TIG

Modern Monitoring Platform FOR FREE!

Joyful Craftsmen

Business Intelligence Crafted to Joy

Agenda

- Brief introduction into components
- DEMO

*Bonus: InfluxDB insights**

About me

- SQL Server Support Engineer @Joyful Craftsmen
- Former SQL Server Lead DBA from ČS, a.s.
- Czech PASS Leader & SQL Saturday co-organizer

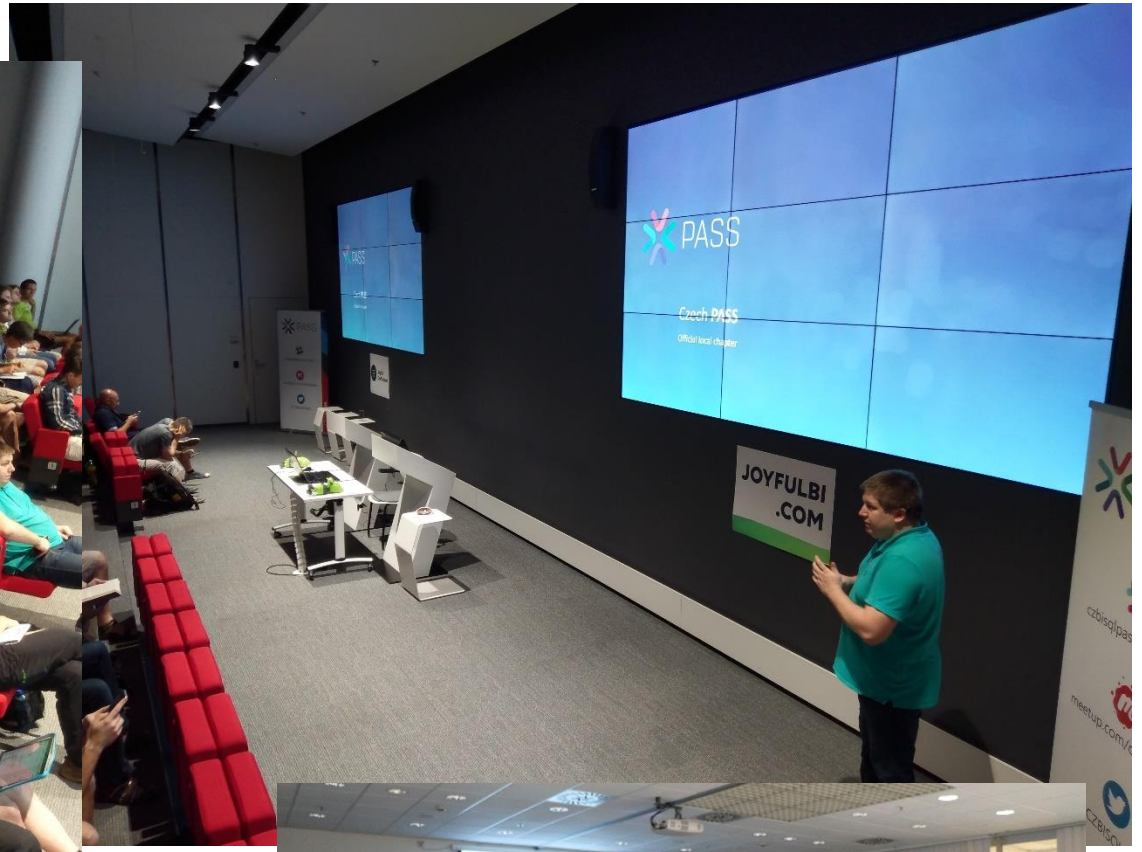
@malekpav

pavel.malek@joyfulcraftsmen.com

https://joyfulcraftsmen.com/

Czech PASS

SQL Saturday Prague #889



PASS

#JOYFULDATA

Components

Database

Monitoring Agent

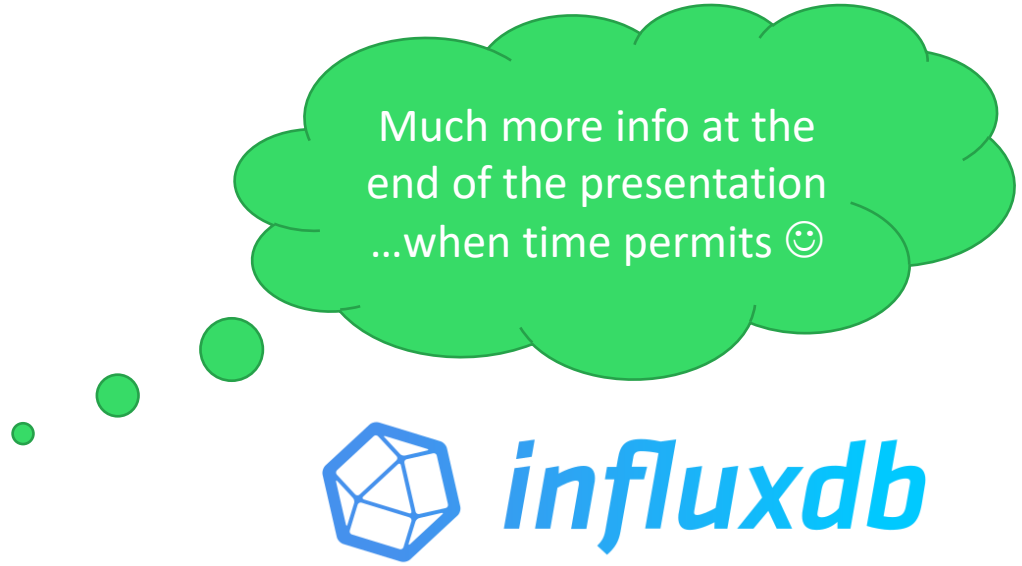
Web frontend

Telegraf

- Takes data on the input, process them and send them to the output

