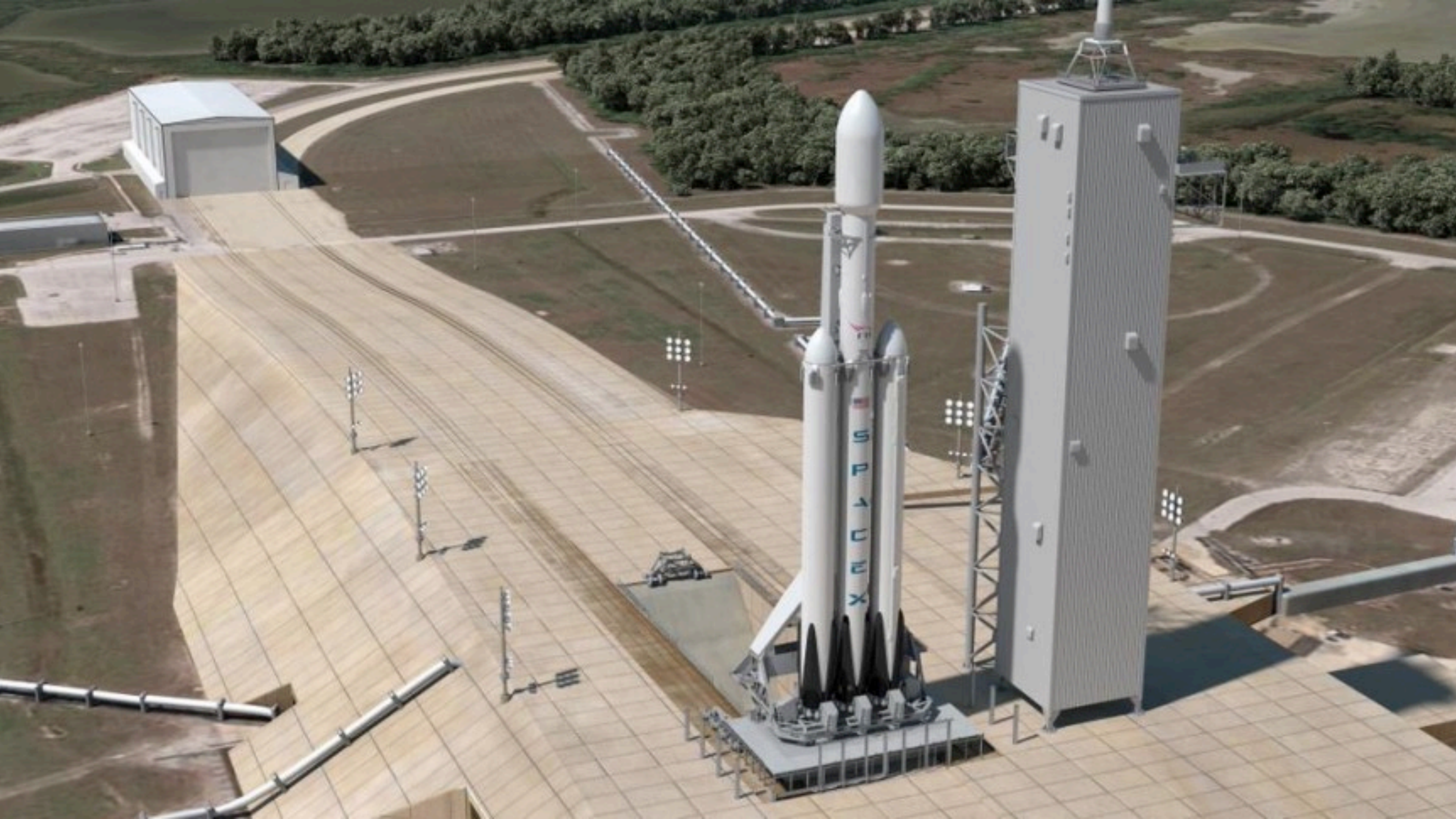
Alerts

Dev Env

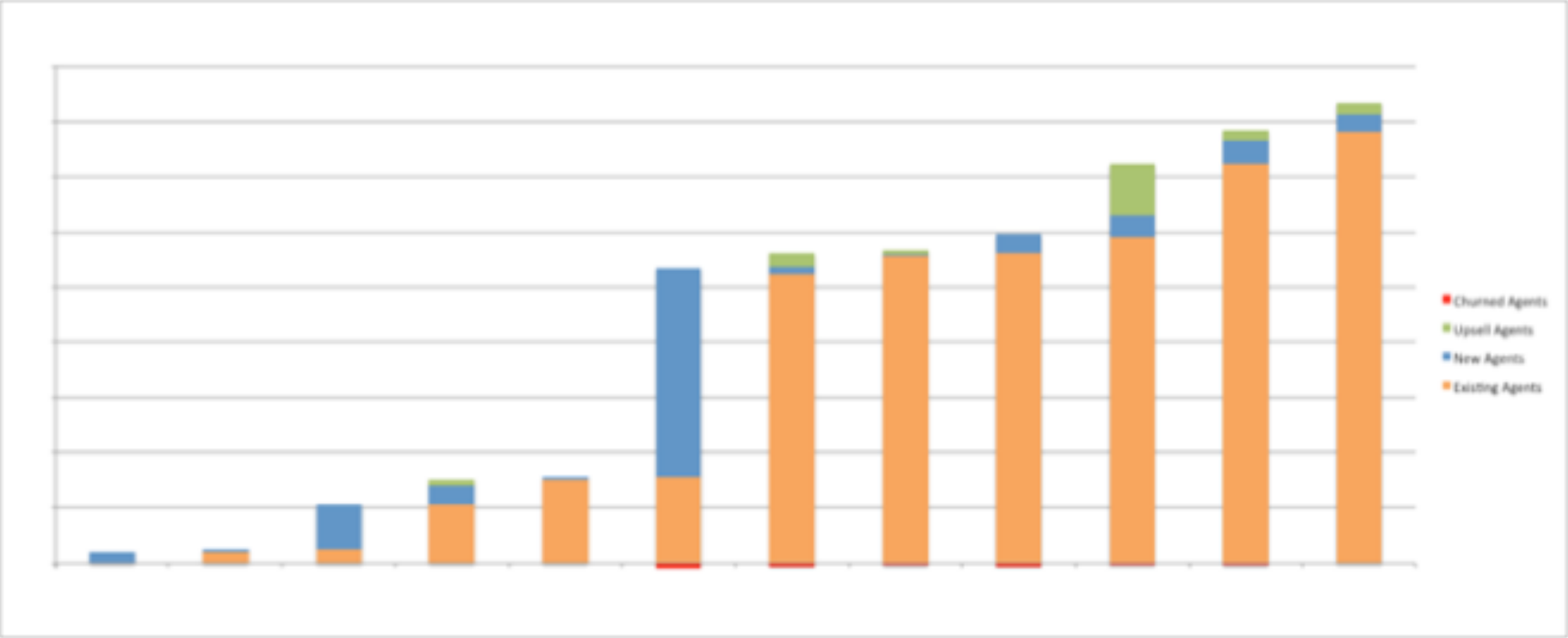
Enterprise Stuff

Dataloop.IO


Dataloop Agents
https (443) websocket
key based auth
3rd Party Agents (CollectD etc)
Exchange
tcp / udp port 2003
https (443)
Proxy
WWW
Collectors
Console
Agent
Graphite
Browsers
username / password
https (443) websocket
https (443)
API
https (443)
Time Series DB (Riak)
protobuff 8089
https (443)
https (443)
https (443)
Metrics Workers
Signup
https (443)
Email Workers
Application DB (MongoDB)
https (443)
tcp (27017)
username / password auth
Alerts Workers
amqp (5672)
amqp (5672)
amqp (5672)
amqp (5672)
Global Event Bus (RabbitMQ)
username / password auth

