DATASTAX

ACCELERATE

Cassandra Capacity Planning Using Facebook Prophet

23 May 2019

DATASTAX®

Speakers

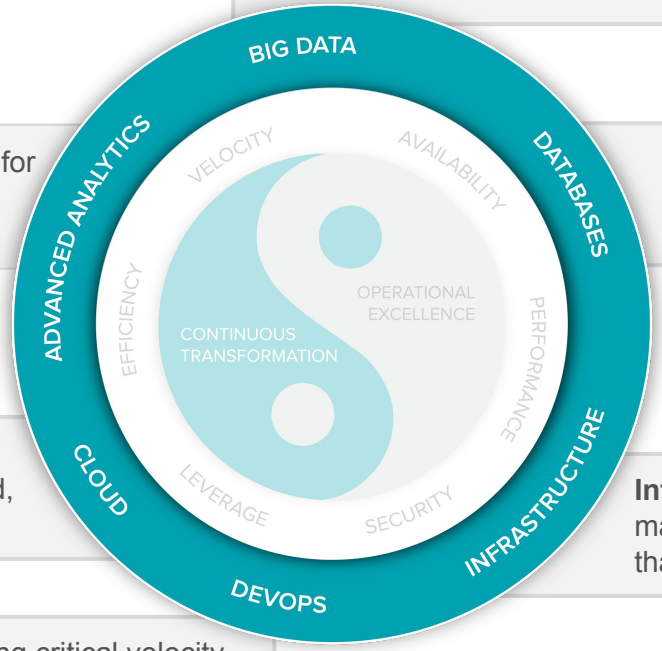
- Pedro Vidigal, Cassandra Team Manager, Pythian
  - Pedro's consultancy experience, along with his MBA in management, gives him the business knowledge necessary to interact with clients and align technology-related decisions with the company and client goals, identify opportunities and risks for the business, and build and design solutions that solve a client's need regardless of the underlying technology.

- Valerie Parham-Thompson, Cassandra DBA, Pythian
  - With experience as an open-source DBA and developer for software-as-a-service environments, Valerie has expertise in web-scale data storage and data delivery.

Pythian

TECHNICAL EXPERTISE

Big Data: Harnessing the transformative power of data on a massive scale



Advanced Analytics: Mining data for insights & business transformation using data science

Databases: Ensuring databases are reliable, secure, available and continuously optimized

Cloud: Using the disruptive nature of cloud for accelerated, cost-effective growth

Infrastructure: Transforming and managing the IT infrastructure that supports the business

DevOps: Providing critical velocity in software deployment by adopting DevOps practices

DATASTAX®

DATASTAX

ACCELERATE

Capacity Planning

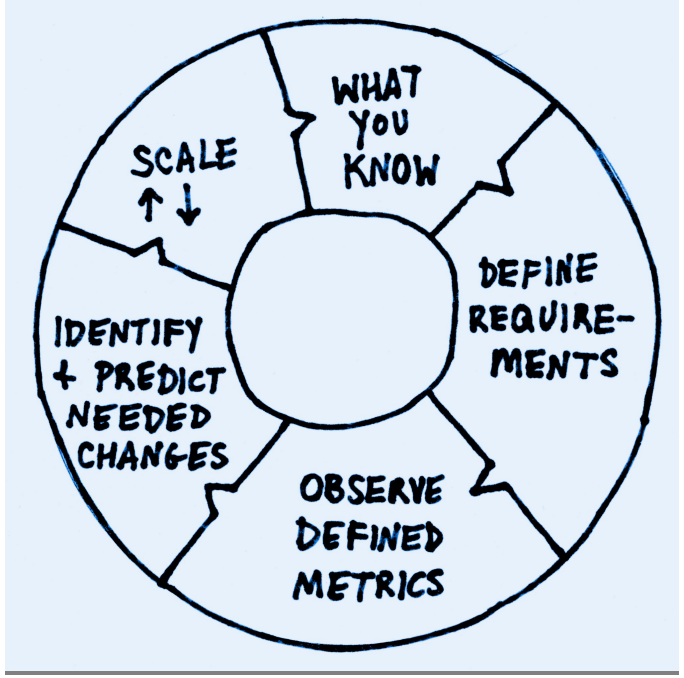
DATASTAX®

What is capacity planning?

The goal of capacity planning is to:

Take **what you know** about the environment, define business **requirements**, observe **metrics** related to those definitions, identify and predict **changes** needed, and then **scale** resources up or down accordingly.

This is an iterative process.



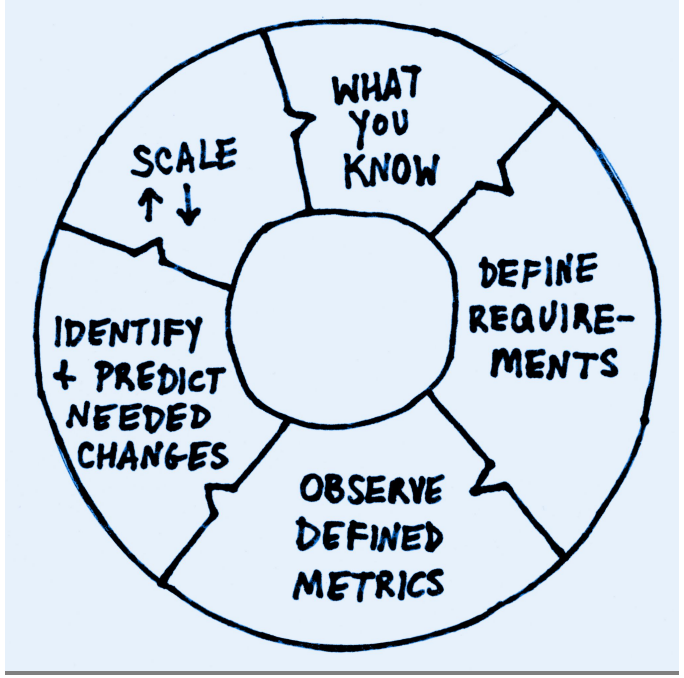
DATASTAX

What is capacity planning?