InfluxDB

- Time series database
- Open source
- Written in GO
- Easy to use
- Automated data retention policy
- Schemaless
- HTTP Based
- Designed to handle high write and query loads
- Integral component of the TICK stack
- Backing store for any use case involving large amounts of timestamped data



Grafana

- Open source visualization and analytics software
- Allows us to query, visualize and explore our data no matter where they are stored
- Offers bundled alerting system

Grafana


PostgreSQL

influxdb


Microsoft
SQL Server


MySQL


Prometheus


elasticsearch


Amazon
CloudWatch


Grafana

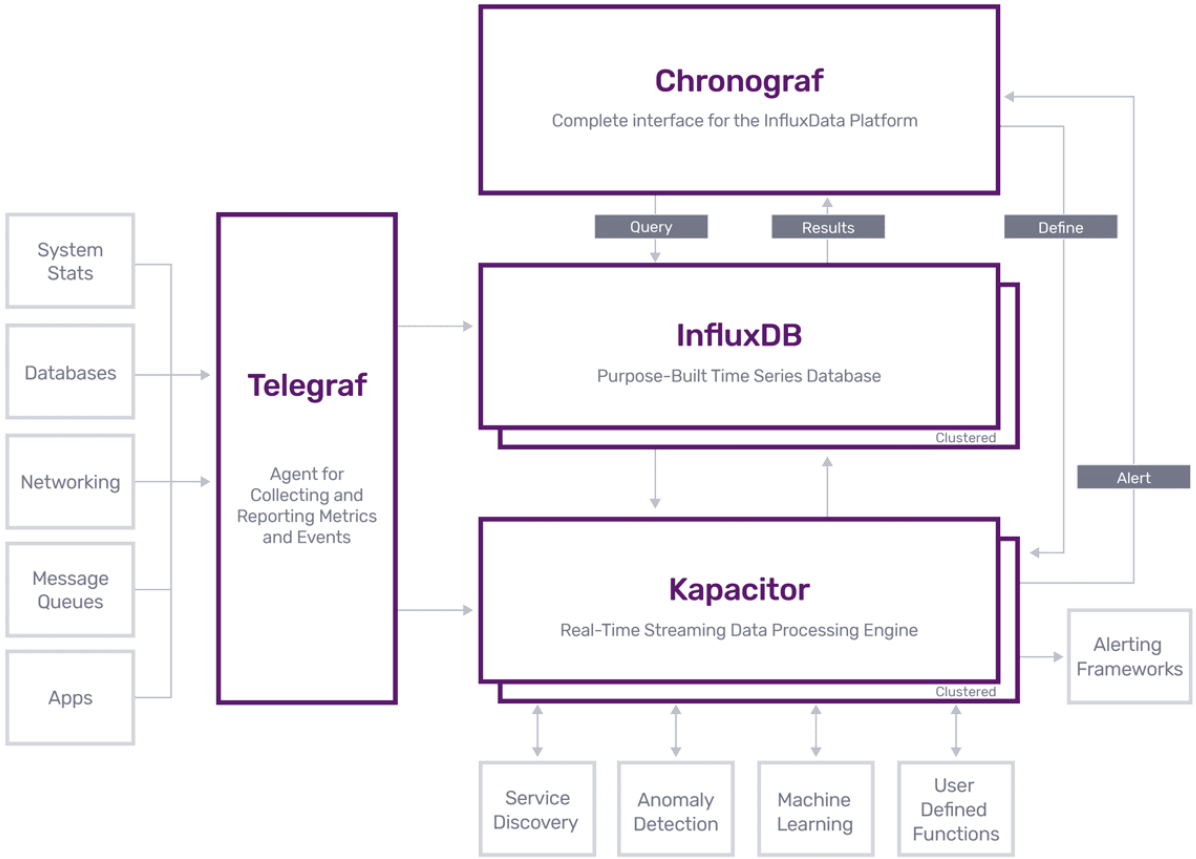
How to “plug it together”