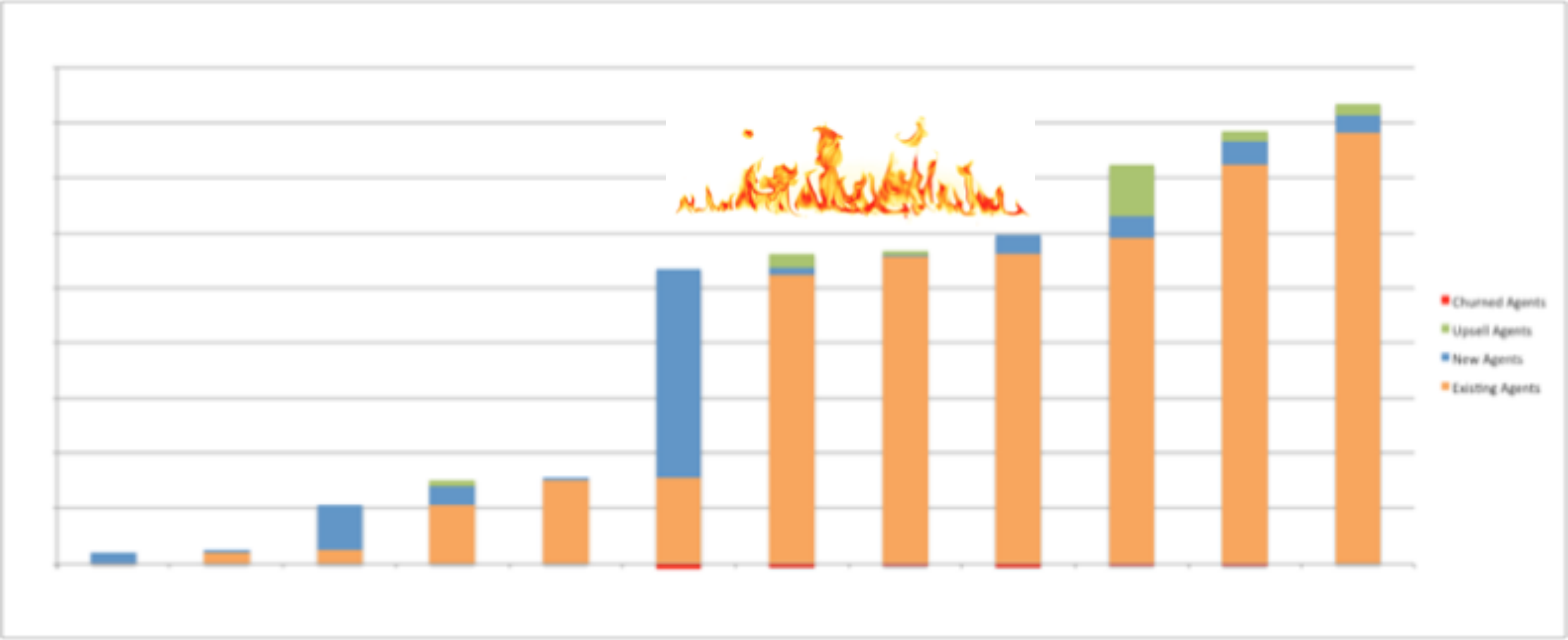

SPACEX

Dataloop Agents by Month

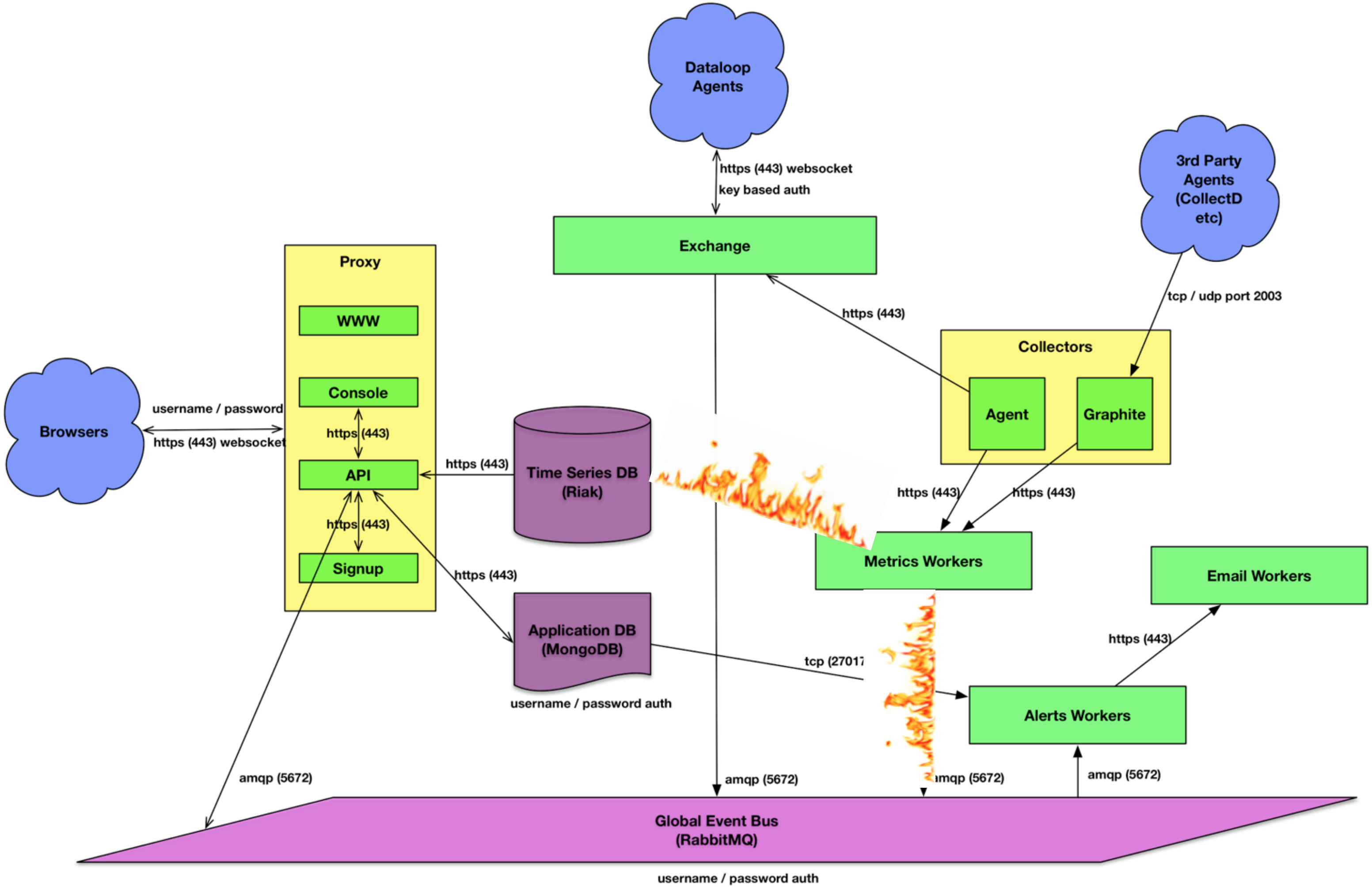


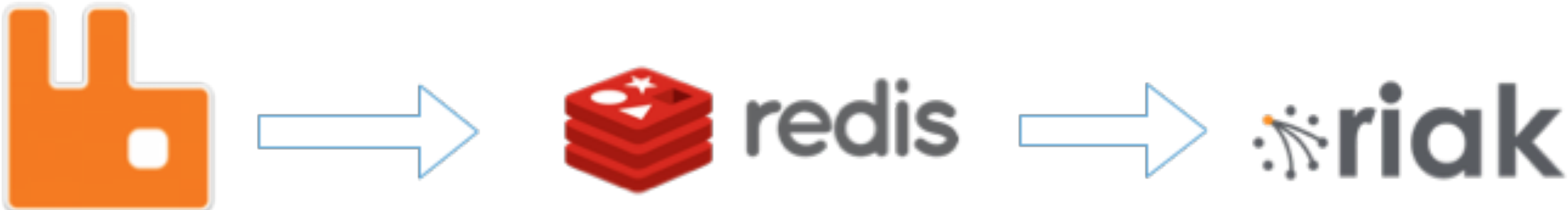
2015

Dataloop Agents by Month



2015

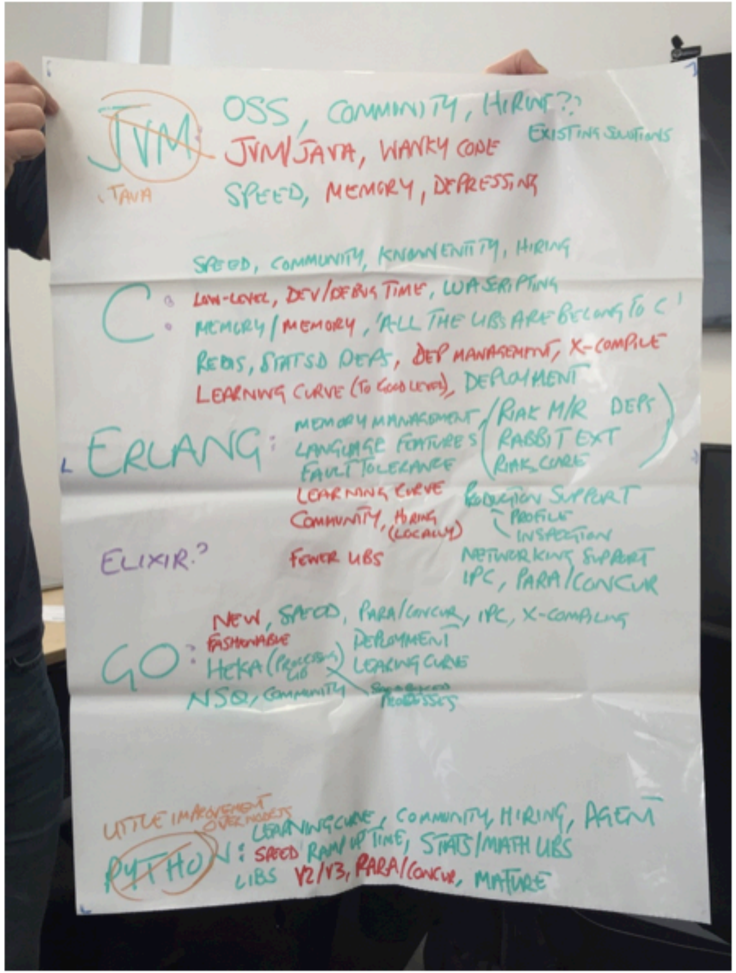

Dataloop Agents
https (443) websocket
key based auth
Exchange
3rd Party Agents (CollectD etc)
tcp / udp port 2003
https (443)
Proxy
WWW
Console
https (443)
API
https (443)
Signup
Browsers
username / password
https (443) websocket
Collectors
Agent
Graphite
https (443)
Time Series DB (Riak)
https (443)
https (443)
https (443)
Metrics Workers
Email Workers
https (443)
Application DB (MongoDB)
username / password auth
tcp (2701
Alerts Workers
amqp (5672)
amqp (5672)
amqp (5672)
amqp (5672)
Global Event Bus (RabbitMQ)
username / password auth



metric worker

rollup worker

- NodeJS metrics workers not scaling
- Memory management was an issue
- Needed big caches to reduce database load