You already have -- and will continue to build -- assumptions and **knowledge about the environment**.

Consider what you know about:

- Traffic peaks
- Relative downtime
- Upcoming application releases
- New features
- Industry trends



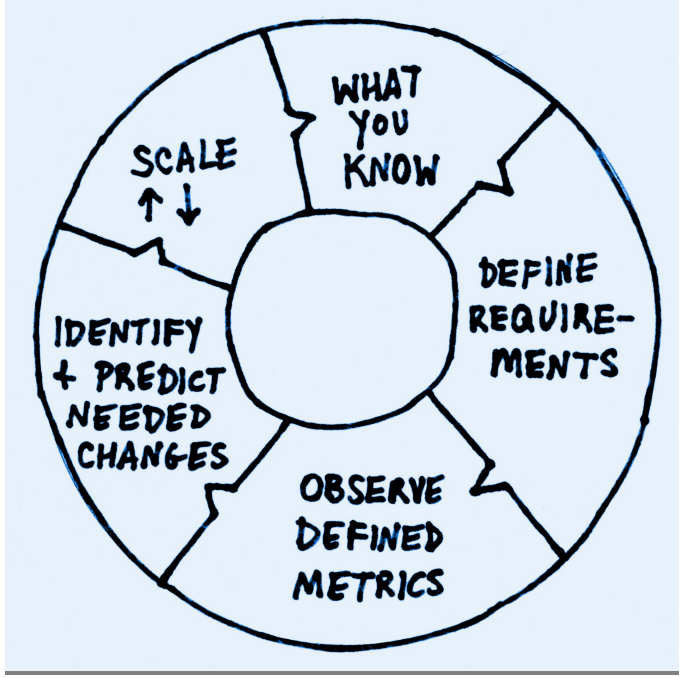
DATASTAX

What is capacity planning?

Capacity planning requires an input of **clearly defined performance requirements**.

Base this on your business needs. Examples are:

Desired response time (e.g., how quickly a page loads)
User experience (e.g., how smooth a video plays back)
Number of X (e.g., downloads) delivered per X (e.g., seconds)



DATASTAX

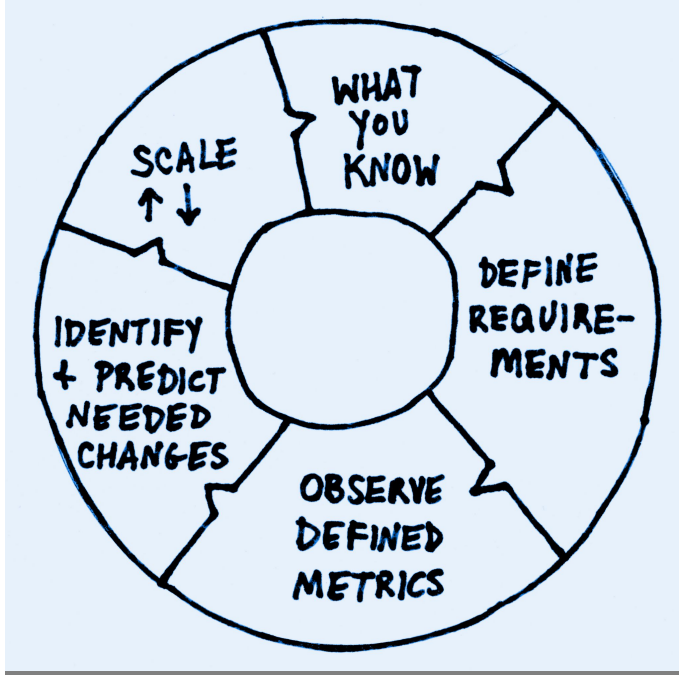
What is capacity planning?

Measure how well the business requirements are being met with related **metrics**.

Many modern monitoring tools have application monitoring or similar to pinpoint performance throughout the application.

Sometimes you will need to use resource monitoring to act as a proxy for measuring performance defined by business requirements.

Think: Is this throughput? Or saturation? Depends on business requirements.



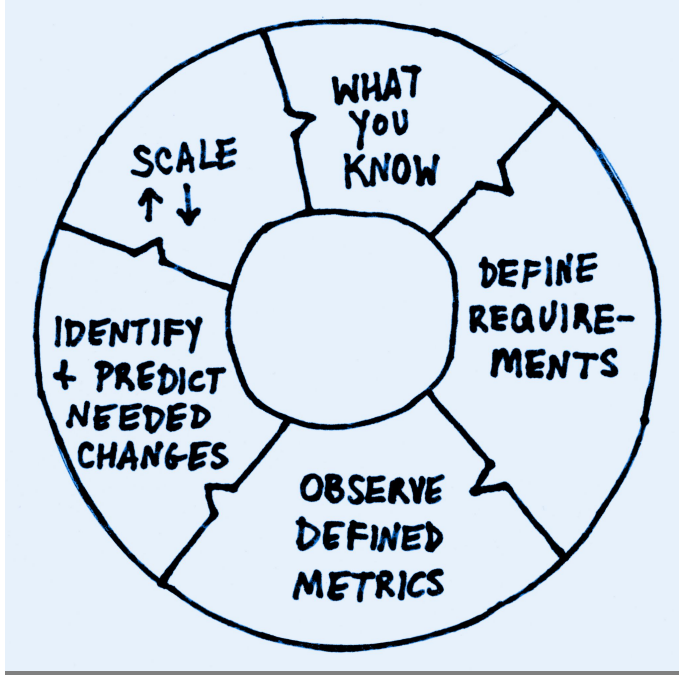
DATASTAX

What is capacity planning?

Using the observed metrics and your business requirements, **identify the current changes** needed. Perhaps you need more memory, stat!

