solarwinds

Innovate Summit 2017

Analytics-Driven Development

Karlo Zatylny

Who am I?

karlo.zatylny@solarwinds.com

$$Life = \int_{birth}^{death} \frac{happiness}{time} \Delta\, time$$

Leave Now!

solarwinds

Take aways:

- Better monitoring through analyzing statistical data sets
- Reduce overall system complexity
- Cleaner data sets - reduced data sets
- Gather intelligence, not data
- Worldwide data grows at the same rate as landfill garbage
- Next generation of computing is data-driven computing

- Are your analytics facilitating the human or making decisions in place of the human?

- P.S. Please use Google® during my presentation if you don't know a word or phrase.

What do we do at SolarWinds?

We monitor the software you write and the hardware you build, which consists of:

- Mostly reading available metrics exposed by APIs and protocols
- Triggering alerts on user defined or calculated thresholds
- Storing loads of data that no one ever sees or ultimately cares about

Order of operations: + - × ÷

solarwinds

What do you do when there is a problem in production?

Mom questions...

Why isn’t the printer printing?

The CPU is spiked. Why?

The software isn’t working. Why?

Why is the software slow?

Time Reproduce Logs Data Performance Counters More Logs

When did the problem start?
Which systems were affected?
What changed?
Can you reproduce the issue?
Can you send me the logs?
Can you run the diagnostics tool?

I am the Google search of my logs

- Yay! You got the logs.
- Let’s chip!
- Luckily the system has 42 sets of logs that are not centralized.
- Our system has centralized logs that create 500 MB of logs per minute.
- I will search for “ERROR.”

What is monitoring?

- Monitoring is the active or passive gathering of numeric and text data for storage and analysis.

Steps:

1. Gather data
2. Analyze data
3. Trigger alerts on significant events
4. Execute remediation actions (email the admin)

- What's missing?
- Solving the actual problem.

Prepare to be monitored

solarwinds

- Metrics-Driven Development
- Write your software in a way that is designed to be monitored
- Instrument your code with metrics ready to give you critical data for:
  - Performance
  - Health
  - Troubleshooting

- Provide an easy way for this data to be read
  - http://blog.librato.com/posts/2014/7/16/metrics-driven-development
  - http://blog.librato.com/posts/metrics-driven-development-2
  - https://www.infoq.com/articles/metrics-driven-development
  - https://sookocheff.com/post/mdd/mdd/
  - http://metrics.dropwizard.io/3.1.0/manual/core/

Data is not enough

Today, most systems pull or receive raw values and perform storage and analysis.

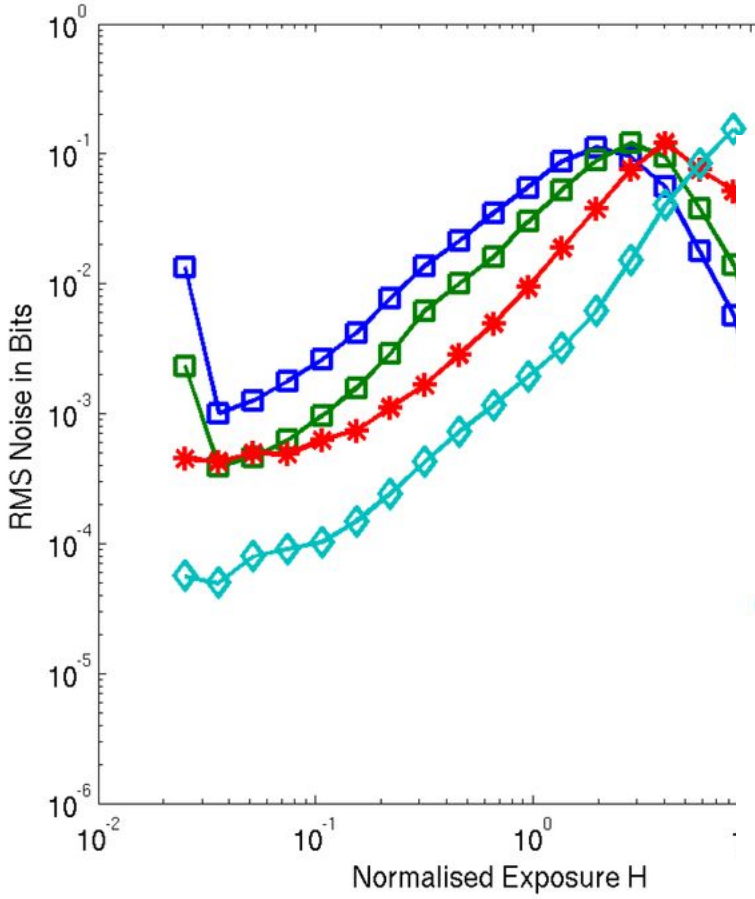
How do we improve this?