- GC cycles too long
- 8 x single processes on an 8 core server

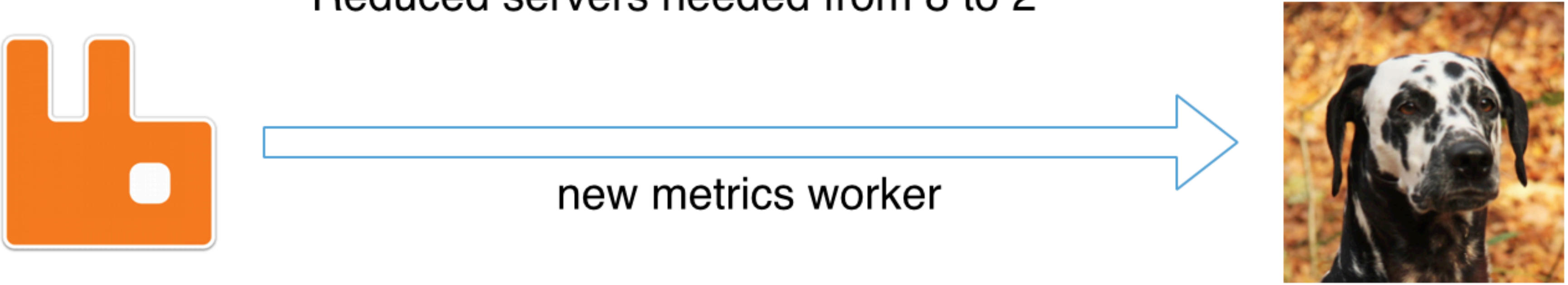

~~JVM~~
JAVA
OSS, COMMUNITY, HIRING??
EXISTING SOLUTIONS
JVM/JAVA, WANKY CODE
SPEED, MEMORY, DEPRESSING
SPEED, COMMUNITY, KNOWN ENTITY, HIRING
C:
LOW-LEVEL, DEV/DEBUG TIME, LUA SCRIPTING
MEMORY/MEMORY, 'ALL THE LIBS ARE BELONG TO C'
REDIS, STATSD DEPS, DEP MANAGEMENT, X-COMPILE
LEARNING CURVE (TO GOOD LEVEL), DEPLOYMENT
ERLANG:
MEMORY MANAGEMENT
LANGUAGE FEATURES
FAULT TOLERANCE
RIAK M/R DEPS
RABBIT EXT
RIAK CORE
LEARNING CURVE
PRODUCTION SUPPORT
COMMUNITY, HIRING (LOCALLY)
PROFILE
INSPECTION
ELIXIR?
FEWER LIBS
NETWORKING SUPPORT
IPC, PARA/CONCUR
GO:
NEW, SPEED, PARA/CONCUR, IPC, X-COMPILING
FASHIONABLE
DEPLOYMENT
HEKA (PROCESSING LIB)
LEARNING CURVE
NSQ, COMMUNITY
SEPARATED PROCESSES
LITTLE IMPROVEMENT OVER NODEJS
~~PYTHON~~:
LEARNING CURVE, COMMUNITY, HIRING, AGENT
SPEED RAMP UP TIME, STATS/MATH LIBS
LIBS V2/V3, PARA/CONCUR, MATURE

