

# Taking Measures

## Jenkins CI User Event – CPH 2014

Robert Sandell – Software Environment and Product Configuration – Sony Mobile Communications

© Sony Mobile Communications

# Background

- We had no easy way of gathering data
- What we had was scattered across different sources.
  - MS SQL, Jenkins Plot Plugin, Mongo, Graphite etc.
- More centralized, easier to log
- All data is not of the same nature
  - Time series vs. event
  - Different data sources needed
- But most data can be aggregated to time series and sent to Graphite



# Disclaimer / Explanation

- We chose to measure externally as much as possible
  - If Jenkins is misbehaving you might not get to the monitoring page to find the data you want.
  - Measuring something new might require plugin update which requires Jenkins restart
- Plugins *(10 min browsing on the wiki)*
  - Disk Usage
  - Build Failure Analyzer
  - Global Build Stats
  - JQS Monitoring
  - Monitoring (Java Melody)
  - Graphite-plugin
  - Dashboard View

# Graphite in short

<http://graphite.readthedocs.org/>

Graphite is an enterprise-scale monitoring tool that runs well on cheap hardware. It was originally designed and written by [Chris Davis](#)

Graphite does two things:

1. Store numeric time-series data
2. Render graphs of this data on demand

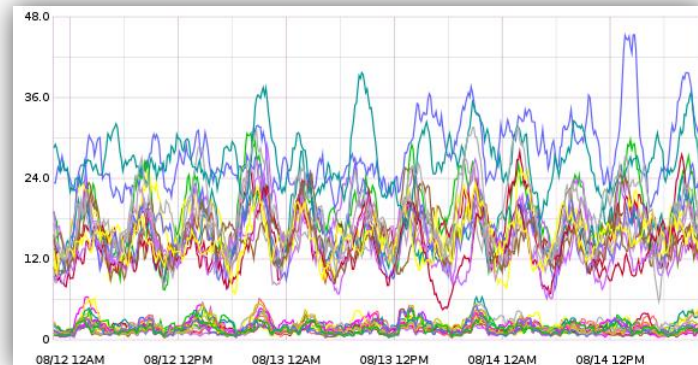
Feeding data into Graphite is simple

```
echo "local.random.diceroll 4 `date +%s`" | nc -q0 graphite.your.com 2003
```

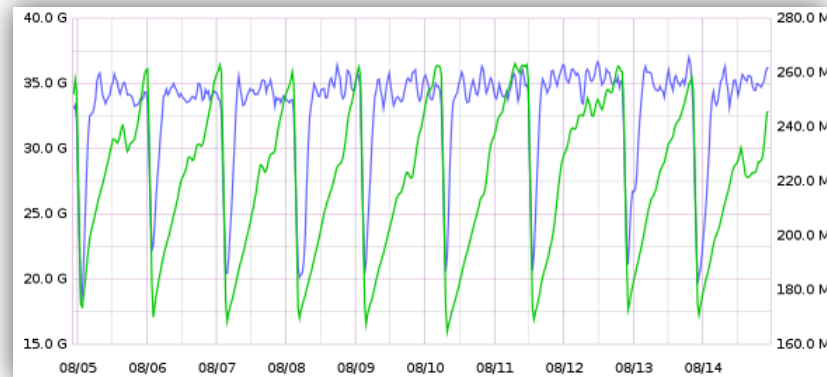
# Collectd - for the fundamentals

<https://collectd.org/>

- collectd is a daemon which collects system performance statistics periodically and provides mechanisms to store the values in a variety of ways, for example in RRD files.
  - Or in our case; sending the data from our masters to a graphite server.
- Graphite can read the RRD format so we are also experimenting with rsync of the local dbs on each slave.
- It has Java and JMX plugins
  - So we can also get data like heap, threads and GC.
  - But also from Tomcat like no. of requests and bytes sent.