1. Download InfluxDB and start the backend
2. Download Telegraf and send some data to InfluxDB
3. Download Grafana, connect it to InfluxDB and visualize the data

Other tools we can use

chronograf

Kapacitor



Summary

- Start with TIG stack is fairly easy with no initial costs
- Customization of official plugin can be bit tricky
- What we covered is the initial configuration
- Don’t forget to secure your endpoints!

Bonus section:

InfluxDB insights

Joyful Craftsmen

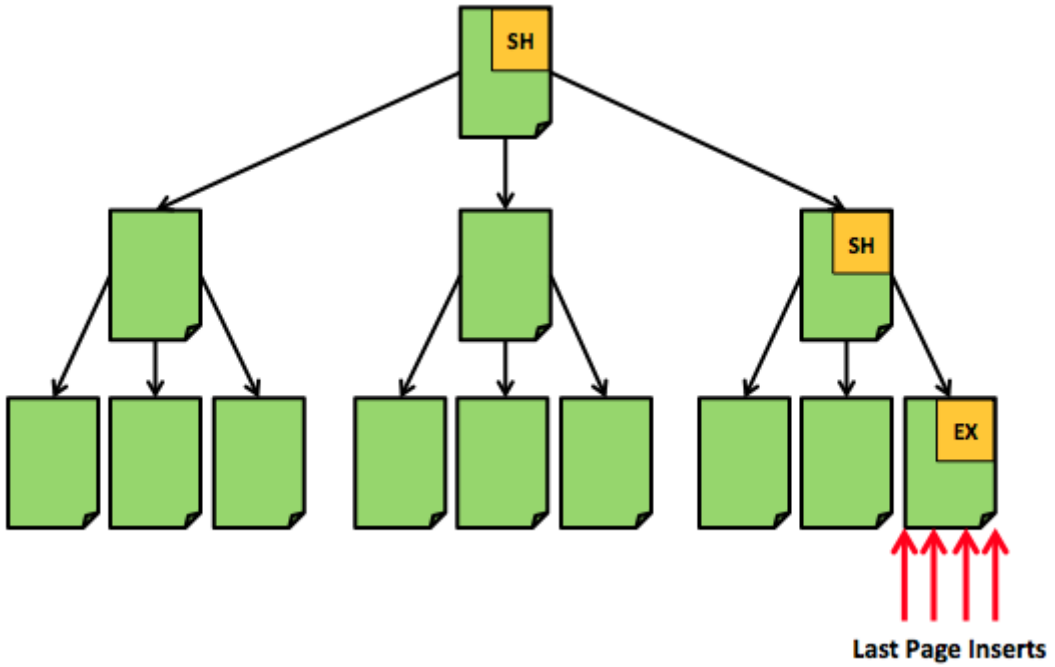
Business Intelligence Crafted to Joy

Why InfluxDB and no SQL Server?

More B-Trees Doesn't Mean Scaling

Imagine the scale of 2000 VMs with 100 sensors run every 10 seconds

What's wrong with B-Trees?



