WOMBAT

MONITORING ERLANG SYSTEMS WITH WOMBAT OAM

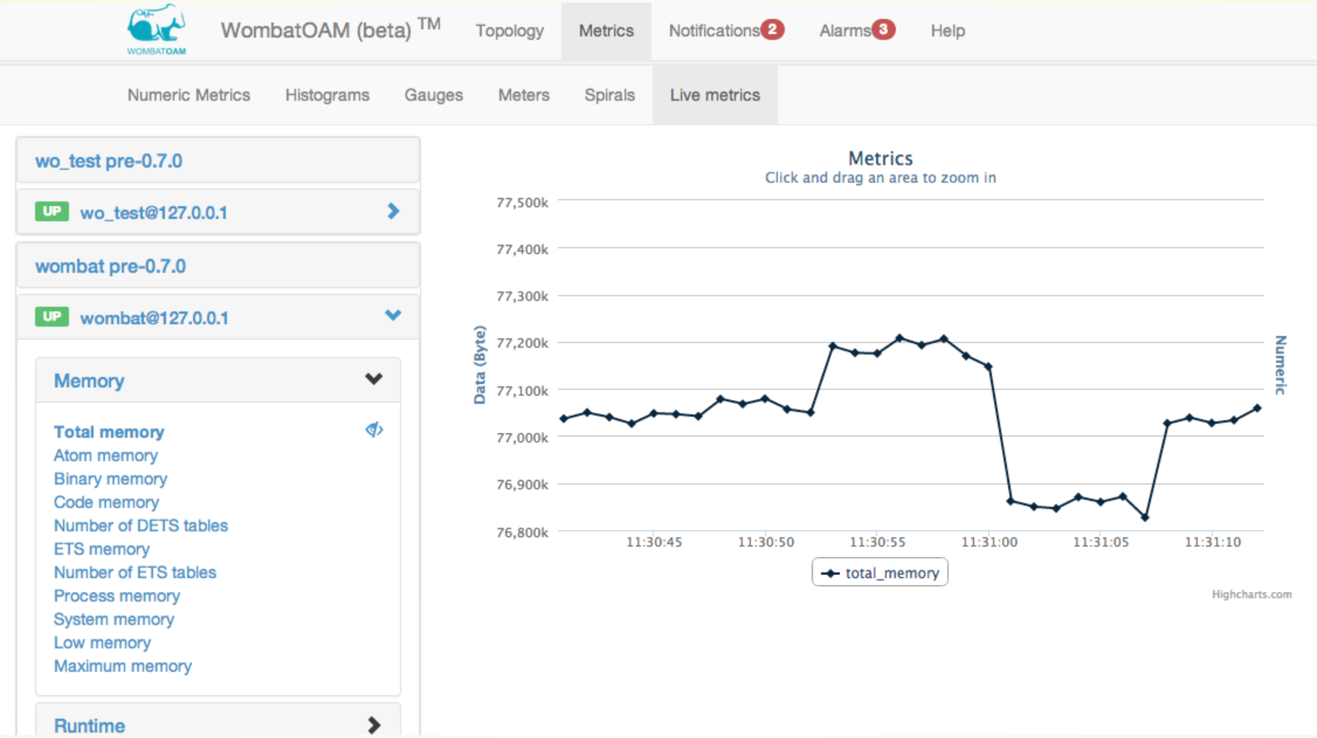
Richard Jonas

Erlang Solutions Ltd.
Budapest

WOMBAT IN WORDS

> *WombatOAM is an operations and maintenance framework for Erlang based systems. It gives you full visibility on what is going on in your Erlang clusters either as a stand-alone product or by integrating into your existing OAM infrastructure.*