For bonus points, **predict necessary changes** to support future needs. (This is what our presentation is about.) This is not the same as trend analysis, but it includes similar elements.

Don’t plan for averages -- Prepare for **peak** + safety margin. The quicker you can go through the whole cycle, the less overhead buffer you need to maintain… but you do need to think about peak traffic.

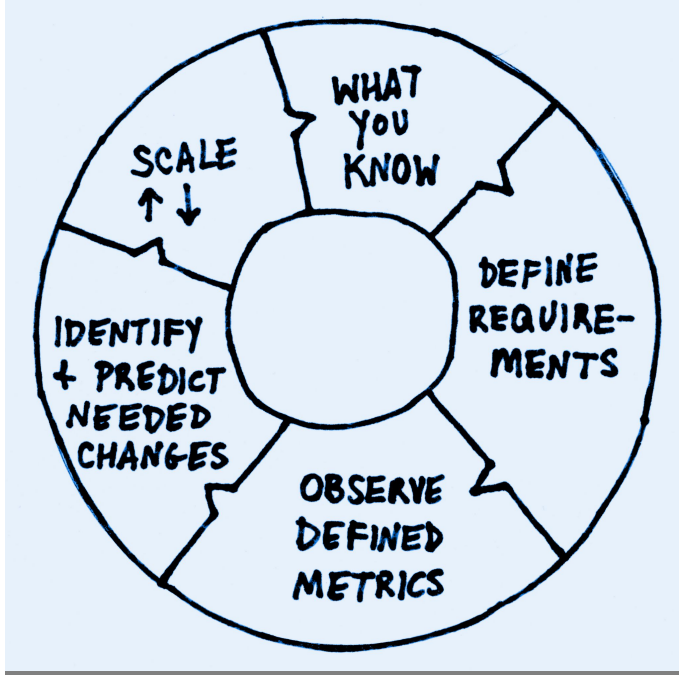


DATASTAX®

What is capacity planning?

And then **scale** based on what you've learned.

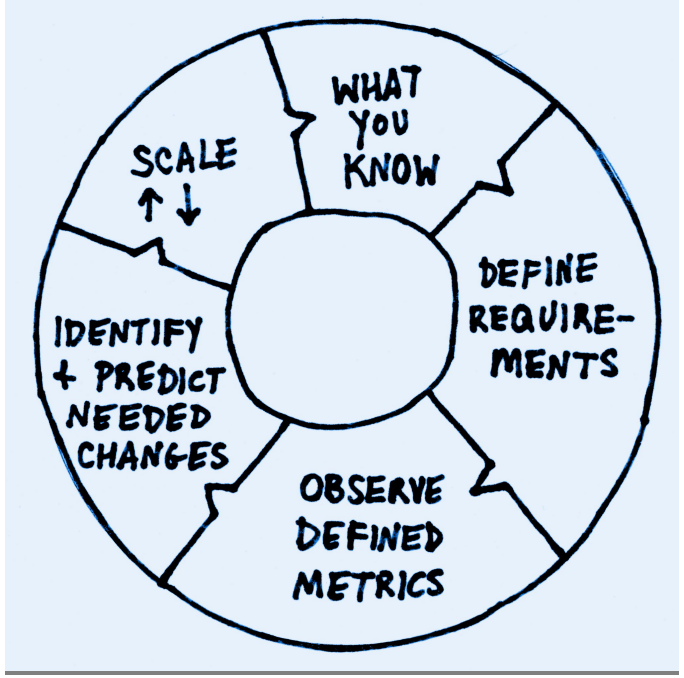
Scale up to support more traffic.
Scale down to save money.
Right-size your resources.



What is capacity planning?

Finally, capacity planning is not a one-time or even yearly task.

It is an ongoing process, **iterating** on past decisions and integrating **new information**.