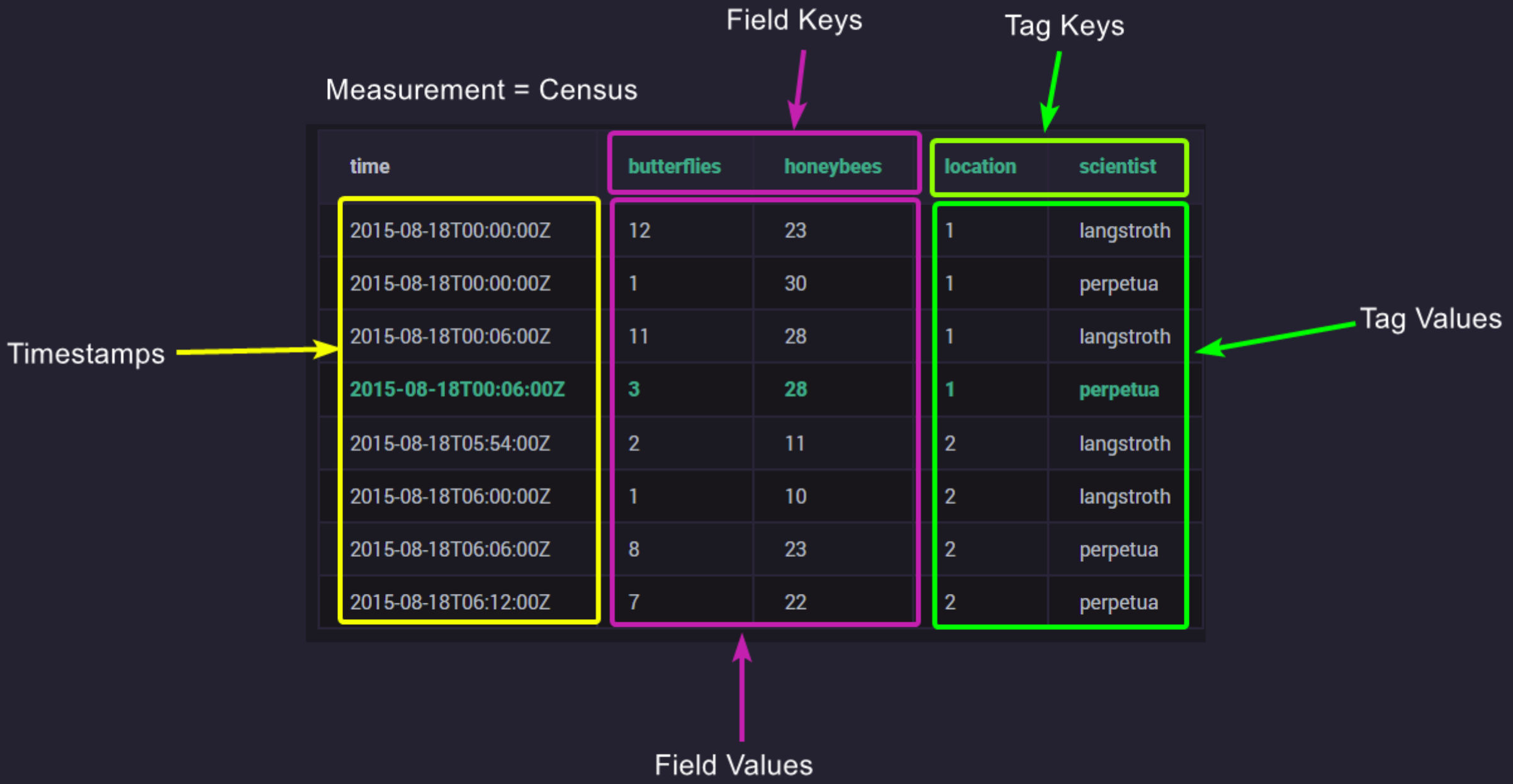
Where is the magic?

Key concepts

- Database
- Measurements
- Timestamps
- Tags
- Fields
- Keys & Values
- Retention policy
- Series

Databases

time	butterflies	honeybees	location	scientist
2015-08-18T00:00:00Z	12	23	1	langstroth
2015-08-18T00:00:00Z	1	30	1	perpetua
2015-08-18T00:06:00Z	11	28	1	langstroth
2015-08-18T00:06:00Z	**3**	**28**	**1**	**perpetua**
2015-08-18T05:54:00Z	2	11	2	langstroth
2015-08-18T06:00:00Z	1	10	2	langstroth
2015-08-18T06:06:00Z	8	23	2	perpetua
2015-08-18T06:12:00Z	7	22	2	perpetua