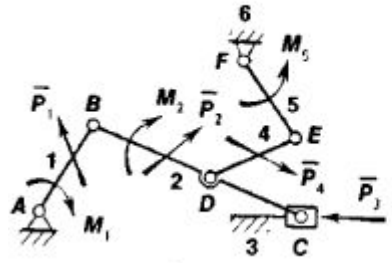
- Random Sampling?
- Distributed Storage?
- ETL?
- Parallelization?
- MapReduce!?
- Stream processing?

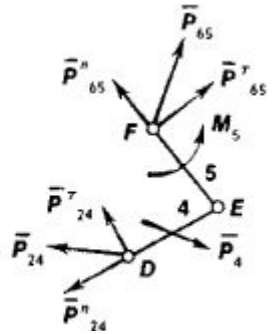
Analytics-Driven Development

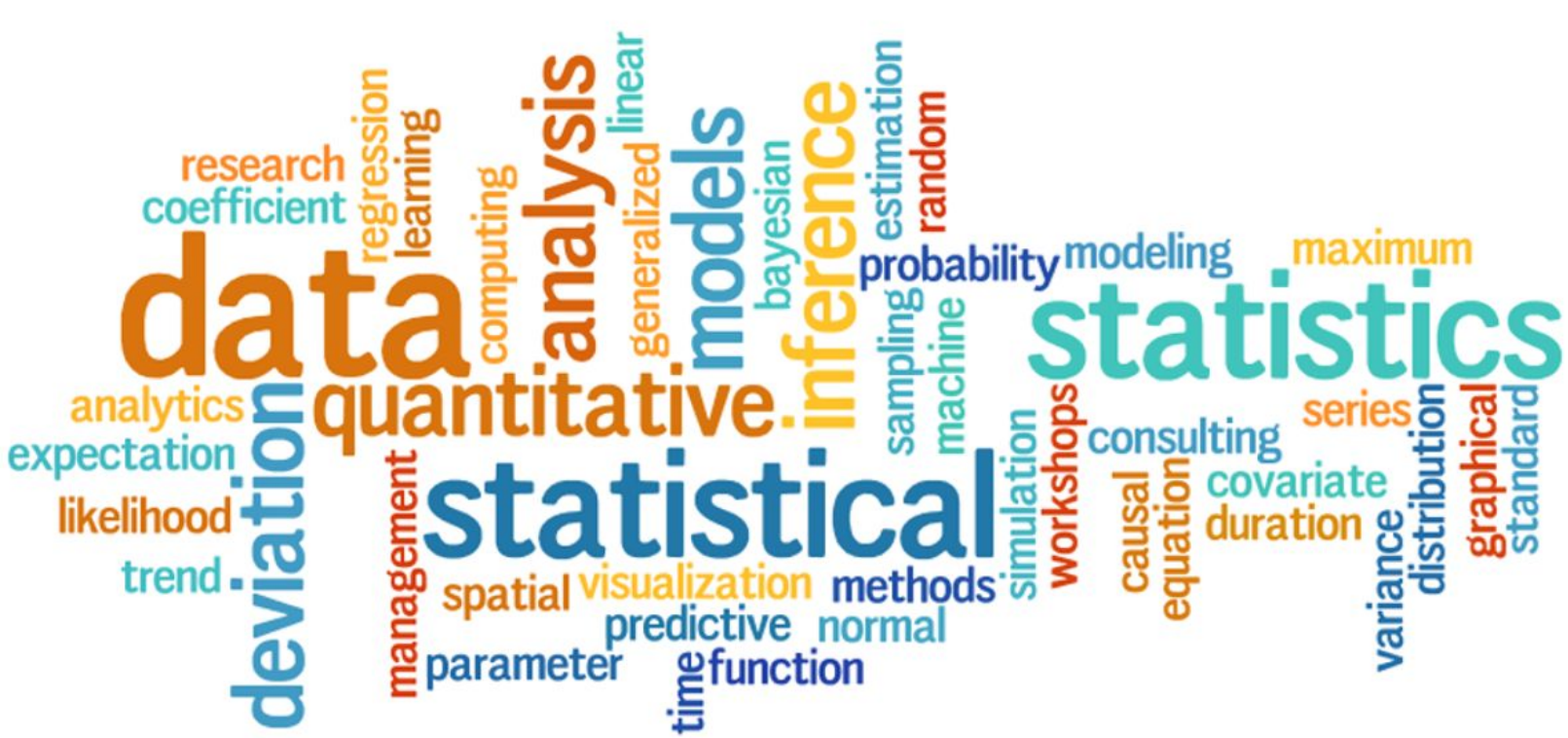
solarwinds

Analyze! It's easy. You learned it in college!







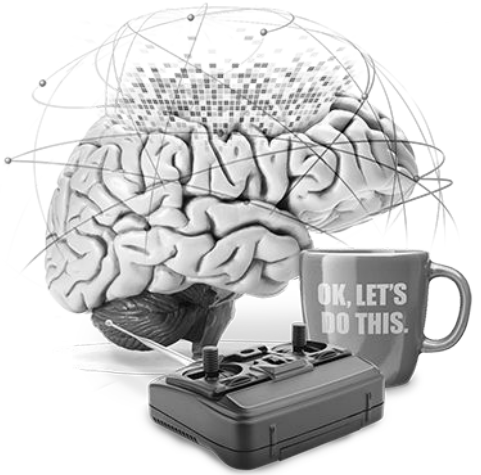


Statistics out of the box

- In addition to providing a method to pull raw data from your system, provide basic statistical analysis.
- Statsd and other agents often provide mean, std dev, and count as part of their data output
- This should be **extended**:
- mean, std dev, **median, quartiles, median absolute deviation, skewness, kurtosis, linear regression trend, cardinality.**

This should be time aware:

Date and time matter in our world, so day of week and hour of day summaries should be stored and updated 168 buckets of data (168 = 24 * 7).



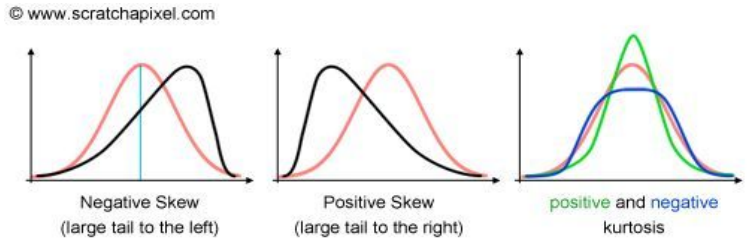
Metric statistics in GoLang coming to a GitHub® near you