DATASTAX

DATASTAX

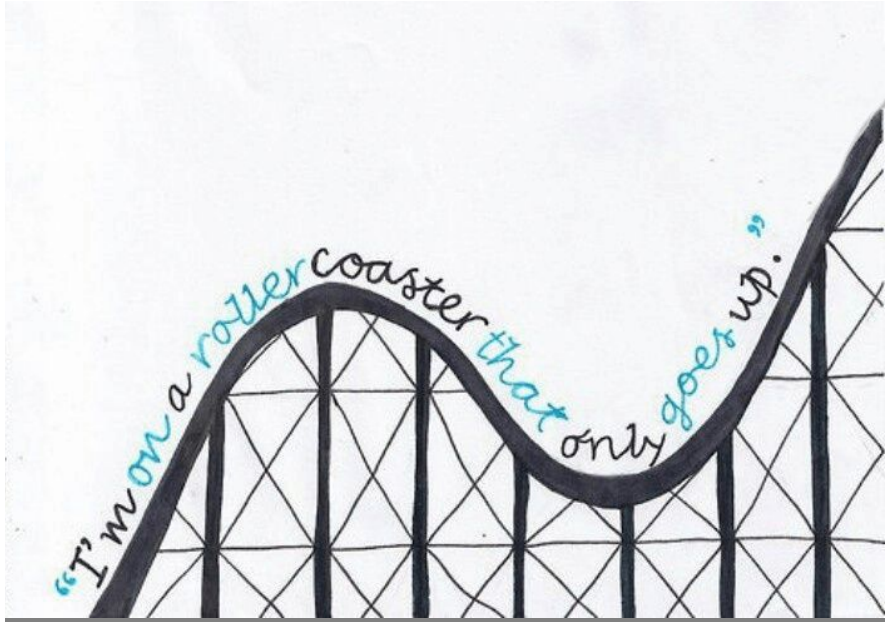
ACCELERATE

Challenges with Capacity Planning in Cassandra

DATASTAX®

First, Opportunities

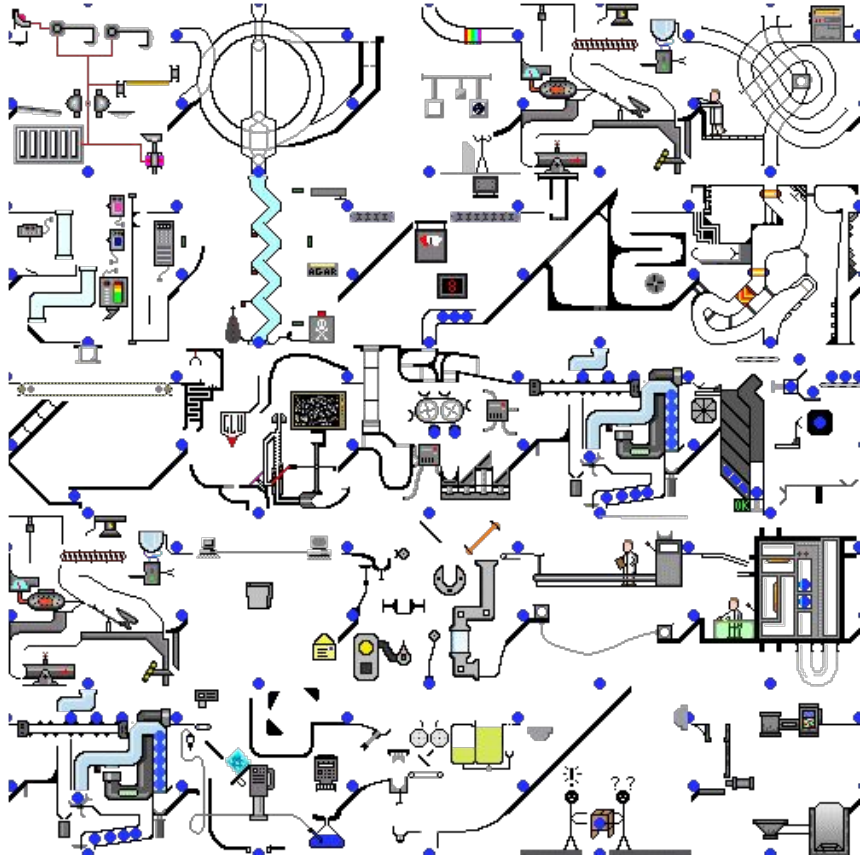
With Cassandra, it is not difficult to scale up and down. Nodes can be (relatively) easily added or decommissioned. This is a vast improvement over most database systems.



DATASTAX

Cassandra Challenge

Cassandra - or any element in your architecture - exists within a **stack of software**, where there may exist **other potential bottlenecks** limiting your throughput.



DATASTAX

Cassandra Challenge

Cassandra cannot always utilize 100% of server resources.

One example: some **disk space** must be left free **for compaction**.

DATASTAX

DATASTAX

ACCELERATE

Facebook Prophet

DATASTAX®

What is Facebook Prophet?

Marketing says:

"Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have **strong seasonal effects** and several seasons of **historical data**."

- Hourly, daily, or weekly observations with at least a few months (preferably a year) of history
- Strong multiple "human-scale" seasonalities: day of week and time of year
- Important holidays that occur at irregular intervals known in advance (e.g., the Super Bowl)

What is Facebook Prophet?

The academics add:

"We propose a **modular regression model** with interpretable parameters that can be intuitively adjusted by analysts with domain knowledge about the time series."

- A reasonable number of missing observations or **large outliers**
- **Historical trend changes**, for instance due to product launches or logging changes
- Trends that are **non-linear growth curves**, where a trend hits a natural limit or saturates

See also the white paper: Capacity at Scale, 27 September 2017

The Forecast Model

The Prophet uses a decomposable time series model with three main model components: trend, seasonality, and holidays. They are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

- **g(t)**: piecewise linear or logistic growth curve for modelling non-periodic changes in time series
- **s(t)**: periodic changes (e.g., weekly/yearly seasonality)
- **h(t)**: effects of holidays (user-provided) with irregular schedules
- **εt**: error term accounts for any unusual changes not accommodated by the model