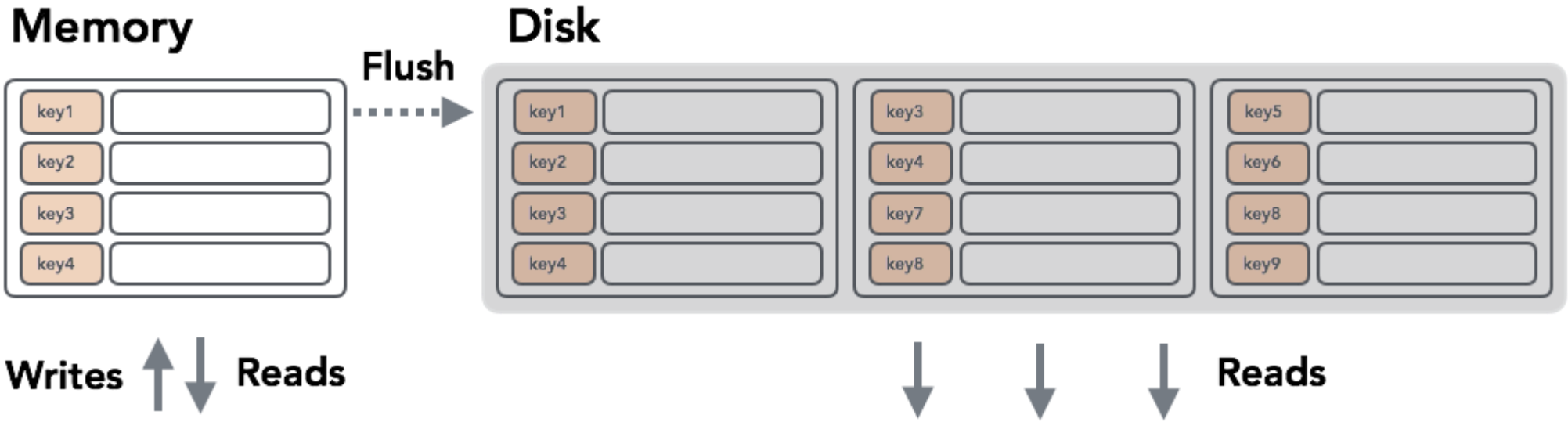


time	butterflies	honeybees	location	scientist
2015-08-18T00:00:00Z	12	23	1	langstroth
2015-08-18T00:00:00Z	1	30	1	perpetua
2015-08-18T00:06:00Z	11	28	1	langstroth
2015-08-18T00:06:00Z	**3**	**28**	**1**	**perpetua**
2015-08-18T05:54:00Z	2	11	2	langstroth
2015-08-18T06:00:00Z	1	10	2	langstroth
2015-08-18T06:06:00Z	8	23	2	perpetua
2015-08-18T06:12:00Z	7	22	2	perpetua

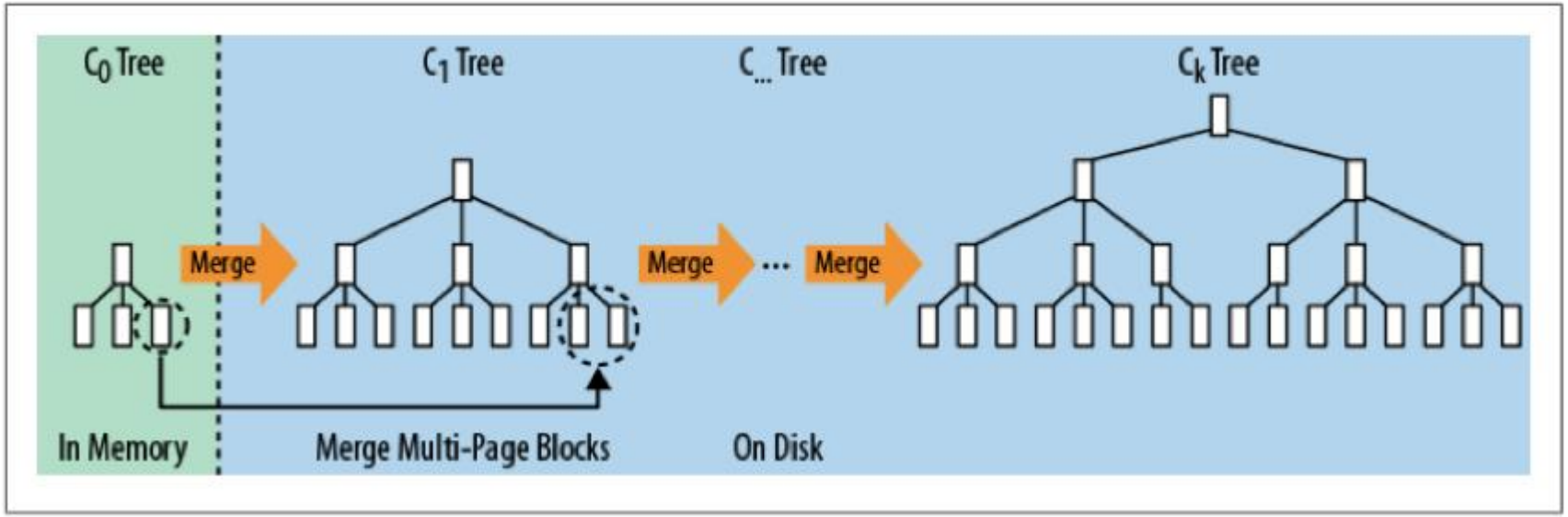
How cool would it be to have a sorted CCI with rowgroup elimination wherever I need?