WOMBAT IN PICTURES



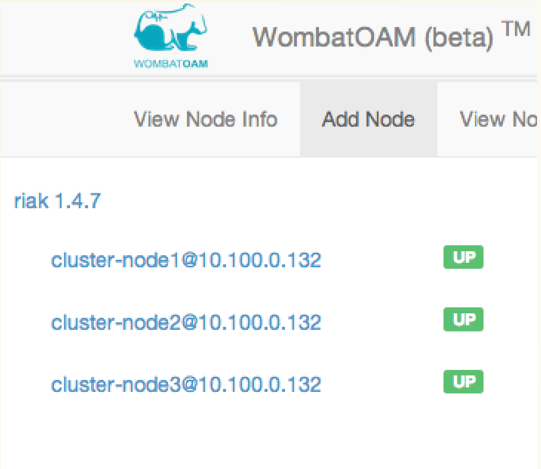
WOMBAT IN DETAILS

- Monitor managed nodes liveliness
- Group managed nodes by Erlang releases
- Gather metrics from different sources, show them in graphs
- Capture logs, show error and crash logs promptly
- Show alarms raised by different applications in managed nodes
- Deploy Erlang releases in the cloud

WHY?

- There is a lot of homegrown solutions for monitoring and management with all of their disadvantages
- Nagios has Erlang plugin but it covers only the very basic functionality (app check)
- folsom/exometer are libraries, one need to capture the data they collected
- entop shows realtime process info (like reductions) but there is no history or peak analysis