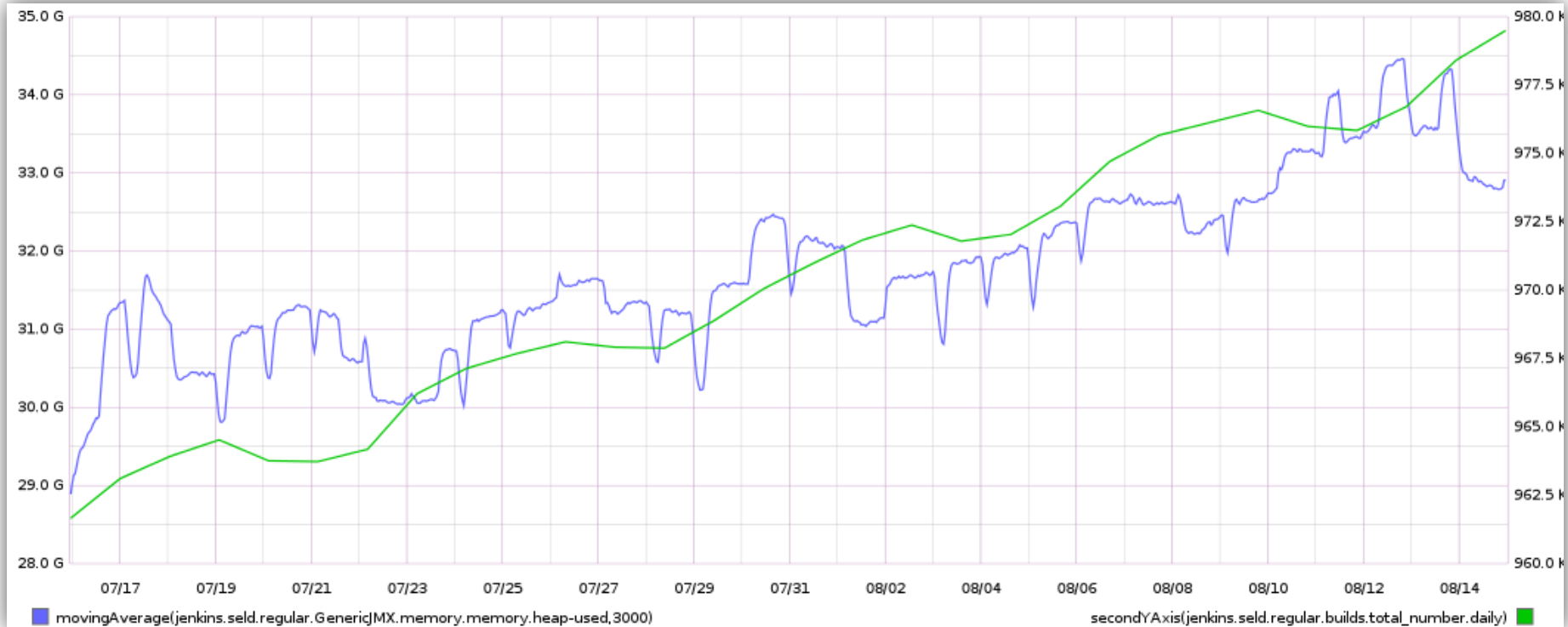
`movingAverage(jenkins.seld.regular.cpu.*.cpu.user,100)`