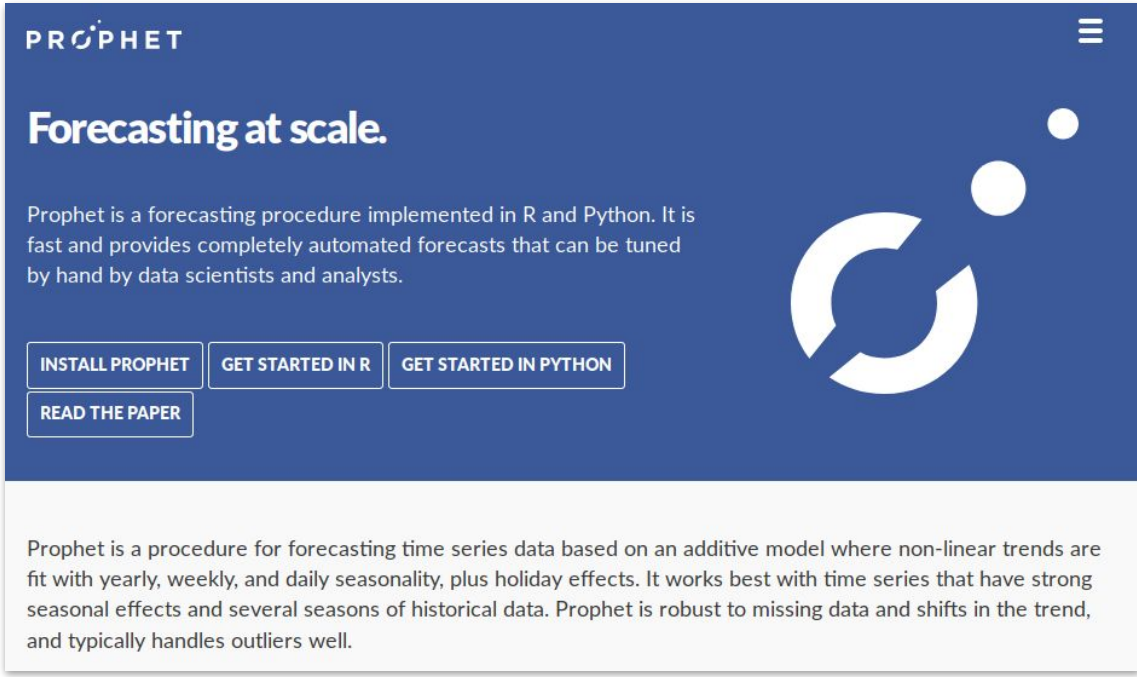
Facebook Prophet Installation

- Prophet is **included in some analytics tools**, such as Google Colaboratory.
- Otherwise, installation could include some **dependencies** (e.g., pystan).
- On Windows, may require you to **build from source**.

- Links to installation:

  https://facebook.github.io/prophet/docs/installation.html

  https://facebook.github.io/prophet/docs/quick_start.html



DATASTAX

ACCELERATE

Cassandra + Prophet

DATASTAX®

Concept

- Monitoring tools such as Prometheus or Datadog generate logs of metrics.

- Predictions are then made on a dataframe with a column ds containing the dates for which a prediction is to be made.

- Prophet follows the ***sklearn*** model API. We create an instance of the *Prophet* class and then call its *fit* and *predict* methods

- The resulting forecast graphs are then interpreted using application and other environment knowledge.

	ds	y
0	2007-12-10	9.590761
1	2007-12-11	8.519590
2	2007-12-12	8.183677
3	2007-12-13	8.072467
4	2007-12-14	7.893572

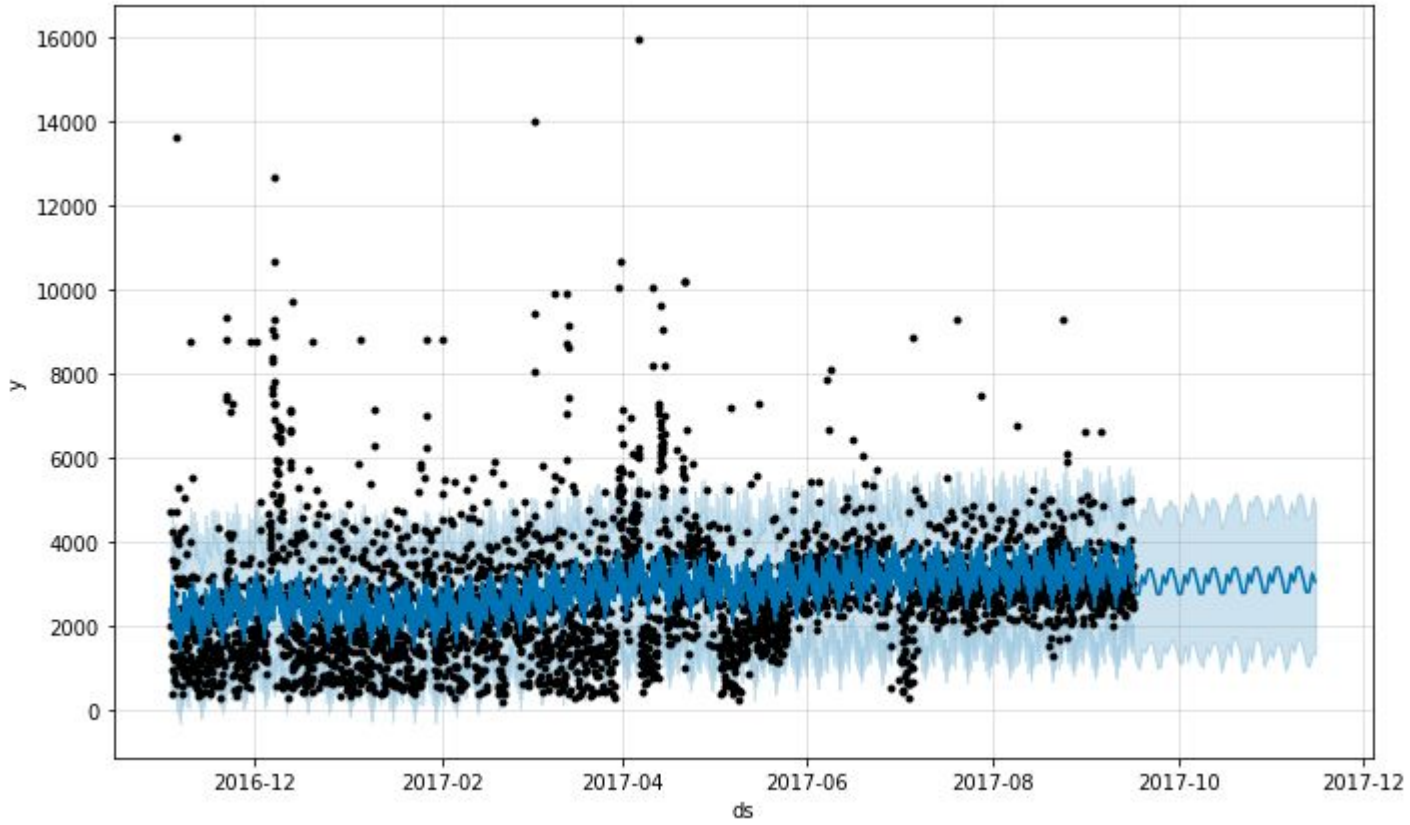
Disk: Write Rate

- Example of **os disk write rate**

```
m = Prophet()
m.fit(data)

future = m.make_future_dataframe(periods=60)
forecast = m.predict(future)

fig1 = m.plot(forecast)
```