TOPOLOGY - MANAGE NODES



riak 1.4.7

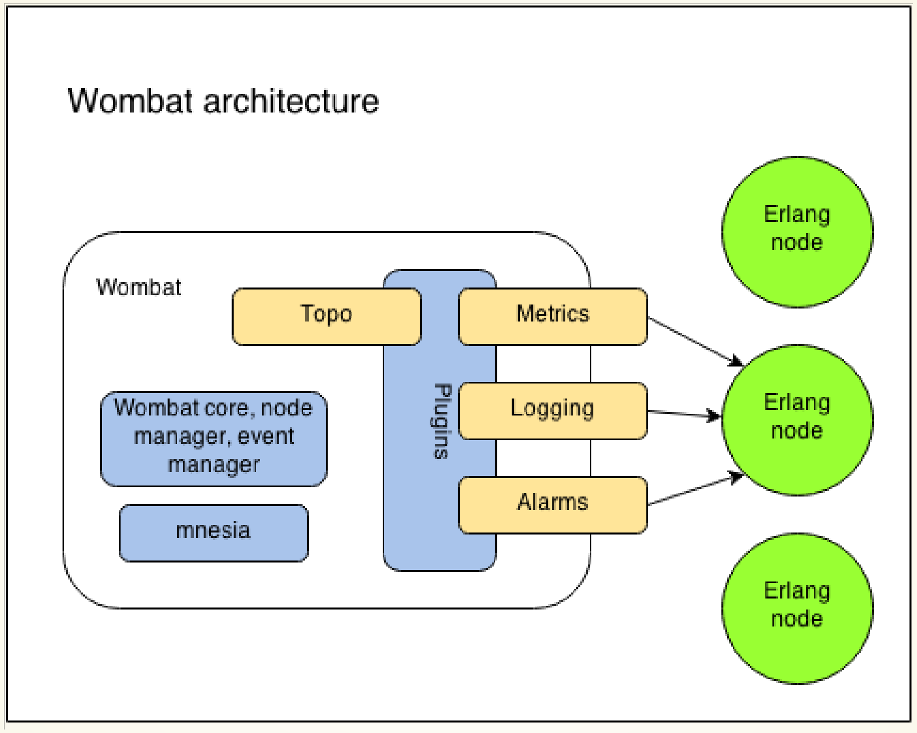
cluster-node1@10.100.0.132 UP

cluster-node2@10.100.0.132 UP

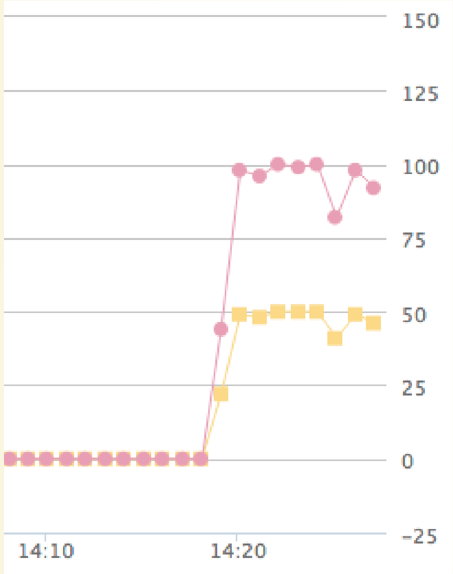
cluster-node3@10.100.0.132 UP

- Group nodes into node families by the Erlang release version
- Show status of the node (up, down, ...)
- Wombat connects as a hidden node
- Add nodes with node discovery

WOMBAT ARCHITECTURE



METRICS - IN DEPTH MONITORING



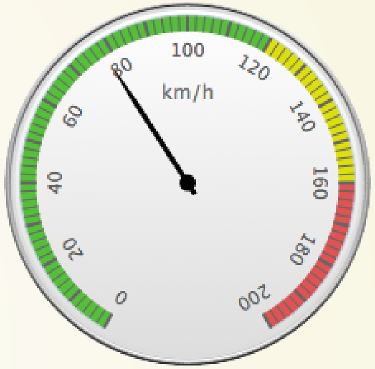
- Metrics agent Erlang module is loaded into managed nodes, Wombat can ask current value of metrics periodically
- Wombat has nearly 100 builtin metrics
- Integrated folsom (e.g. Riak metrics) and exometer support
- Integrated with Graphite

METRICS - CONFIGURATION

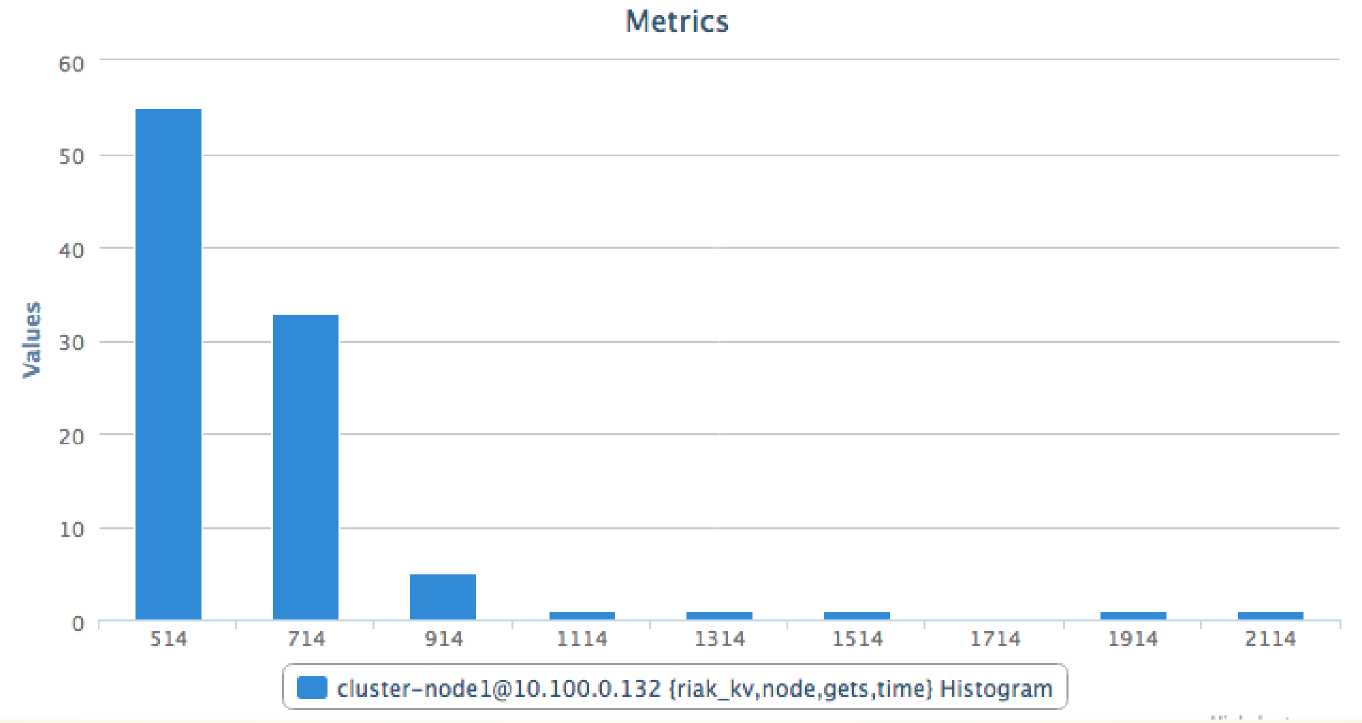
- All metrics are collected periodically
- Metrics are consolidated to show larger periods of time
- `10s:1h 1m:7d 15m:31d`
  - get metrics sample in every 10 seconds, keep them for 1 hour
  - in every minute compute the average of the samples (1st consolidated metrics series) and keep them for a week
  - in every 15 minute compute the average of averages for the previous 15 minute and keep those for a month