LSM Tree

- Used in HBase, Cassandra, MongoDB, RocksDB…

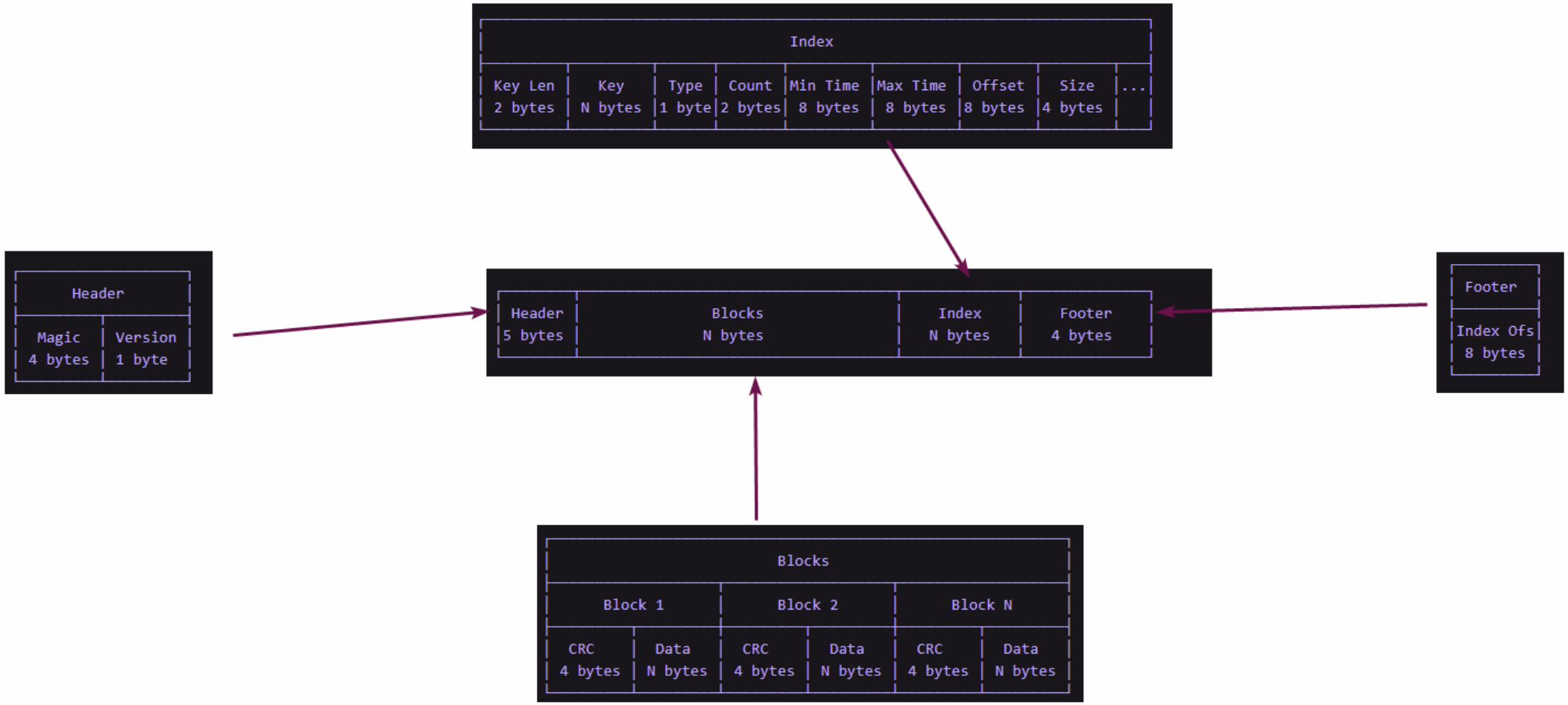


LSM Tree


C0 Tree
C1 Tree
C... Tree
Ck Tree
Merge
Merge
...
Merge
In Memory
Merge Multi-Page Blocks
On Disk