■ Used Heap  
■ Perm-Gen on secondary axis

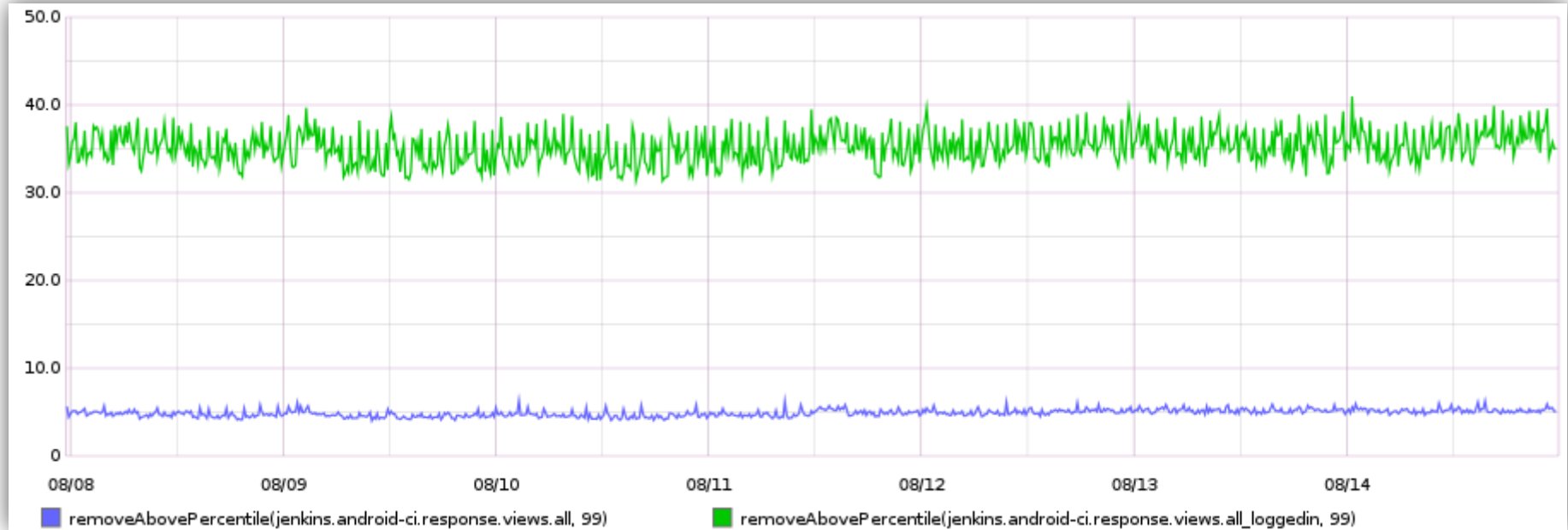


# Heap over a month



# Response time

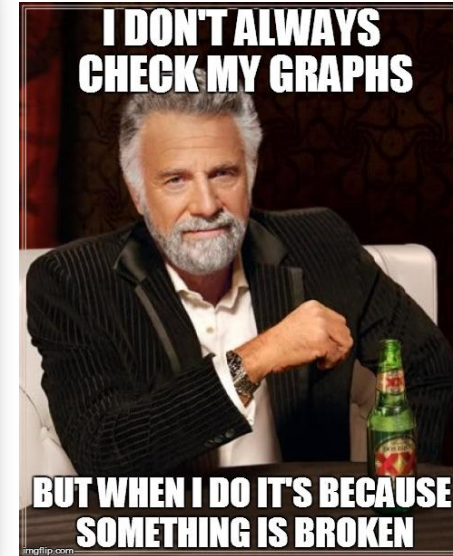
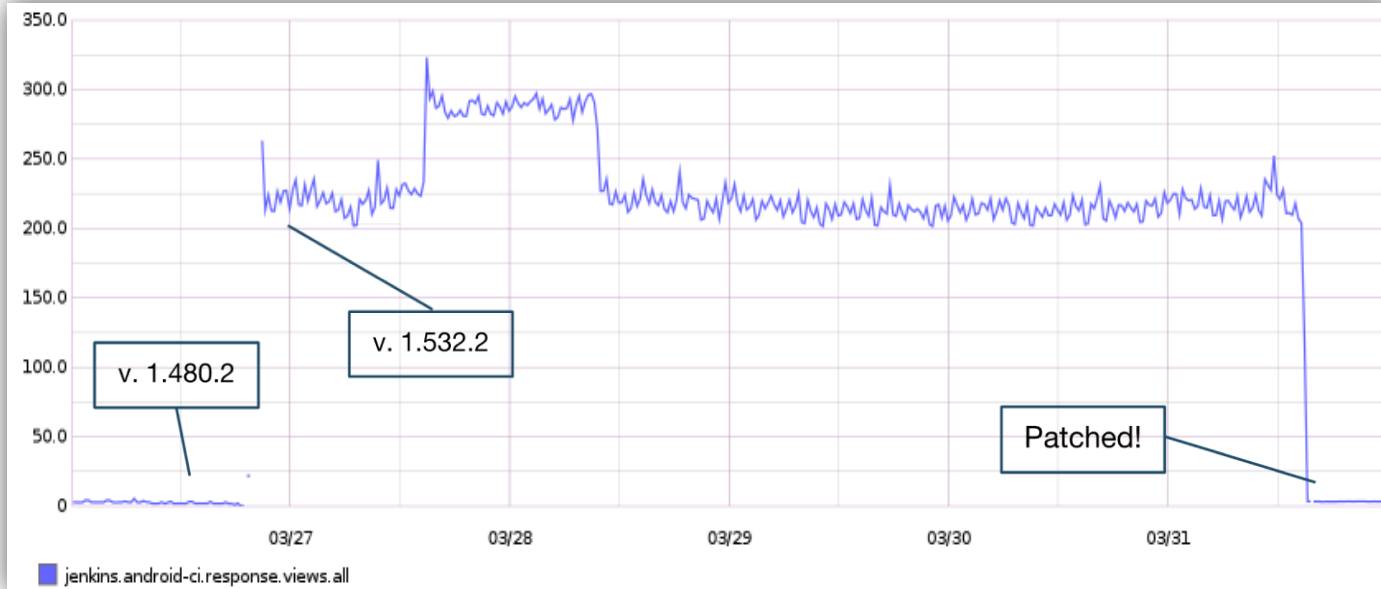
How long does it take to get the all jobs view, using plain old curl



