

Fast and efficient operational time series storage: The missing link in dynamic software analysis

Symposium on Software Performance

This research was in part funded by
Bavarian Ministry of Economic Affairs
and Media, Energy and Technology.

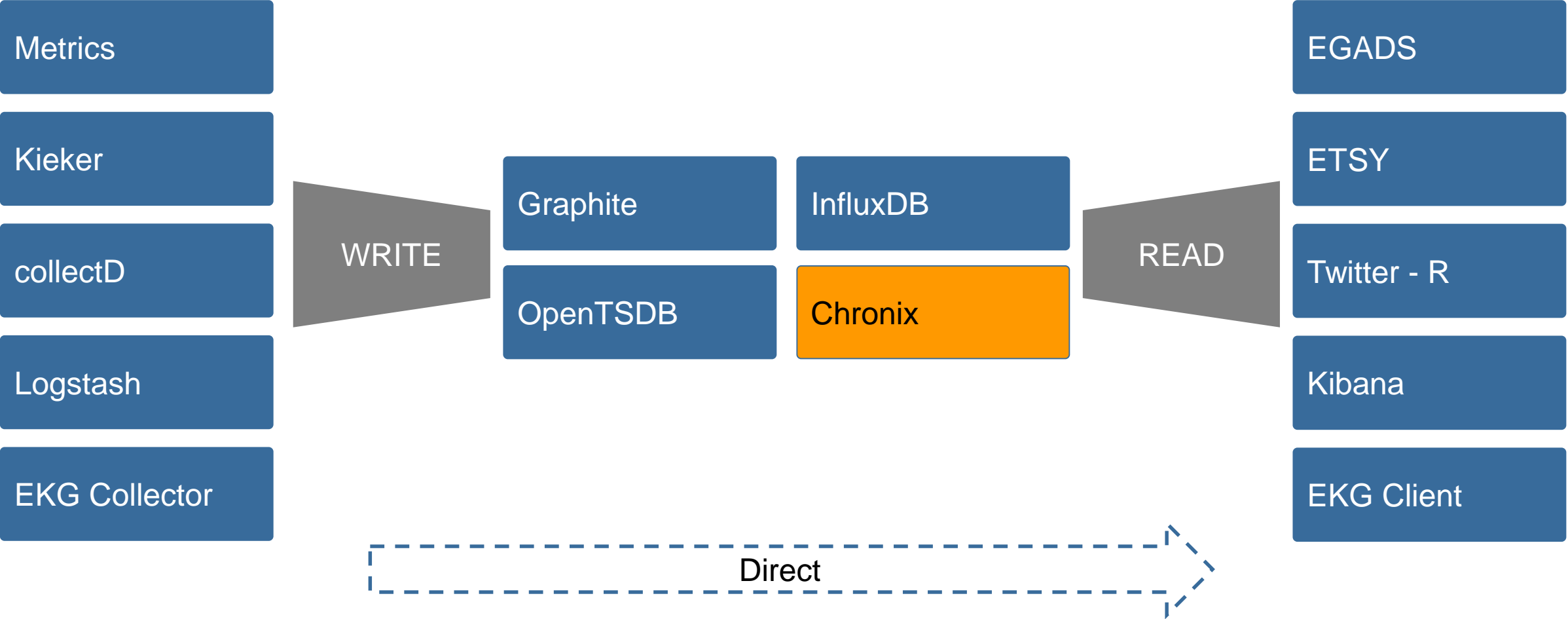
Munich, 05.11.2015
Florian Lautenschlager, Andreas Kumlehn, Josef Adersberger,
Michael Philippsen

What is operational data?

- Typical operational data are runtime metrics, e.g. CPU load, memory consumption, logs, exceptions, etc.
- Operational data is best represented as time series.
- Continuously harvested along a multitude of dimensions.
- Expected wide range of the values along each of the dimensions.
- Frequencies of time spans tend to vary a lot.