Storage engine key components

TSM File


Index
Key Len
2 bytes
Key
N bytes
Type
1 byte
Count
2 bytes
Min Time
8 bytes
Max Time
8 bytes
Offset
8 bytes
Size
4 bytes
...
Header
Magic
4 bytes
Version
1 byte
Header
5 bytes
Blocks
N bytes
Index
N bytes
Footer
4 bytes
Footer
Index Ofs
8 bytes
Blocks
Block 1
Block 2
Block N
CRC
4 bytes
Data
N bytes
CRC
4 bytes
Data
N bytes
CRC
4 bytes
Data
N bytes

Series ~ Indexes

Arbitrary series number	Retention policy	Measurement	Tag set
series 1	`autogen`	`census`	`location = 1` , `scientist = langstroth`
series 2	`autogen`	`census`	`location = 2` , `scientist = langstroth`
series 3	`autogen`	`census`	`location = 1` , `scientist = perpetua`
series 4	`autogen`	`census`	`location = 2` , `scientist = perpetua`

Series cardinality

The number of unique database, measurement, tag set, and field key combinations in an InfluxDB instance.

email	status
lorr@influxdata.com	start
lorr@influxdata.com	finish
marv@influxdata.com	start
marv@influxdata.com	finish
cliff@influxdata.com	start
cliff@influxdata.com	finish

Example Query

```
select percentile(90, value) from cpu
where time > now() -12h and "region" = 'west'
group by time(10m), host
```

Example Query

```
select percentile(90, idle) from cpu
where time > now() -12h and "region" = 'west'
group by time(10m), host
```

Cpu,host=A,region=west#idle -> 123455
Cpu,host=B,region=west#idle -> 266535

Discouraged schema design

- Don’t have too many series
- Don’t encode data in measurement names
- Don’t put more than one piece of information in one tag

Tradeoffs

- Data with same timestamp sent multiple times = same data

Pro: Simplified **conflict resolution** increases write performance.
Con: Cannot store duplicate data; may overwrite data in rare circumstances.

Tradeoffs

- Updates to existing data are a rare occurrence & contentious updates never happen

Pro: Restricting access to updates allows for increased query and write performance.
Con: Update functionality is significantly restricted.

Tradeoffs

- The vast majority of writes are for data with very recent timestamps and the data is added in time ascending order.

Pro: Adding data in time ascending order is significantly more performant.
Con: Writing points with random times or with time not in ascending order is significantly less performant.

Tradeoffs

- Scale is critical. The database must be able to handle a *high* volume of reads and writes.

Pro: The database can handle a *high* volume of reads and writes.
Con: The InfluxDB development team was forced to make tradeoffs to increase performance.

Tradeoffs

- No one point is too important.

Pro: InfluxDB has very powerful tools to deal with aggregate data and large data sets.
Con: Points don't have IDs in the traditional sense, they are differentiated by timestamp and series.

Tradeoffs

- Being able to write and query the data is more important than having a strongly consistent view.

Pro: Writing and querying the database can be done by multiple clients and at high loads.
Con: Query returns may not include the most recent points if database is under heavy load.

Tools inside InfluxDB

- InfluxQL
  - Autofill
  - Period Time
  - No joins!
- Continuous Queries
  - Like ETLs

Continuous Queries sample

```
CREATE CONTINUOUS QUERY "cq_basic_br" ON "transportation"
BEGIN
      SELECT mean(*)
      INTO "downsampled_transportation"."autogen".:MEASUREMENT
      FROM /.*/
      GROUP BY time(30m),*
END
```

Joyful Craftsmen

EVERYBODY MAKING DATA-DRIVEN DECESIONS SMARTLY.

Do you want to automate?

Contact: pavel.malek@joyfulcraftsmen.com

Thank you!

Joyful Craftsmen

Business Intelligence Crafted to Joy