METRIC TYPES

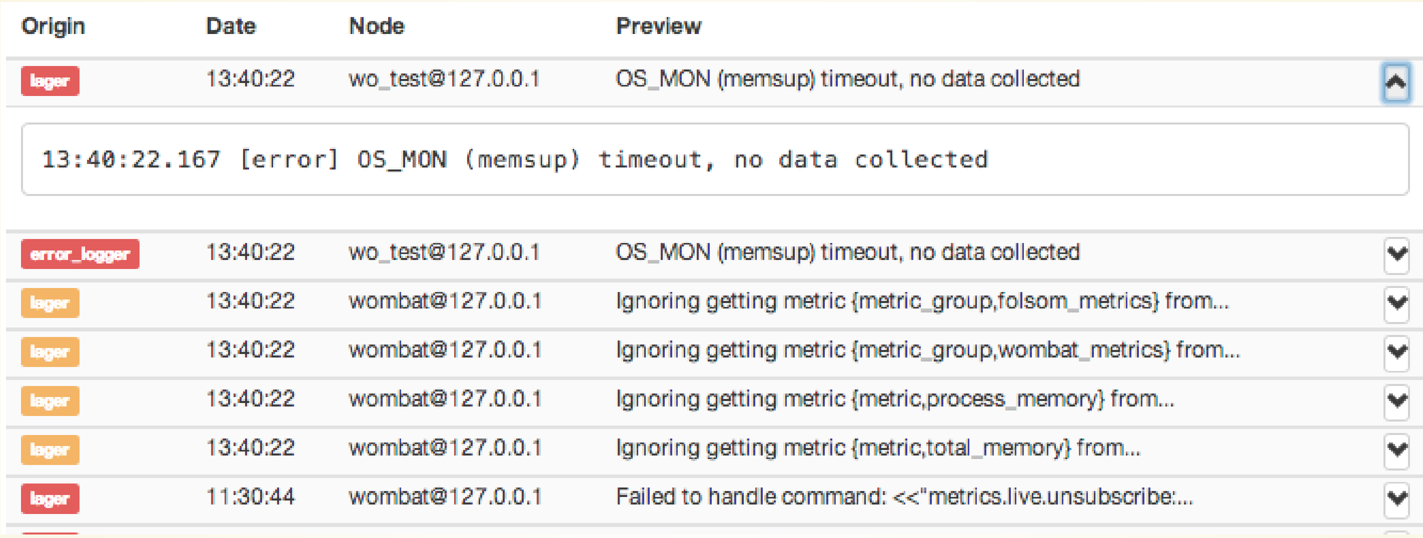
- counters (page impression)
- gauges (number of online users)
- spirals (last minute and total value)
- meters (one, five and fifteen minute moving averages)
- histograms (mean, median, 50, 90, 95, 99 percentiles)