- Decided on Erlang
- Memory management
- Fault tolerance
- Good libraries for Rabbit and Riak
- Live code tracing

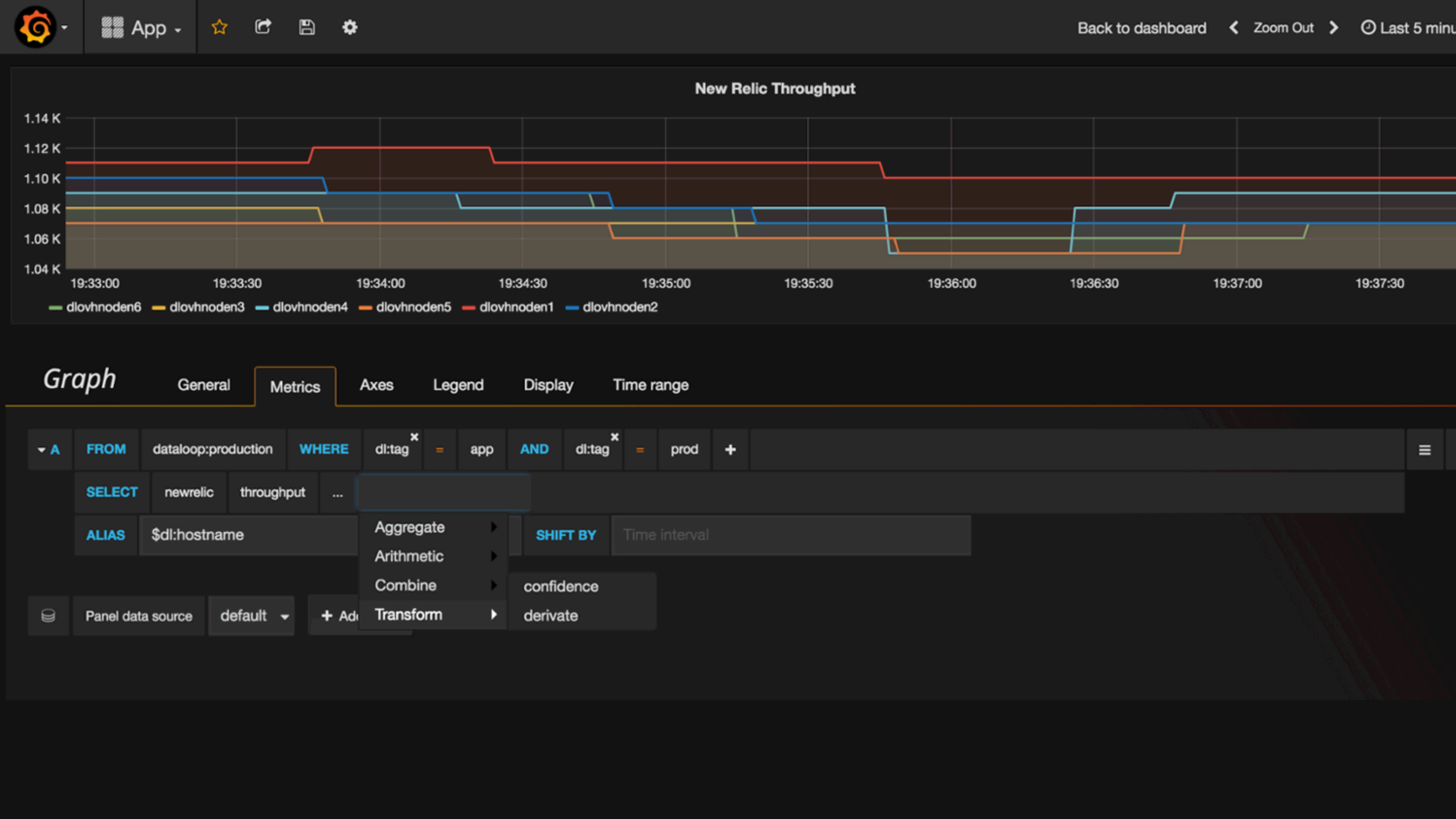
- Approximately 6 weeks from no Erlang experience to working version
- No more crashes
- Reduced servers needed from 16 to 8

- Open Source Time-Series DB
- Written in Erlang
- Based on Riak-Core and uses ZFS
- Optimised for write throughput
- Needed for developer analytics features
- https://dalmatiner.io/

- Worked with Erlang solutions
- Cross trained team (Dave and Tomasz)
- Removed the Redis
- Reduced servers needed from 8 to 2



new metrics worker



A FROM dataloop:production WHERE dl:tag = app AND dl:tag = prod +

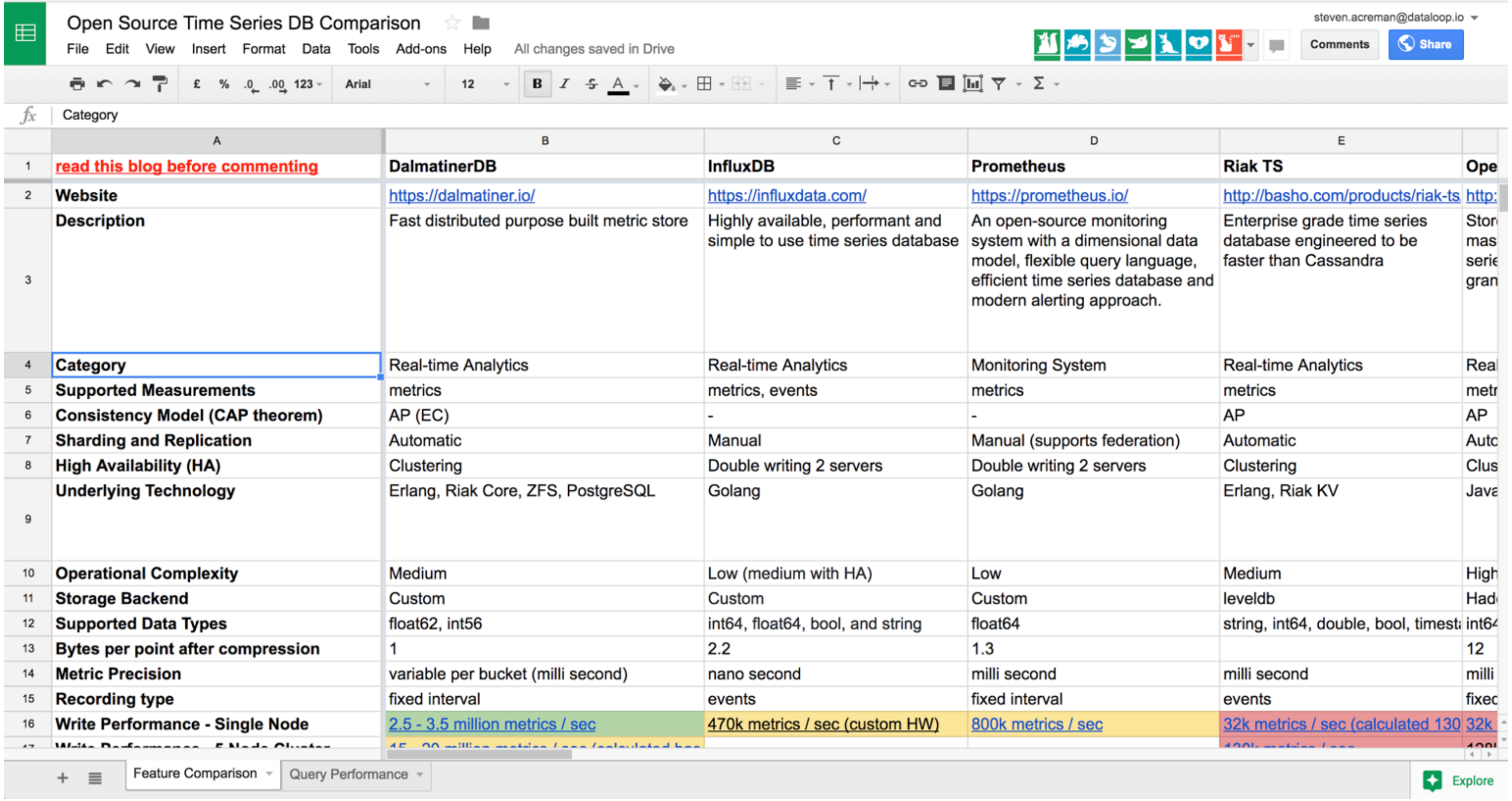
SELECT newrelic throughput ...