“...interactive response times often make a qualitative difference in data exploration, monitoring, online customer support, rapid prototyping, debugging of data pipelines, and other tasks.” [Dremel: Interactive Analysis of Web-Scale Datasets, Sergey Melnik et al.]

A typical toolchain for dynamic software analysis: collection framework, time series storage, time series analysis framework



Research Question:

Is it possible to exploit the characteristic features of operational data to create a time series database that requires less space and provides faster queries?

Chronix

Efficient storage

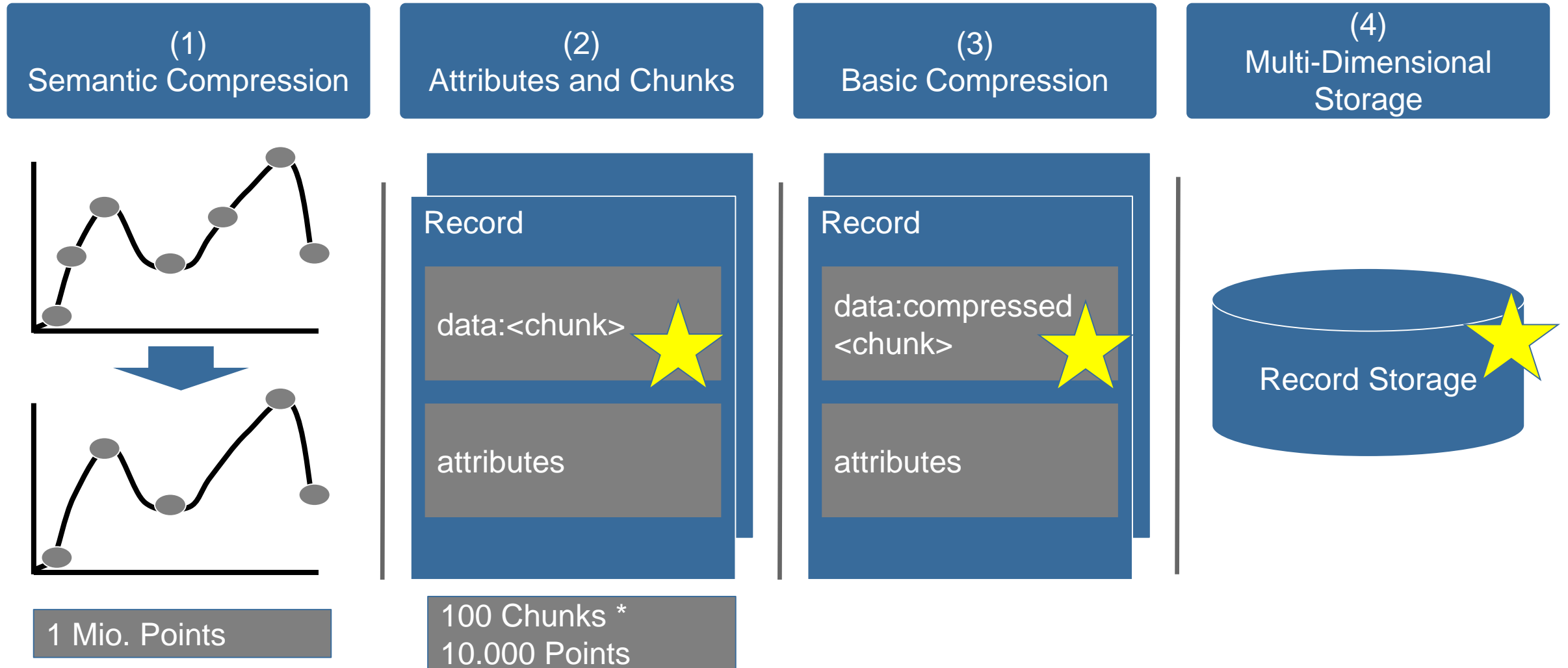
Fast queries

Extendable with analysis functions

Store every kind of operational data as time series

Scalable and portable

Yes. Chronix' architecture enables both efficient storage of time series and millisecond range queries.