Put this into Nagios and you have the beginnings of an early warning system



# Lesson: You should also measure your staging servers, or this might happen.



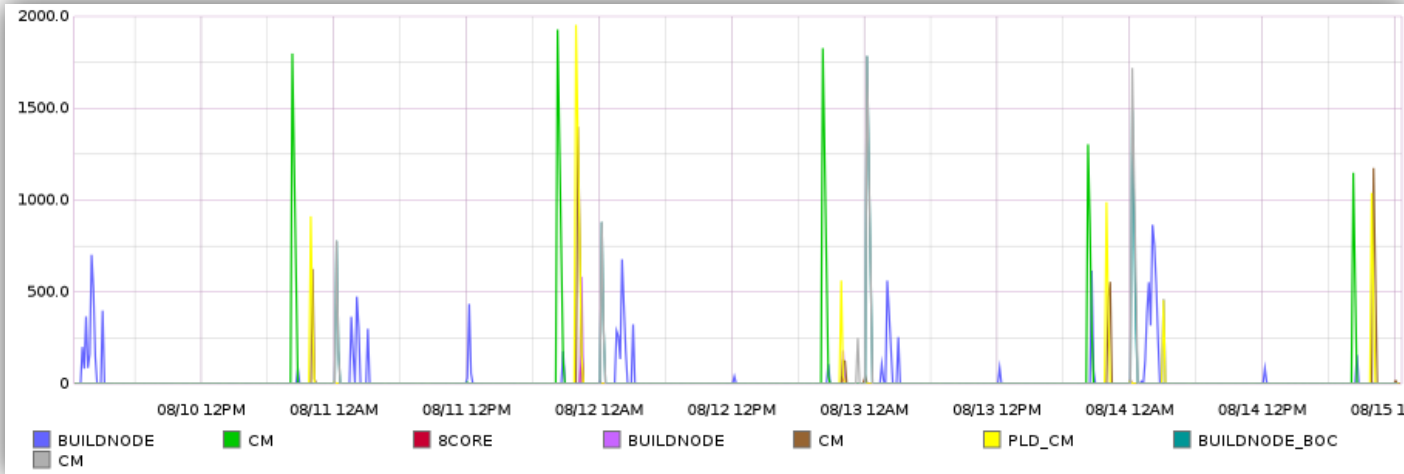


# Queue time

How long does a build stay in the queue before it can start

- One "kicker" job scheduled to run on master every 10 min.
- The kicker starts a parameterized job for each interesting label with a node label parameter so it runs on that label.