ALIAS $dl:hostname

SHIFT BY Time interval

Panel data source default

But did you try..

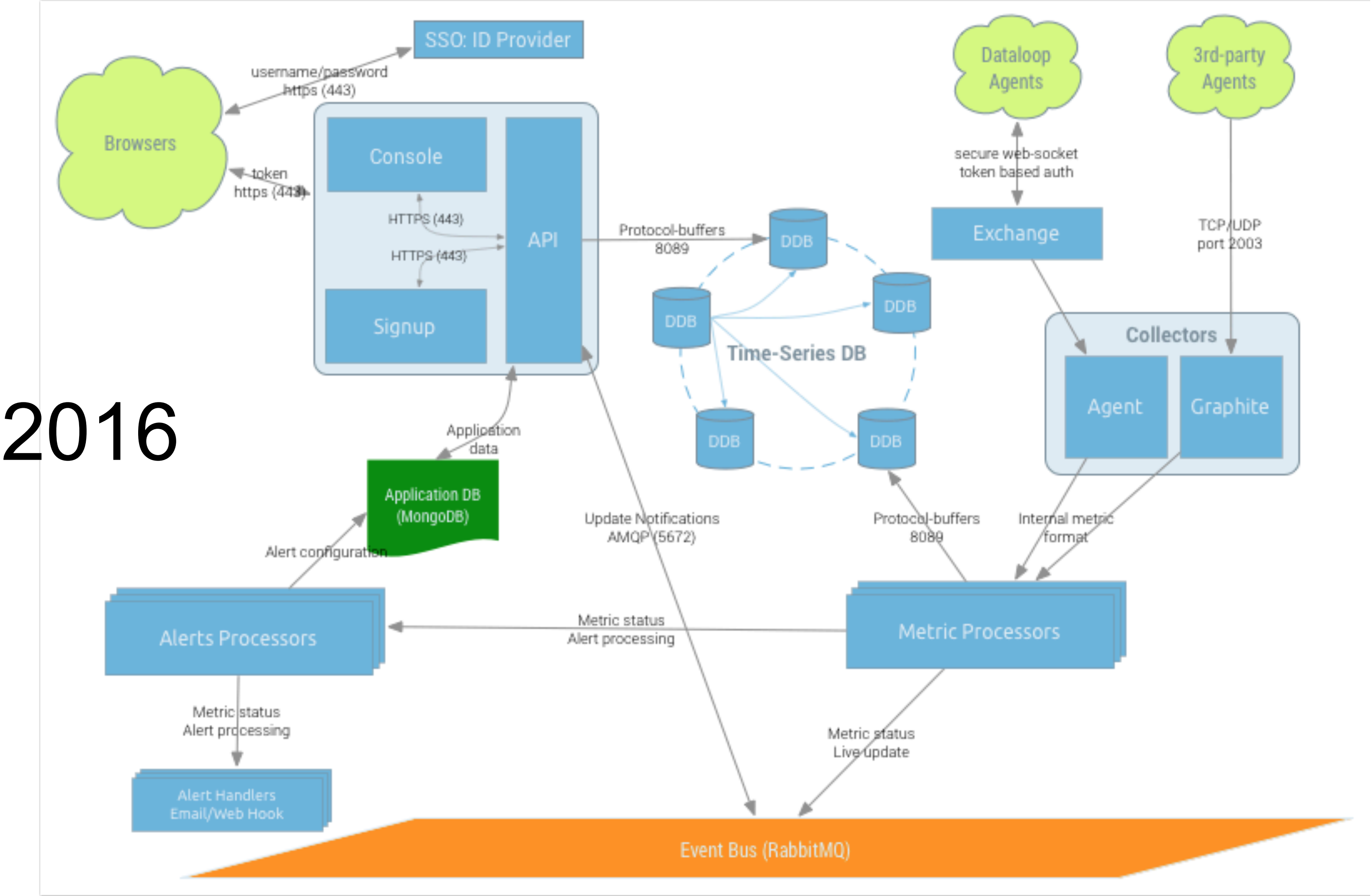


Open Source Time Series DB Comparison

fx Category

	A	B	C	D	E	
1	read this blog before commenting	**DalmatinerDB**	**InfluxDB**	**Prometheus**	**Riak TS**	**Ope**
2	**Website**	https://dalmatiner.io/	https://influxdata.com/	https://prometheus.io/	http://basho.com/products/riak-ts	http:
3	**Description**	Fast distributed purpose built metric store	Highly available, performant and simple to use time series database	An open-source monitoring system with a dimensional data model, flexible query language, efficient time series database and modern alerting approach.	Enterprise grade time series database engineered to be faster than Cassandra	Stor mas serie gran
4	**Category**	Real-time Analytics	Real-time Analytics	Monitoring System	Real-time Analytics	Real
5	**Supported Measurements**	metrics	metrics, events	metrics	metrics	metr
6	**Consistency Model (CAP theorem)**	AP (EC)	-	-	AP	AP
7	**Sharding and Replication**	Automatic	Manual	Manual (supports federation)	Automatic	Autc
8	**High Availability (HA)**	Clustering	Double writing 2 servers	Double writing 2 servers	Clustering	Clus
9	**Underlying Technology**	Erlang, Riak Core, ZFS, PostgreSQL	Golang	Golang	Erlang, Riak KV	Java
10	**Operational Complexity**	Medium	Low (medium with HA)	Low	Medium	High
11	**Storage Backend**	Custom	Custom	Custom	leveldb	Had
12	**Supported Data Types**	float62, int56	int64, float64, bool, and string	float64	string, int64, double, bool, timest	int64
13	**Bytes per point after compression**	1	2.2	1.3		12
14	**Metric Precision**	variable per bucket (milli second)	nano second	milli second	milli second	milli
15	**Recording type**	fixed interval	events	fixed interval	events	fixed
16	**Write Performance - Single Node**	2.5 - 3.5 million metrics / sec	470k metrics / sec (custom HW)	800k metrics / sec	32k metrics / sec (calculated 130	32k
17	**Write Performance - 5 Node Cluster**	15 - 20 million metrics / sec (calculated bas			130k metrics / sec	128k