HISTOGRAM


Metrics
Values
60
50
40
30
20
10
0
514
714
914
1114
1314
1514
1714
1914
2114
🟦 cluster-node1@10.100.0.132 {riak_kv,node,gets,time} Histogram

NOTIFICATIONS



Origin	Date	Node	Preview
lager	13:40:22	wo_test@127.0.0.1	OS_MON (memsup) timeout, no data collected

```
13:40:22.167 [error] OS_MON (memsup) timeout, no data collected
```

Origin	Date	Node	Preview
error_logger	13:40:22	wo_test@127.0.0.1	OS_MON (memsup) timeout, no data collected
lager	13:40:22	wombat@127.0.0.1	Ignoring getting metric {metric_group,folsom_metrics} from...
lager	13:40:22	wombat@127.0.0.1	Ignoring getting metric {metric_group,wombat_metrics} from...
lager	13:40:22	wombat@127.0.0.1	Ignoring getting metric {metric,process_memory} from...
lager	13:40:22	wombat@127.0.0.1	Ignoring getting metric {metric,total_memory} from...
lager	11:30:44	wombat@127.0.0.1	Failed to handle command: <<"metrics.live.unsubscribe:...

Collect logs (error, warning, crash), alarms, shows popup if it is configured

CAPTURE LOGS

- Wombat load remotely log handlers. One handler for `error_logger` and one for `lager` .
- Handlers delegate log messages to Wombat if their log level is high enough. Log level can be set at runtime for each managed node, separatedly.
- Log throttling is implemented, not to pour a ton of logs into Wombat.