- And a "simple" Scriptler build step in the downstream build.

```
int t = findExecutor().getTimeSpentInQueue()
long estStart = System.currentTimeMillis() - t
graphite.println(prefix + LABEL + suffix + " "
    + t/1000 + " " + estStart/1000)
```

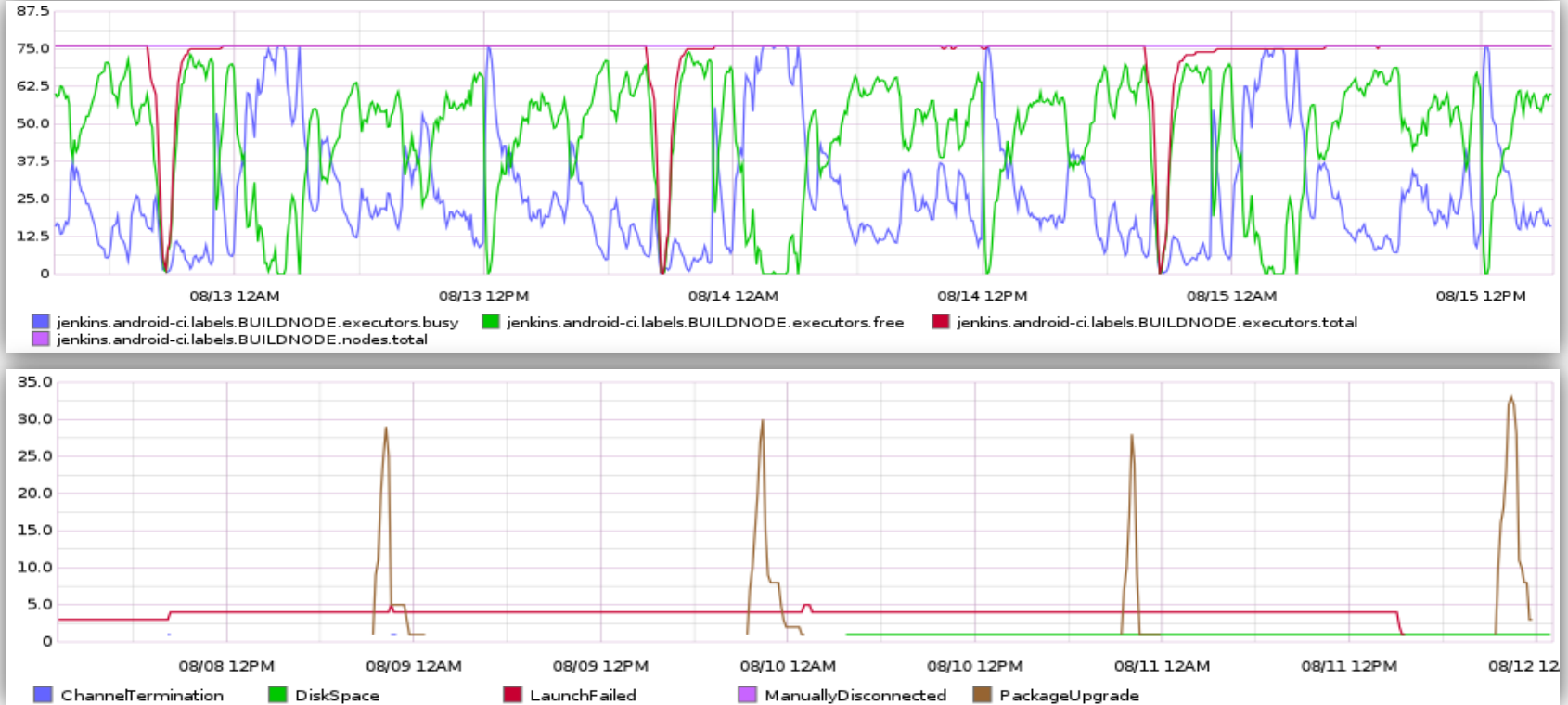


```
aliasByNode(jenkins.android-{ci,ci-platform,ci-cm}.labels.*.queuetime.ten_minutes,3)
```