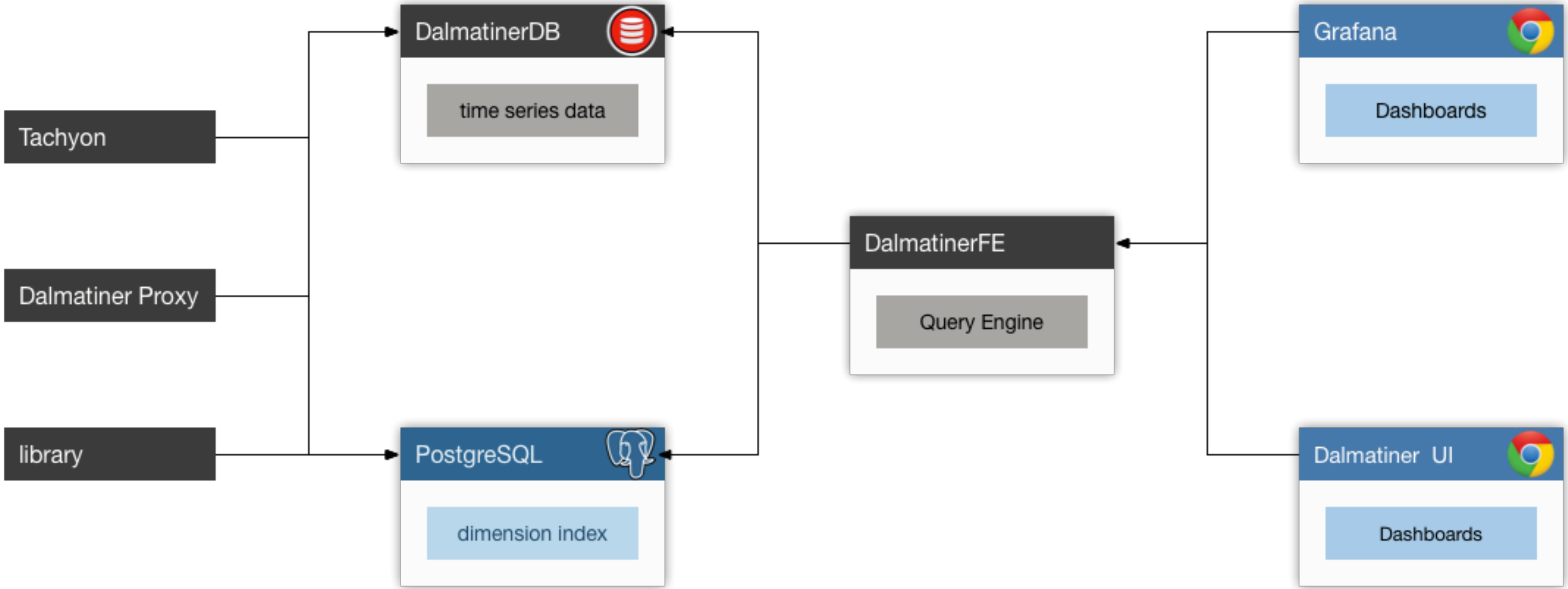
https://blog.dataloop.io/time-series-database-benchmarks


SSO: ID Provider
Browsers
username/password
https (443)
token
https (443)
Console
HTTPS (443)
HTTPS (443)
Signup
API
Protocol-buffers
8089
DDB
DDB
DDB
DDB
DDB
DDB
Time-Series DB
Dataloop
Agents
3rd-party
Agents
secure web-socket
token based auth
Exchange
TCP/UDP
port 2003
Collectors
Agent
Graphite
Application
data
Application DB
(MongoDB)
Alert configuration
Update Notifications
AMQP (5672)
Protocol-buffers
8089
Internal metric
format
Alerts Processors
Metric status
Alert processing
Metric Processors
Metric status
Alert processing
Alert Handlers
Email/Web Hook
Metric status
Live update
Event Bus (RabbitMQ)

2016

PS. Dataloop is hiring Erlang developers!

More about DalmatinerDB



Tachyon

Dalmatiner Proxy

library

DalmatinerDB

time series data

PostgreSQL

dimension index

DalmatinerFE

Query Engine

Grafana

Dashboards

Dalmatiner UI

Dashboards

Story time

The language that isn’t performant

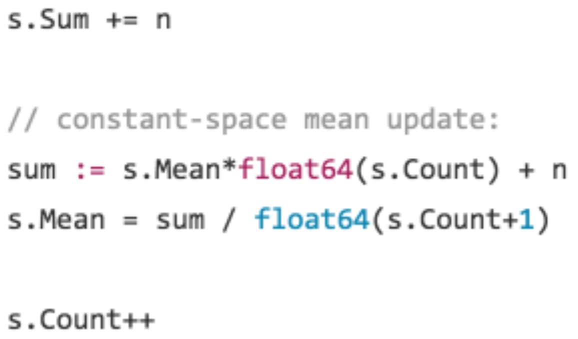
A long long time ago (6.31152E+07 seconds)

Monitoring a cloud

Finding a solution blew up

That (other) crazy person at EUC who doesn’t know it’s his fault

Algorithm beats bare s



```
s.Sum += n

// constant-space mean update:
sum := s.Mean*float64(s.Count) + n
s.Mean = sum / float64(s.Count+1)

s.Count++
```

Reinventing the wheel

Without reinventing the wheel

ZFS

- Compression
- Checksumming
- Snapshots

riak_core

- Distribution
- Cluster management
- Scaling

Postgres

- Dimensions
- Relational data
- Fast lookups
- Complex queries

Data layout (shiny new feature)

- Fully positionally indexed
- Very compressible - yay zfs!
- 64 bit per data point stored at ~1 b
- As simple as it gets

Query Engine (shiny new feature)

- Streaming query engine
- Typed SQL like function based language
- Data crunching done in C - oh my!

Serverless infrastructure my ass

Serverless infrastructure, my assumption is we are talking about informed decisions regarding state and its location

Stateless Components

- Frontend / Query Engine
- Proxy

Stateful Components

- Postgres - Metric Metadata
- DalmatinerDB - Metric Data

Combining Stateful and Stateless

- Minimal highly stable API between them
- Very modular
- Important: difference between internal and external API
- Features can be implemented in the parts when they matter
- Reduces downtime and maintenance requirements
- Fast iterations w/o compromising data
- One change that required updating two components at the same time in the past two years