DATASTAX

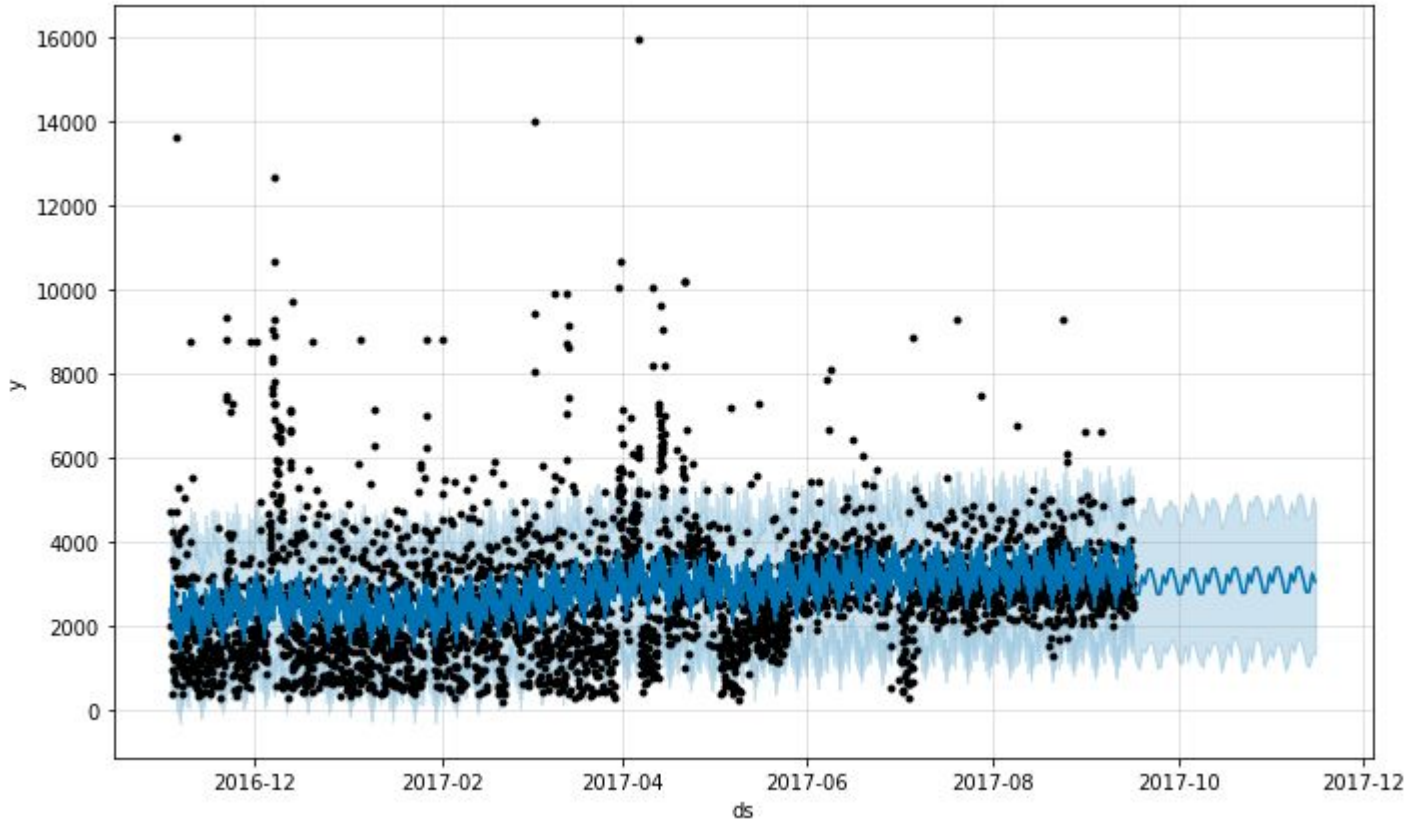
Observign Used disk space, makes more sense