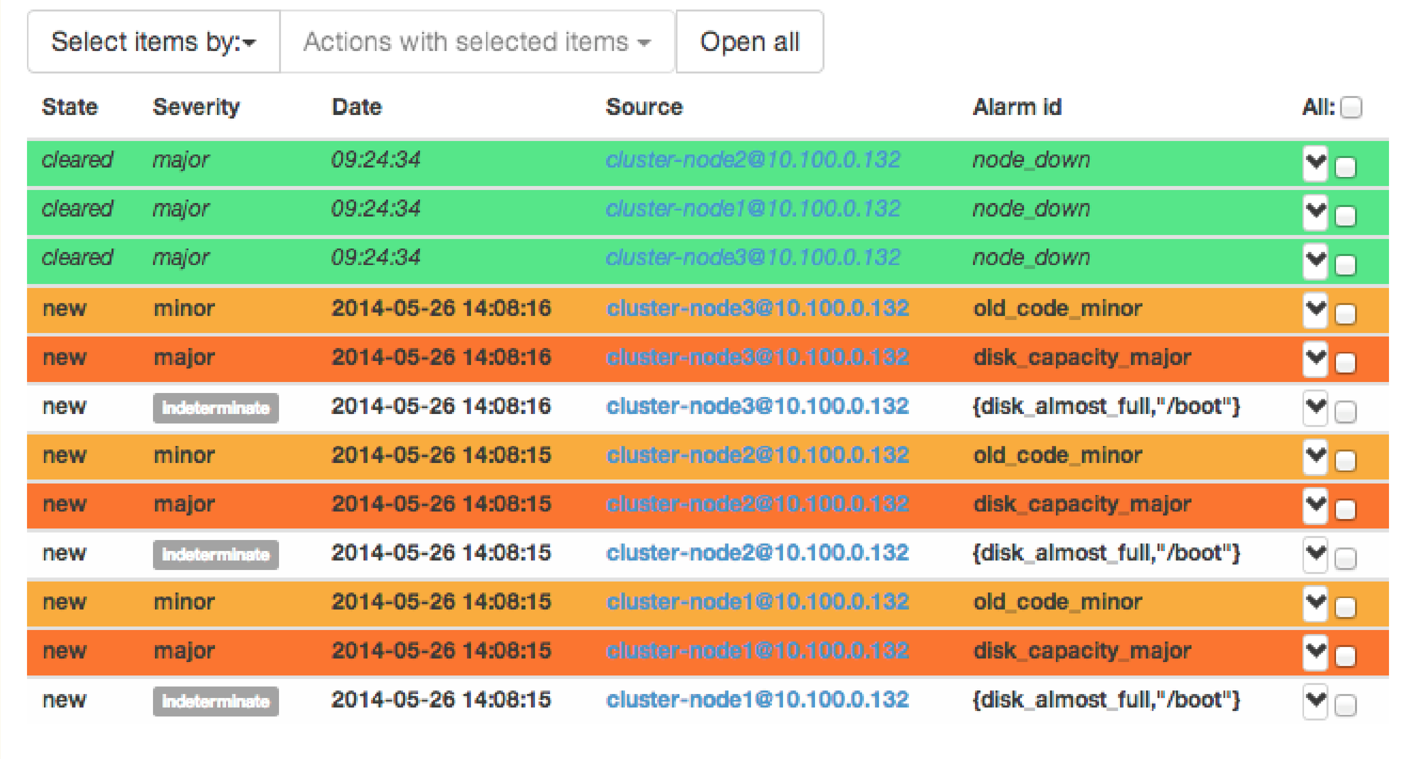
ALARMS

- Wombat subscribes to SASL alarm handler to be notified about alarms raised by the standard framework
- Wombat has 20+ builtin alarms, collected by Wombat alarm agent loaded into the node
- elarm alarms are also collected and managed

BUILT-IN ALARMS

- node is down
- old code is loaded
- process/port/memory/ets/module/atom usage is approaching to its limit
- application/module version difference in a node family
- OS cpu/disk/memory usage is high
- too much open file/port

ALARMS IN ACTION



State	Severity	Date	Source	Alarm id	All: □
cleared	*major*	*09:24:34*	*cluster-node2@10.100.0.132*	*node_down*	▾ □
cleared	*major*	*09:24:34*	*cluster-node1@10.100.0.132*	*node_down*	▾ □
cleared	*major*	*09:24:34*	*cluster-node3@10.100.0.132*	*node_down*	▾ □
new	minor	2014-05-26 14:08:16	cluster-node3@10.100.0.132	old_code_minor	▾ □
new	major	2014-05-26 14:08:16	cluster-node3@10.100.0.132	disk_capacity_major	▾ □
new	indeterminate	2014-05-26 14:08:16	cluster-node3@10.100.0.132	{disk_almost_full,"/boot"}	▾ □
new	minor	2014-05-26 14:08:15	cluster-node2@10.100.0.132	old_code_minor	▾ □
new	major	2014-05-26 14:08:15	cluster-node2@10.100.0.132	disk_capacity_major	▾ □
new	indeterminate	2014-05-26 14:08:15	cluster-node2@10.100.0.132	{disk_almost_full,"/boot"}	▾ □
new	minor	2014-05-26 14:08:15	cluster-node1@10.100.0.132	old_code_minor	▾ □
new	major	2014-05-26 14:08:15	cluster-node1@10.100.0.132	disk_capacity_major	▾ □
new	indeterminate	2014-05-26 14:08:15	cluster-node1@10.100.0.132	{disk_almost_full,"/boot"}	▾ □

DEPLOYMENT

- Register infrastucture providers
- Upload tarball releases
- Define node families
- Deploy applications

What has Wombat ever done for us?