# Offline/Available Slaves

Enhancement from last year: [https://github.com/rsandell/getting-groovy-with-jenkins/blob/master/src/main/groovy/step2/offline\\_slaves.groovy](https://github.com/rsandell/getting-groovy-with-jenkins/blob/master/src/main/groovy/step2/offline_slaves.groovy)  
Plus <https://github.com/katzijenkins-to-graphite>



# Availability Dashboard

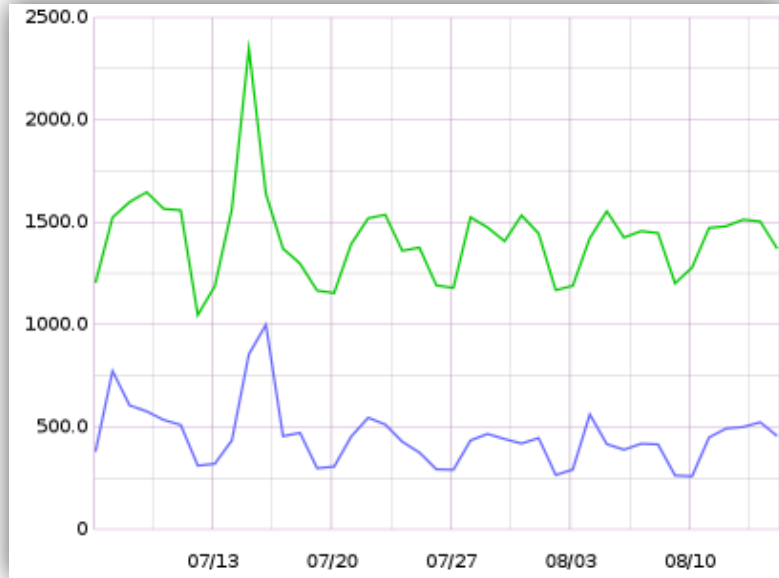
SONY

<http://www.highcharts.com/>

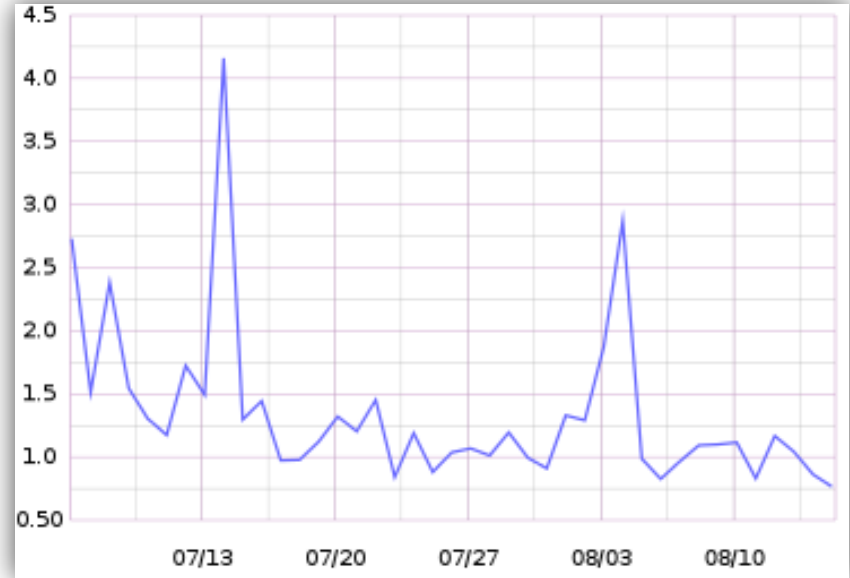


# Build Failure Analyzer – Aggregated data

<https://wiki.jenkins-ci.org/display/JENKINS/Build+Failure+Analyzer>