Disk: Write Rate

- Example of **os disk write rate**

```
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

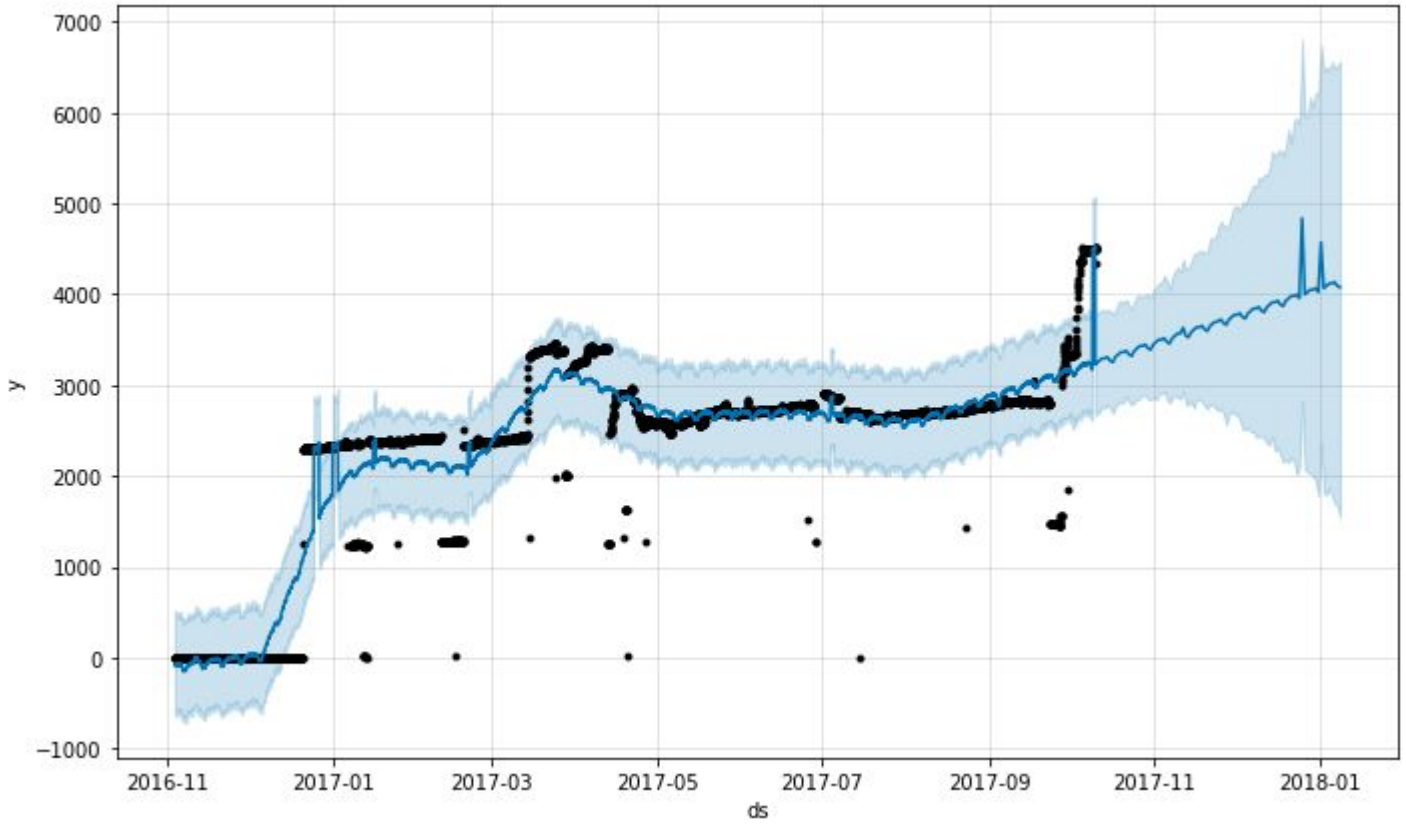
	ds	yhat	yhat_lower	yhat_upper
3855	2017-11-11 04:00:00	3133.404821	1393.168738	4978.006530
3856	2017-11-12 04:00:00	2800.806658	1053.173242	4544.717545
3857	2017-11-13 04:00:00	2803.199043	1030.323741	4622.505354
3858	2017-11-14 04:00:00	3243.646829	1340.007803	5086.424775
3859	2017-11-15 04:00:00	3044.518417	1308.816581	4884.173036



DATASTAX

Disk: Growth

- Observing disk **usage**
- We know from the environment that the disk space was extended.
- But we could see that **disk needed to be extended** (or the **cluster expanded**).



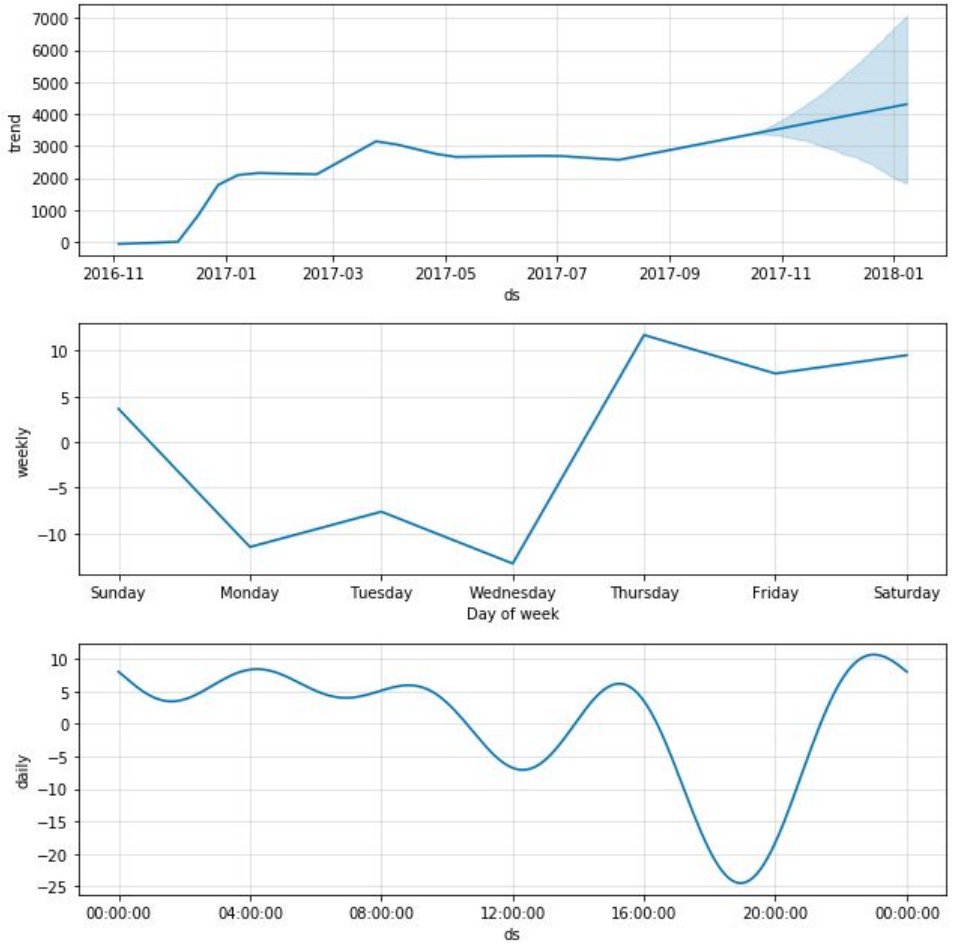
DATASTAX

Example: Analyze seasonality

- Prophet provides a way to analyse the overall growth trend as well as weekly and daily seasonality.
- In this scenario we can expect a bigger number of operations (writes) from Thursday to Sunday.

Forecast Components

