

Front-end Metrics Cheat Sheet

Statsd.net

Collects and aggregates live application measurements. Forwards them on to Graphite.

Types of metrics:

- * **Count** - record an event
- * **Timing** - record how long an event took
- * **Gauge** - report a value at a point in time

Receive metrics on:

- * UDP
- * HTTP
- * TCP

Examples:

LogCount()

myshop.login.attempt- a user attempts to log in
myshop.login.success- the login attempt succeeded
myshop.login.failed- the login attempt failed

This pattern lets you graph the ratio of successes (attempt:success) or failures (attempt:failed)

LogTiming()

myshop.purchase.CreditCard - time in MS to clear transaction

Graphite

Stores data in a time-series database. Provides an easy-to-use front-end for graphing this data.

Feed Graphite from:

- * Statsd / Statsd.net
- * Collectd (for hardware metrics)

Publish data via JSON to

- * Alerting systems (e.g. Nagios, Zabbix)
- * SQL db for further querying
- * BI system for predictive analytics

Graphite Query Primer:

Collapse and sum a series of counts together:
sum(myshop.addToBasket.tshirt.*)

Collapse and sum a subset of metrics:
sum(myshop.addToBasket.tshirt.{small,large})

Find the average of a timing measurement:
avg(myshop.db.fetchProducts.mean)

Summarize up to a larger time window:
summarize(myshop.purchase.CreditCard, "1day")

More Info

Etsy's Legendary Blog Post:

- * <http://tinyurl.com/measure-everything>

People to follow:

- * @Dieter_be
- * @indec
- * @ThomasArdal
- * @librato

Talks:

- * <http://tinyurl.com/not-in-production> - it's not in production unless it's monitored

Github:

- * Statsd.net: <https://github.com/lukevenediger/statsd.net>
- * Graphite: <https://github.com/graphite-project/>
- * JSClient: <https://github.com/lukevenediger/statsdnet-javascript-client/>
- * .Net Client:

Guidance:

- * Metric Anti-patterns: <http://tinyurl.com/metrics-ap>
- * Logging Transactions: <http://tinyurl.com/metrics-tran>

About Luke Venediger:

- * My blog: <http://lukevenediger.me/>
- * Twitter: @lukevenediger