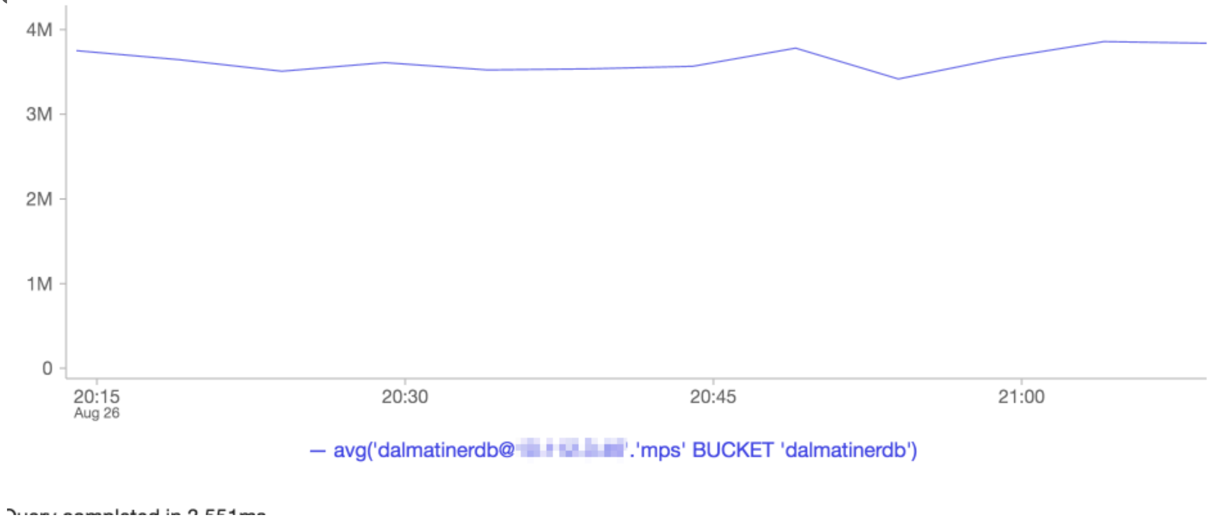
Showing off

Write Performance

- 16 core vCPU
- 110GB RAM
- 10.000G Disk

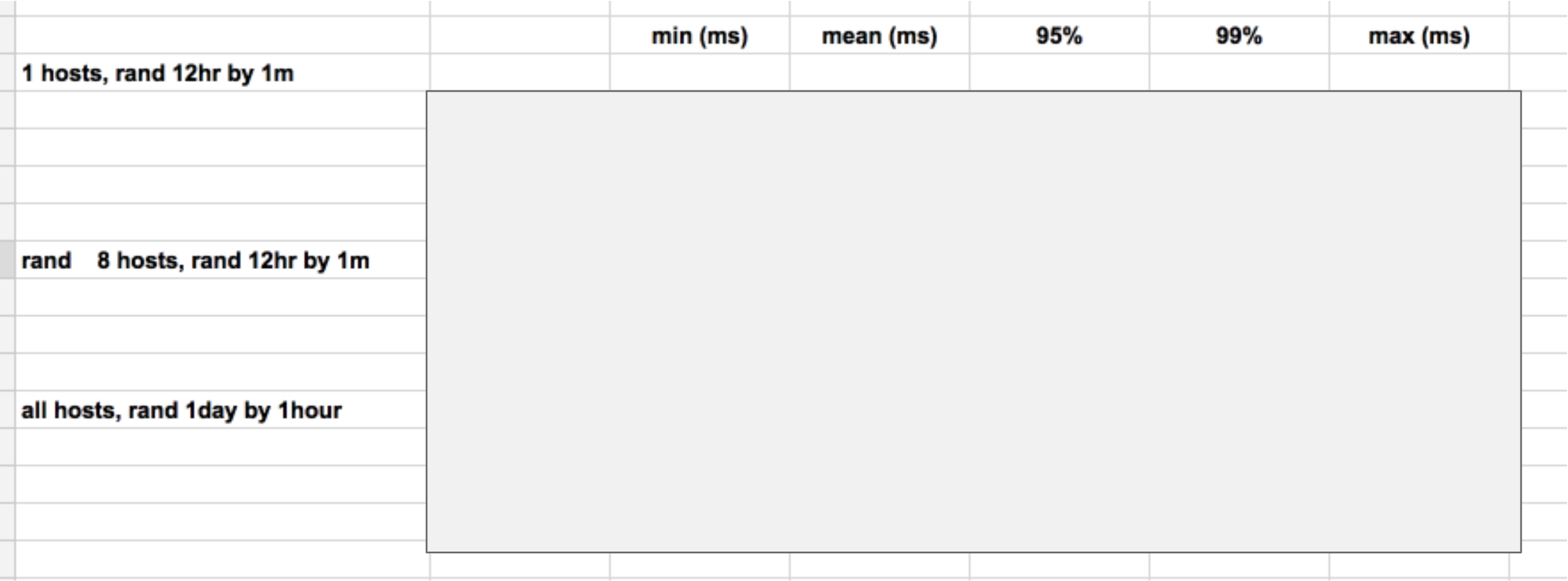
Write Performance

- 2.5-3.5 Million metrics ingested per second
- Thanks to riak_core architecture scales near linear
- Most (