solarwinds

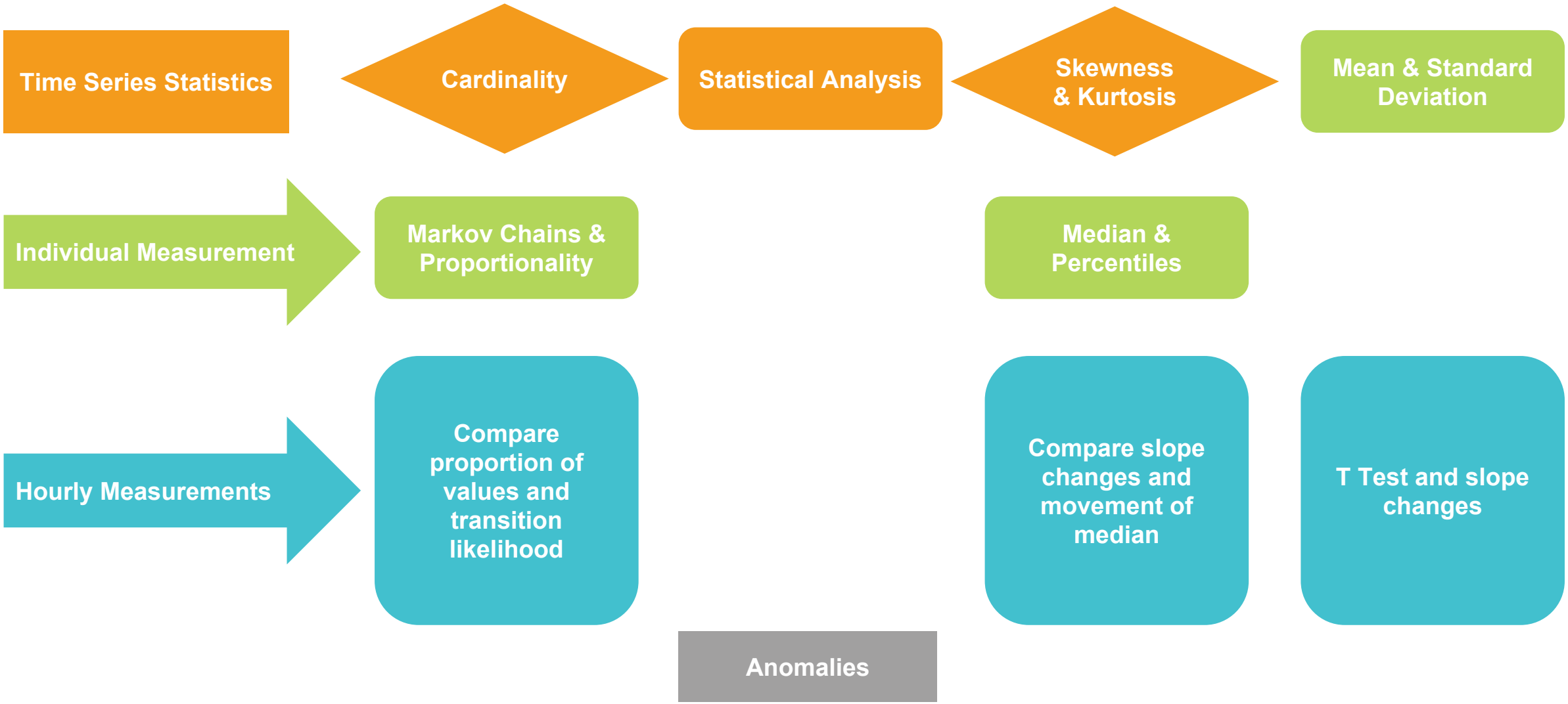
```go
package analytics

// MetricStatistics is the struct to hold all the statisitcs for a metric
type MetricStatistics struct {
    Minimum float64
    Maximum float64
    Mean float64
    StandardDeviation float64
    Variance float64
    Median float64
    MedianAbsoluteDeviation float64
    FirstQuartile float64
    ThirdQuartile float64
    Sum float64
    Slope float64
    SlopeYIntercept float64
    MeanSquaredError float64
    Skewness float64
    Kurtosis float64
    Count int32
    Cardinality int32
    CardinalityRatio int32
}
```

Why these statistics?

- Cardinality - Low means Markov Chains and high means percentiles/statistical.
- Skewness and Kurtosis - Find your strangeness to determine percentile or statistical.
- Standard Deviation and Median Absolute Deviation - Zero is bad in either case and would eliminate either statistical or meaningful low percentile.
- Median vs. Mean - Significant difference between the median and mean shows that percentiles are better.
- Congratulations—you just did some machine learning!



Basic Analytics Decision Tree



What to measure?