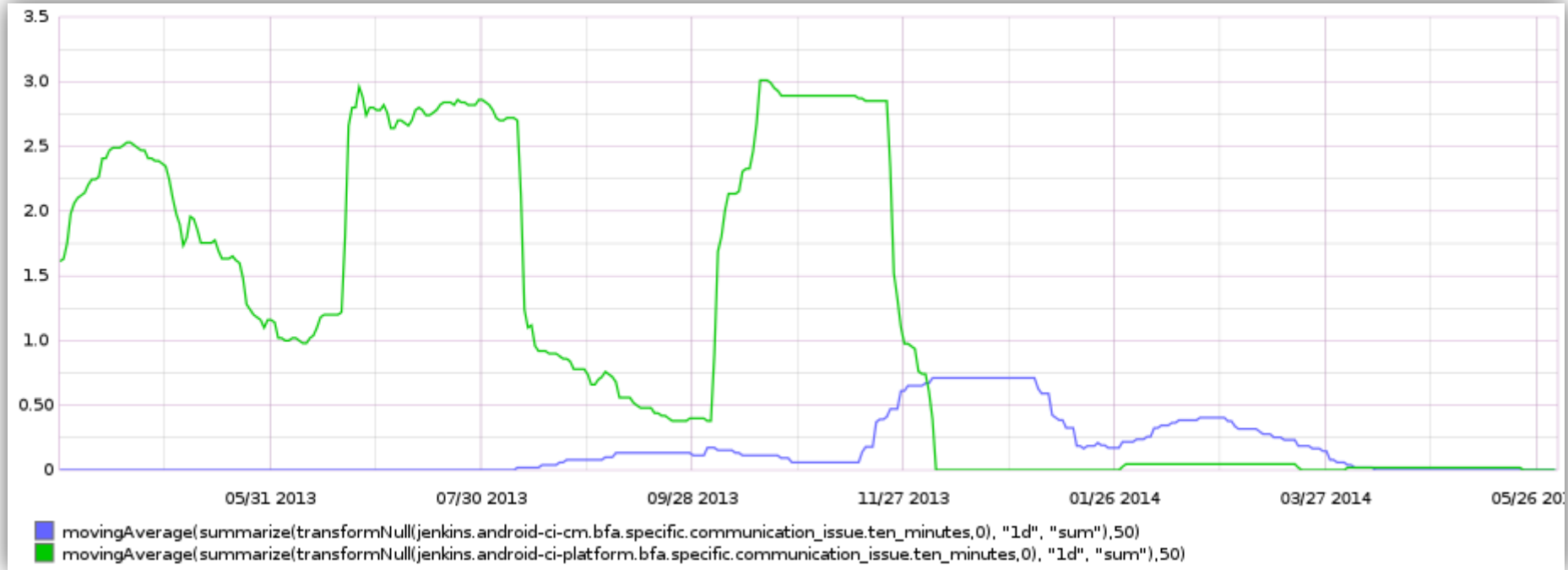


■ Known  
■ Unknown



```
asPercent(  
  summarize(sumSeries(transformNull(jenkins.android-{ci,ci-platform,ci-cm}.bfa.categories.environment.hourly,0)),"1d"),  
  summarize(sumSeries(jenkins.android-{ci,ci-platform,ci-cm}.builds.started.total.hourly),"1d")  
)
```

# BFA: Specific Issues



# Conclusion

## MEASURE



- Good investigative tool
- Start with the low hanging fruit
  - Gives you more ideas for future data gathering.
  - Could turn out to be fun
- A dashboard a day keeps the manager away