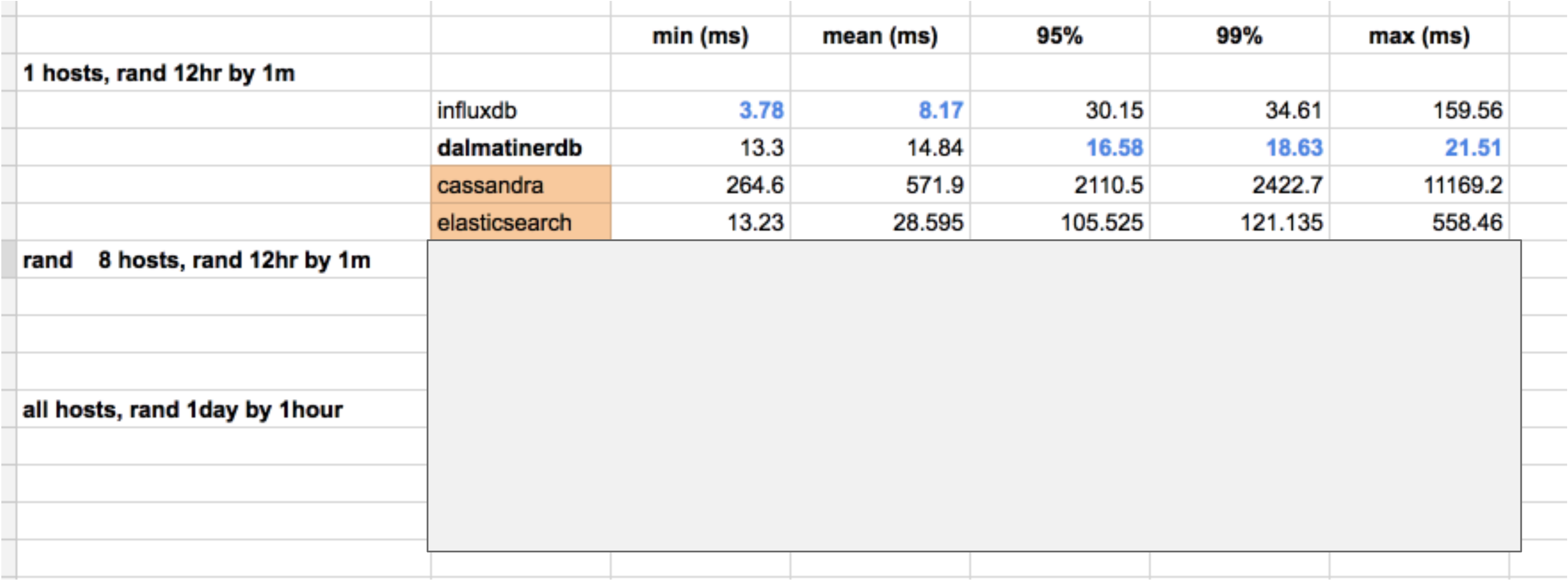
Query completed in 2.551ms

Query Performance



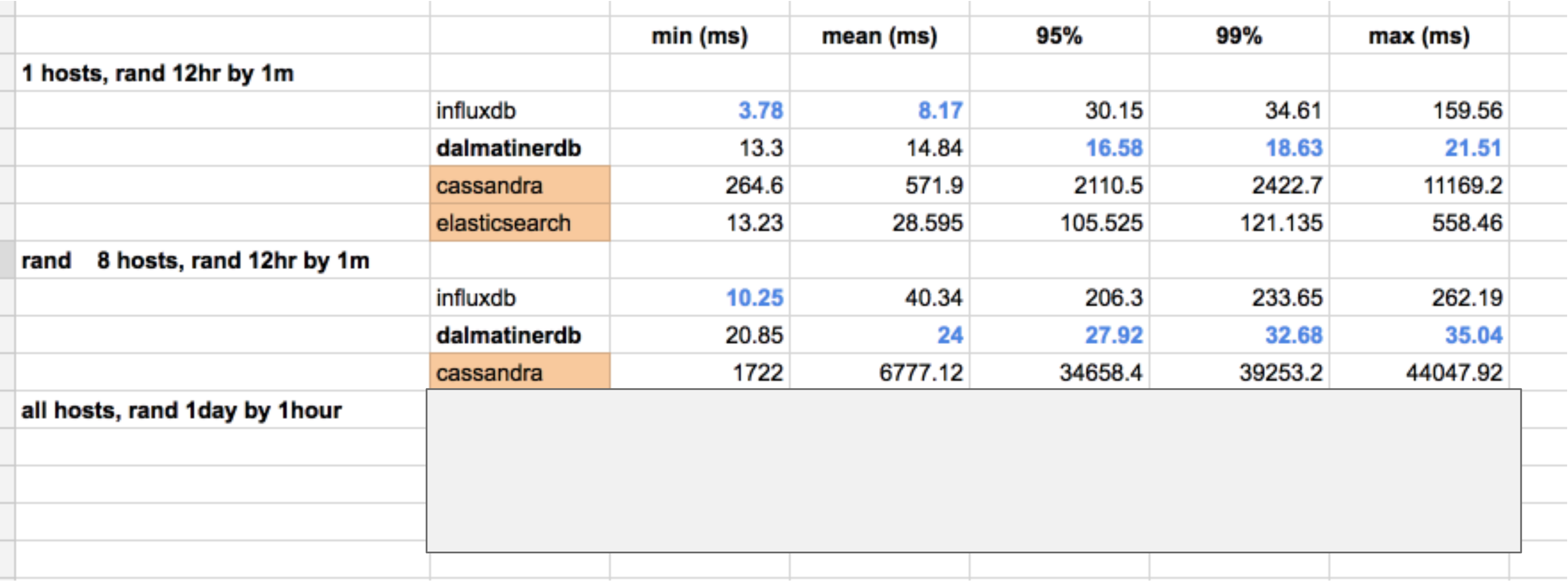
		min (ms)	mean (ms)	95%	99%	max (ms)
1 hosts, rand 12hr by 1m						
rand 8 hosts, rand 12hr by 1m						
all hosts, rand 1day by 1hour						

Query Performance



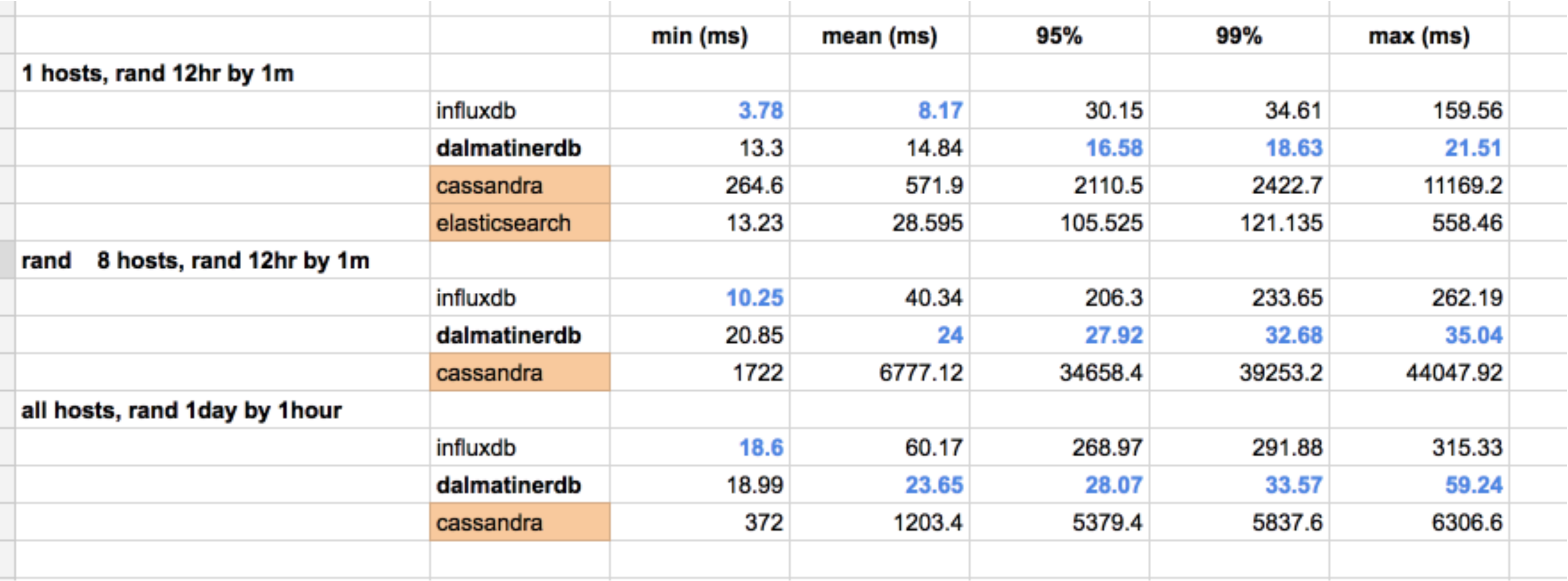
		min (ms)	mean (ms)	95%	99%	max (ms)
1 hosts, rand 12hr by 1m						
	influxdb	3.78	8.17	30.15	34.61	159.56
	dalmatinerdb	13.3	14.84	16.58	18.63	21.51
	cassandra	264.6	571.9	2110.5	2422.7	11169.2
	elasticsearch	13.23	28.595	105.525	121.135	558.46
rand 8 hosts, rand 12hr by 1m						
all hosts, rand 1day by 1hour						

Query Performance



		min (ms)	mean (ms)	95%	99%	max (ms)
1 hosts, rand 12hr by 1m						
	influxdb	**3.78**	**8.17**	30.15	34.61	159.56
	dalmatinerdb	13.3	14.84	**16.58**	**18.63**	**21.51**
	cassandra	264.6	571.9	2110.5	2422.7	11169.2
	elasticsearch	13.23	28.595	105.525	121.135	558.46
rand 8 hosts, rand 12hr by 1m						
	influxdb	**10.25**	40.34	206.3	233.65	262.19
	dalmatinerdb	20.85	**24**	**27.92**	**32.68**	**35.04**
	cassandra	1722	6777.12	34658.4	39253.2	44047.92
all hosts, rand 1day by 1hour						

Query Performance



		min (ms)	mean (ms)	95%	99%	max (ms)
1 hosts, rand 12hr by 1m						
	influxdb	**3.78**	**8.17**	30.15	34.61	159.56
	dalmatinerdb	13.3	14.84	**16.58**	**18.63**	**21.51**
	cassandra	264.6	571.9	2110.5	2422.7	11169.2
	elasticsearch	13.23	28.595	105.525	121.135	558.46
rand 8 hosts, rand 12hr by 1m						
	influxdb	**10.25**	40.34	206.3	233.65	262.19
	dalmatinerdb	20.85	**24**	**27.92**	**32.68**	**35.04**
	cassandra	1722	6777.12	34658.4	39253.2	44047.92
all hosts, rand 1day by 1hour						
	influxdb	**18.6**	60.17	268.97	291.88	315.33
	dalmatinerdb	18.99	**23.65**	**28.07**	**33.57**	**59.24**
	cassandra	372	1203.4	5379.4	5837.6	6306.6

The End!

(please try DalmatinerDB)
..and Project Fifo
and Dataloop.IO

Q&A