The Golden Metrics

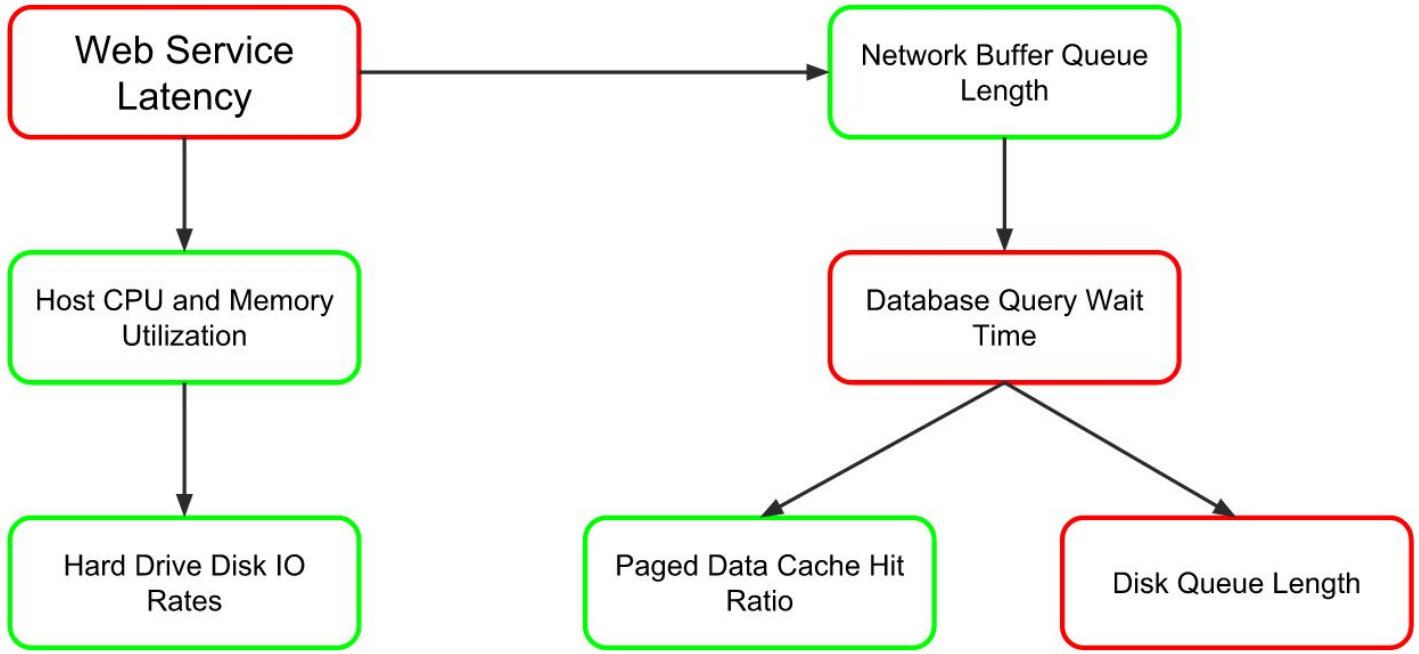
1. Latency - how long operations are taking
2. Traffic - how much an operation is being used
3. Errors - how many times an operation is failing
4. Saturation - amount of capacity an operation is at

- Every metric you add should help answer one of these four categories so that you can track and troubleshoot an issue. (See also RED method, TSA method, and USE method.)
- Rates > Counters - Use counters to create rates. Rates can be compared statistically.

https://landing.google.com/sre/book/chapters/monitoring-distributed-systems.html#xref_monitoring_golden-signals
https://www.circonus.com/2017/08/system-monitoring-with-the-use-dashboard/
http://www.brendangregg.com/tsamethod.html

Relationships out of the box

Many metrics are dependent on other metrics within a system.
Metrics should have a hierarchical relationship mapped out by the developers to show relationships between metrics to help in root cause analysis.



Coding metrics like logging

solarwinds

- Code currently gets instrumented with logging that is enabled by configuration for detail level
- Metrics and performance counters are often always enabled
- Proposal: run statistical calculations at levels similar to logging
  - Don’t spend CPU cycles when everything is okay.
  - Have relationships in place so that when something does go wrong then the analytics are ready.
  - Distributed analytics!
  - Everything a counter?
  - Everything a gauge? Make it a rate.
- Your metric is not important unless you are the top level and you are latency.
  - Assume that you don’t need to do analytics and you should be a lower level.
  - Primary, Secondary, Tertiary - log analogs to Error, Information, Debug.

Save your 168

- I want to store something!
- Store your 168 hourly statistical summaries of your metrics
  - Fixed storage size
  - Predictable IO
- Store your anomalies
- This is a new compact data set!
  - New analytics, clustering, trending, machine learning

The new standard of metrics - REST API example

solarwinds

- /metrics
  - Endpoint which GETs all the names of the metrics in the system
  - Metrics identified and described. e.g., data type, valid range, counter, importance
- /metricsrelationships
  - Get the full hierarchy of metric relationships for the system
- /metricsconfiguration
  - Configurations for sampling rates for statistical analysis and event stream thresholds
- /metrics/{metric_name}
  - Gets the latest analytics including the last n raw data points and the calculated statistics
- /metrics/{metric_name}/relationships
  - Gets all the child and parent relationships for that metric
- /metrics/{metric_name}/{day_of_week}/{hour_of_day}
  - Return statistical analysis for the specified period