LEAKING PROCESSES

```
%% Spawn some processes
W = fun() -> receive _ -> ok end end.
P = [spawn(W) || _ <- lists:seq(1, 1000)].

%% Stop them
[Pid ! stop || Pid <- P].
```

LEAKING MESSAGES

```
[self() ! ok || _ <- lists:seq(1, 10000)].

flush().
```

SHOW CRASHES

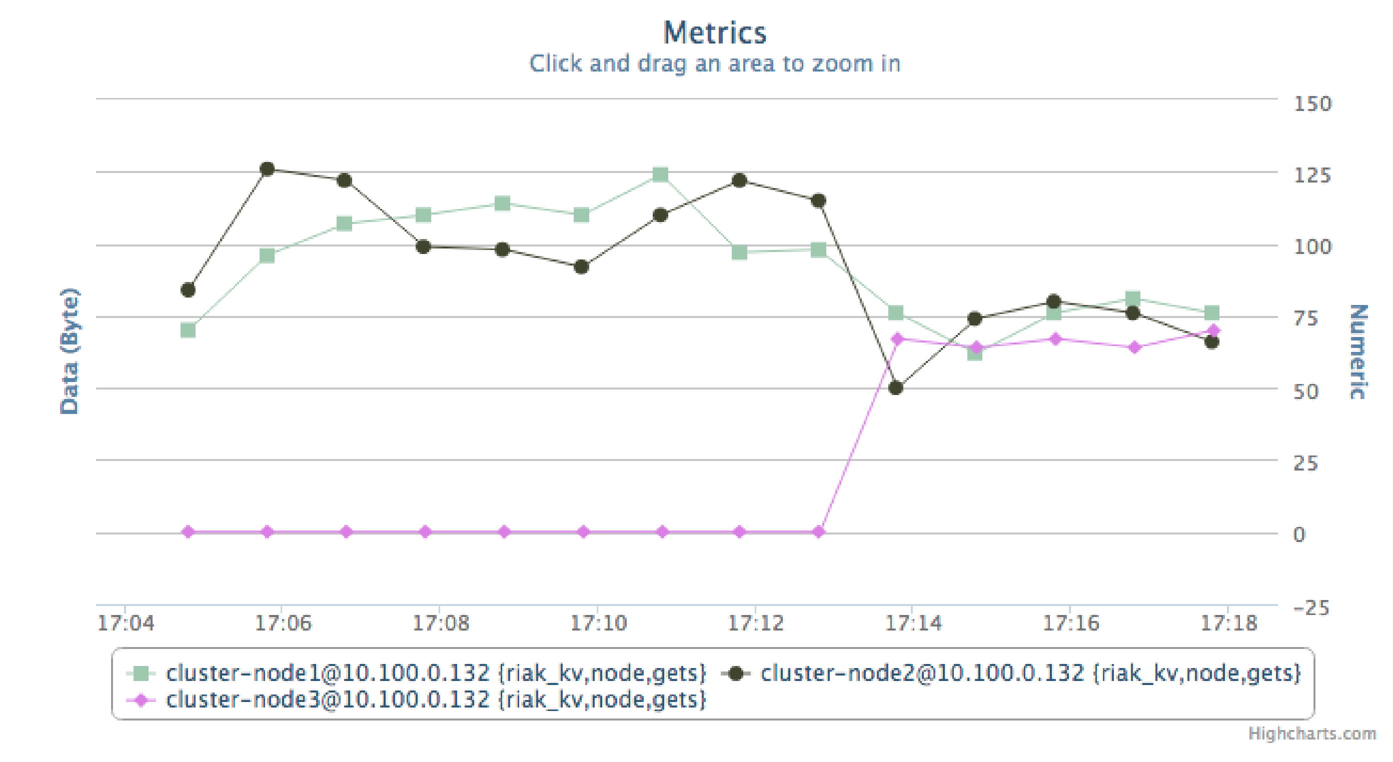
```
%% Start a supervisor with can have child servers
%% and start a new child
wo_test_crash_sup:new_child().

%% Crash that child to simulate
%% Wombat crash log capture capabilty
wo_test_crash_srv:please_crash().
```

DETECTING OLD CODE

```
l(wo_test_crash_srv).
```

UNEVEN LOAD-BALANCING



FUTURE PLANS

CONTACT DETAILS

http://erlang-solutions.com

http://erlangcentral.org