```
fig2 = m.plot_components(forecast)
```



DATASTAX

Example: Special Events

- Consider special events, perhaps Black Friday or Christmas week for an e-commerce website, or spikes related to product releases or marketing.

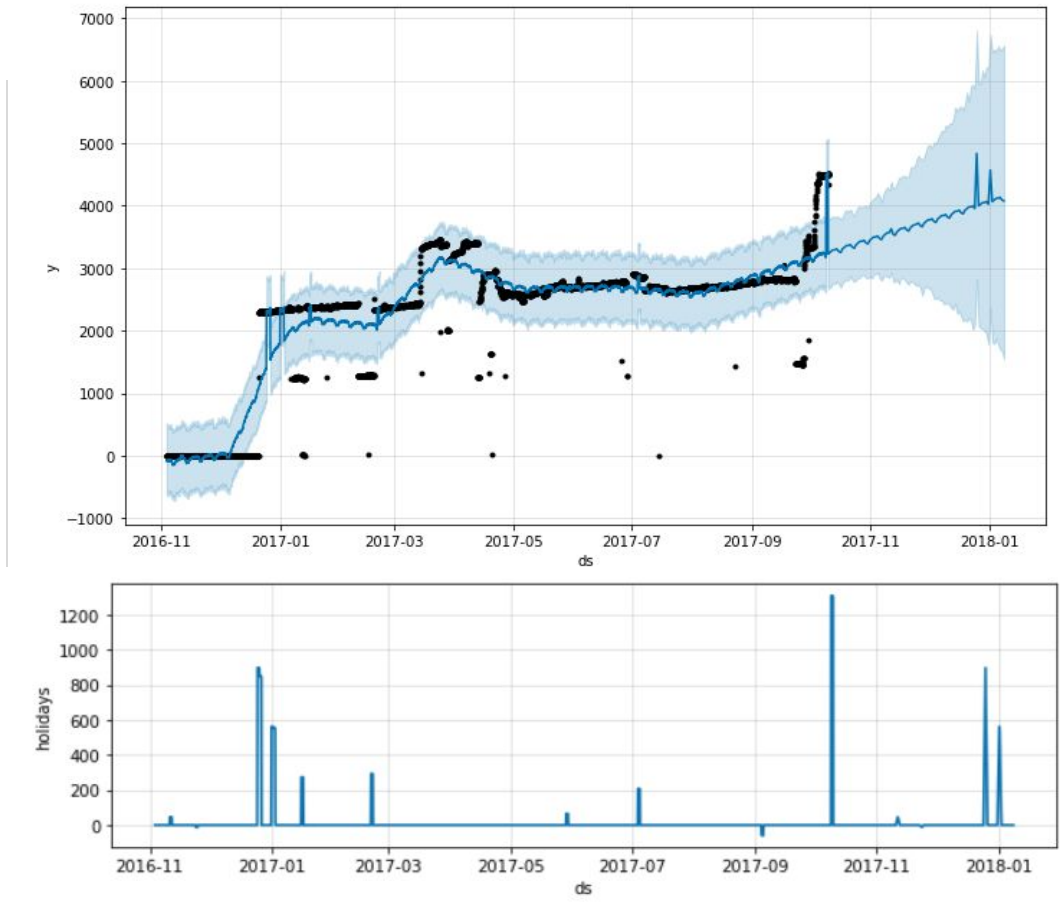
```
m = Prophet(yearly_seasonality=False)

m.add_country_holidays(country_name='US')

m.fit(data)

future = m.make_future_dataframe(periods=60)
forecast = m.predict(future)

fig2 = m.plot_components(forecast)
```



DATASTAX

Example: Special Events

- It´s easy to use your own defined events

Special Events

In [4]:

```
## set a dataframe for each holiday

# thanksgiving
thanksgiving = pd.DataFrame({
  'holiday': 'thanksgiving',
  'ds': pd.to_datetime(['2015-11-26',
                        '2016-11-24',
                        '2017-11-23',
                        '2018-11-22',
                        '2019-11-28']),
  'lower_window': 0,
  'upper_window': 1,
})

christmas = pd.DataFrame({
  'holiday': 'christmas',
  'ds': pd.to_datetime(['2017-12-25',
                        '2018-12-25',
                        '2019-12-25']),
  'lower_window': -1,
  'upper_window': 1,
})

special_events = pd.concat((thanksgiving,
                            christmas))
```

In [6]:

```
special_events
```

Out[6]:

	holiday	ds	lower_window	upper_window
0	thanksgiving	2015-11-26	0	1
1	thanksgiving	2016-11-24	0	1
2	thanksgiving	2017-11-23	0	1
3	thanksgiving	2018-11-22	0	1
4	thanksgiving	2019-11-28	0	1
0	christmas	2017-12-25	-1	1
1	christmas	2018-12-25	-1	1
2	christmas	2019-12-25	-1	1

DATASTAX

Example: Special Events

- Once the table is created, special event effects are included by passing them with the *holiday* argument.

```
m = Prophet(yearly_seasonality=False,
            holidays=special_events)

forecast = m.fit(data).predict(future)
```

- The *holiday* effect can be seen in the forecast dataframe, as well as

```
forecast[(forecast['thanksgiving'] + forecast['christmas']).abs() > 0][
        ['ds', 'thanksgiving', 'christmas']][-5:]
```

	ds	thanksgiving	christmas
264	2016-11-25 14:00:00	-1173.803066	0.0
265	2016-11-25 16:00:00	-1173.803066	0.0
266	2016-11-25 18:00:00	-1173.803066	0.0
267	2016-11-25 20:00:00	-1173.803066	0.0
268	2016-11-25 22:00:00	-1173.803066	0.0

DATASTAX

ACCELERATE

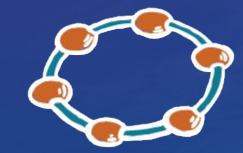
Demo!

DATASTAX®

DATASTAX

ACCELERATE

THANK YOU



DATASTAX®

Contact Us

- Pedro Vidigal @PedroCAVidigal
- Valerie Parham-Thompson @dataindataout

- Download the data files and notebook at: https://github.com/pythian/capcontrol