The key data type of Chronix is called a record.
It stores a compressed chunk of the time series and its attributes.

```
record{  
  data:compressed{<chunk>}  
  //technical fields  
  id: 3dce1de0-...-93fb2e806d19  
  version: 1501692859622883300  
  start: 1427457011238  
  end: 1427471159292  
  
  //optional attributes  
  host: prodI5  
  process: scheduler  
  group: jmx  
  metric: heapMemory.Usage.Used  
  max: 896.571  
}
```

Data:compressed{<chunk of time series data>}

- Time Series: time stamp, numeric value
- Traces: calls, exceptions, ...
- Logs: access, method runtimes
- Complex data: models, test coverage, anything else...

Optional attributes

- Arbitrary attributes for the time series
- Attributes are indexed
- Make the chunk searchable
- Can contain pre-calculated values

Chronix also provides aggregations and higher-level time series analyses in its query language that other TSDBs do not.

Aggregations (ag)

- Maximum
- Minimum
- Average
- Standard Deviation
- Percentile

Analyses (detect)

- A trend analysis based on a linear regression model.
- An outlier analysis using the IQR.
- A frequency analysis validating the occurrence within a defined time range.

```
q=host:prod? AND group:(jmx OR .net) & fq={!ANALYZE ag=dev}
```

```
q=host:* AND -group:(jmx OR .net) & fq={!ANALYZE detect=frequency=10:6}
```


Benchmarks represent typical use cases in time series analysis. The queries are collected from real-world analyses.

- We have collected, arranged, and counted queries of real analyses.

Time Range (Days)	#Queries
1	30
7	30
14	10
91	2

We repeat the 72 queries 20 times to stabilize results.

- Three real-world project's operational time series data (14,195 time series, 512 Mio. points).
 - Project 1: Web application for searching car information (8 web server, 20 search server)
 - Project 2: Retail application for orders, billing, and customer relations (2 servers, 1 central database)
 - Project 3: Sales application of a car manufacturer (2 servers, 1 central database)

Chronix outperforms related TSDBs in write throughput, storage efficiency, and access times.

Table 1: Write throughput in MByte/seconds.

Project	CSV (MByte)	Graphite	InfluxDB	OpenTSDB	Chronix	Chronix (SC)
	129	4.96	2.74	0.61	11.73	11.73
	5,400	5.24	2.69	0.63	21.26	22.88
3	9,600	4.19	2.73	0.55	22.86	24.12
avg	5,043	4.80	2.72	0.59	18.61	19.58

4-33 times higher

Chronix outperforms related TSDBs in write throughput, storage efficiency, and access times.

Table 2: Storage demands in MBytes.

Project	CSV	Graphite	InfluxDB	OpenTSDB	Chronix	Chronix (SC)
1	29	758	144	254	4	2
2	5,400	14,000	11,000	263	226	74
3	9,600	26,000	21,000	758	632	162
total	15,129	40,758	32,144	1,275	862	238

5 - 171 times less space

Chronix outperforms related TSDBs in write throughput, storage efficiency, and access times.

Table 3: Query times for 20 · 72 queries in sec.

		Graphite	InfluxDB	OpenTSDB	Chronix	Chronix (SC)
	30	12.3	20.4	12.8	12.3	10.2
7	30	15.1	100.3	30.3	13.6	11.0
14	10	5.5	33.6	10.7	4.5	3.8
91	2	2.3	17.7	4.2	1.1	0.9
total		35.2	172.0	58.0	31.5	25.9
Outlier		48.6	95.1	66.5	41.2	35.1
Trend		41.3	175.3	62.7	35.1	35.1
total		89.9	270.4	129.2	76.9	66.4

save 26-85% of the query times

speeds things up by 29-75%

Chronix is open-source. Check <http://www.chronix.io/> or @ChronixDB



Chronix is currently more a proof-of-concept than production-ready. Work is going on!

Contact: florian.lautenschlager@qaware.de