Sample call to a named metric

solarwinds

GET http://172.20.220.167/api/metrics/GraphPostDuration

Authorization Headers Body Pre-request Script Tests

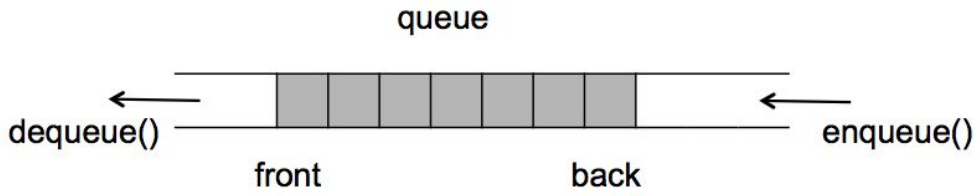
Type No Auth

Body Cookies Headers (4) Test Results

Pretty Raw Preview JSON


```
{
    "minimum": 0,
    "maximum": 147,
    "mean": 5.4666666666666668,
    "standardDeviation": 26.806758454736272,
    "variance": 718.60229885057458,
    "varianceSum": 0,
    "median": 0,
    "medianAbsoluteDeviation": 0,
    "firstQuartile": 0,
    "thirdQuartile": 0,
    "sum": 164,
    "slope": 0.67452725250278089,
    "slopeYIntercept": -4.9885057471264371,
    "meanSquaredError": 25.701415883093841,
    "skewness": 5.16041345134966,
    "kurtosis": 22.903709676331008,
    "count": 30,
    "cardinality": 5,
    "cardinalityRatio": 0.16666666666666666
}
```

Sample Metric - Queue Length



I put my requests into a queue.
Common metric to measure: Queue Length
Issue: Queue Length doesn't tell me how latent my operations are
Solution: Item in Queue Duration metric
Why: Tells me how long an item is in the queue, and therefore how much latency is being added to my operation because of my queue.
Solution 2: Queue and Dequeue Rates
Why: Tells me how many items per unit time my operation can handle, thus telling me a saturation measurement. Answers the question, "When is my operation full, or beyond its capacity?" or "When do I need to add a scaling component to my service?"

The doctor is in!

solarwinds

Health Checks

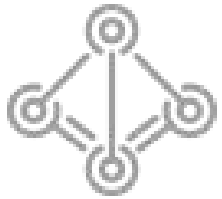
- Give a meaningful health check with a numeric value
- Health = Normalized Latency Health && Normalized Error Health && Normalized Traffic Health && Normalized Saturation Health (Max value of 1 where each gives a value of 0.25 max or 4 values of 1.0)
- Avoid spikes by smoothing the last N measurements so that your health is general for recent history
- https://github.com/kubernetes/community/blob/master/contributors/design-proposals/instrumentation/custom-metrics-api.md

Practical automated examples

Polling frequency - If you notice the latency in a certain device, don't poll it at the default timeout. Give it a little more timeout and save timeout and retransmit expense.

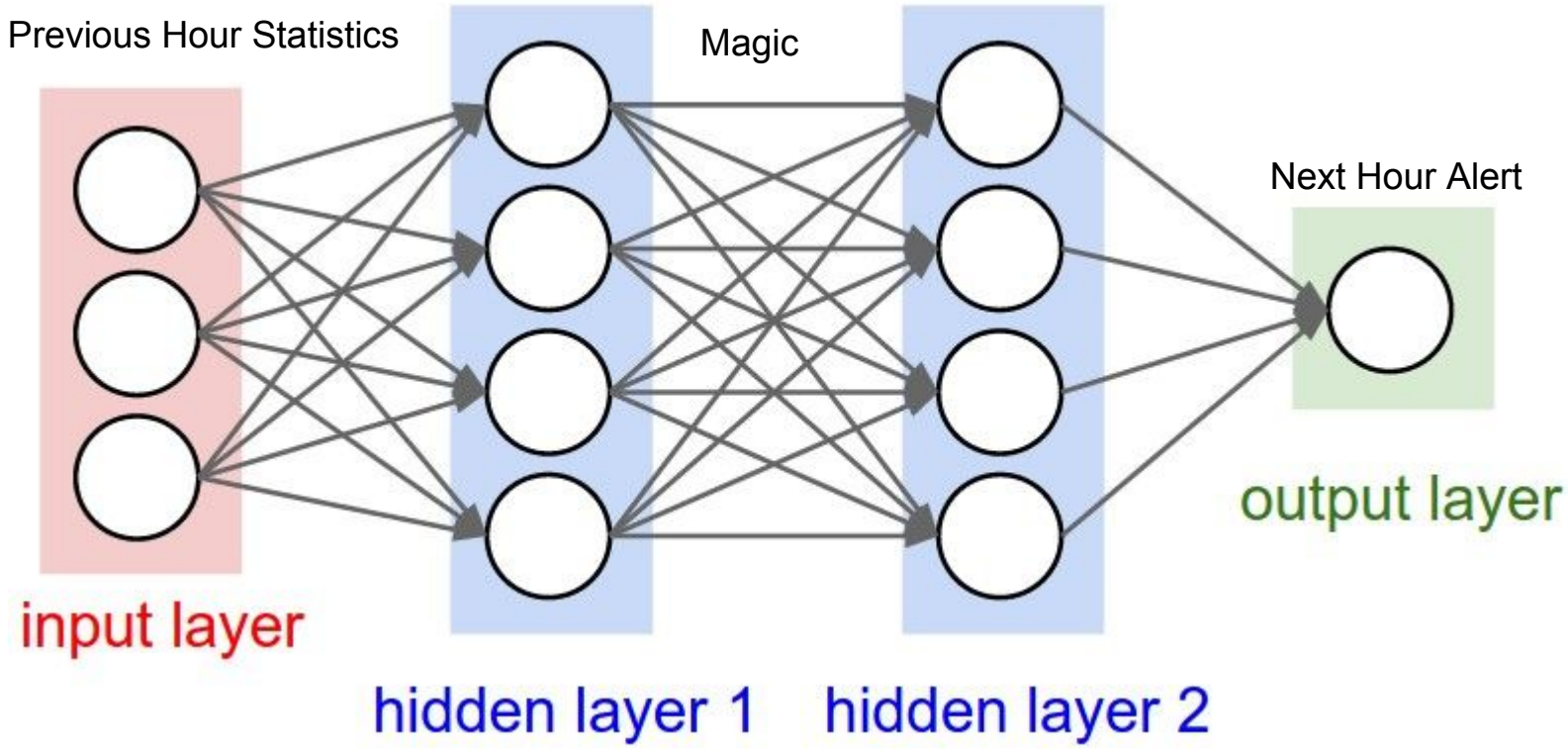
Data size for error - If you notice a pattern where larger or smaller data set inputs creates an error case, then chunk your data and send larger or smaller data input to compensate.

Load balancing - If a certain service responses slower or with more errors, send it less traffic or kill it.



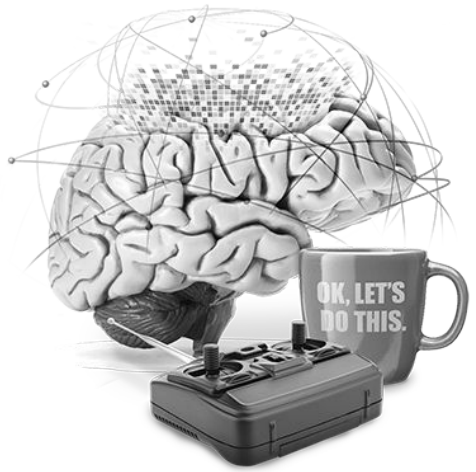
Future applications

Machine learning and deep learning algorithms can use hourly statistical summaries as input for neural networks to look at predicting alerts in the next hour.



Why don’t you just tell me what movie you want to see?

- Event configuration allows for global or metric specific thresholds to allow for a push of events when a system metric is outside of a determined calculation range.
- Monitoring systems then ingest events and query data only when necessary.
  - No need to read 5639 values of 2 from our disk queue length service
  - Stop storing unnecessary data
  - Passive monitoring



Summary

Prepare to be monitored
Instrument knowing how you will be analyzed
Analyze yourself!

Legal stuff that lawyers want me to show