WOMBAT

MONITORING ERLANG SYSTEMS WITH WOMBAT OAM

Richard Jonas

Erlang Solutions Ltd.
Budapest
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**

- 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
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)
NOTIFICATIONS

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, separately.
- 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

<table>
<thead>
<tr>
<th>State</th>
<th>Severity</th>
<th>Date</th>
<th>Source</th>
<th>Alarm id</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>cleared</td>
<td>major</td>
<td>09:24:34</td>
<td>cluster-node2@10.100.0.132</td>
<td>node_down</td>
<td></td>
</tr>
<tr>
<td>cleared</td>
<td>major</td>
<td>09:24:34</td>
<td>cluster-node1@10.100.0.132</td>
<td>node_down</td>
<td></td>
</tr>
<tr>
<td>cleared</td>
<td>major</td>
<td>09:24:34</td>
<td>cluster-node3@10.100.0.132</td>
<td>node_down</td>
<td></td>
</tr>
<tr>
<td>new</td>
<td>minor</td>
<td>2014-05-26 14:08:16</td>
<td>cluster-node3@10.100.0.132</td>
<td>old_code_minor</td>
<td></td>
</tr>
<tr>
<td>new</td>
<td>major</td>
<td>2014-05-26 14:08:16</td>
<td>cluster-node3@10.100.0.132</td>
<td>disk_capacity_major</td>
<td></td>
</tr>
<tr>
<td>new</td>
<td>indeterminate</td>
<td>2014-05-26 14:08:16</td>
<td>cluster-node3@10.100.0.132</td>
<td>[disk_almost_full,&quot;/boot&quot;]</td>
<td></td>
</tr>
<tr>
<td>new</td>
<td>minor</td>
<td>2014-05-26 14:08:15</td>
<td>cluster-node2@10.100.0.132</td>
<td>old_code_minor</td>
<td></td>
</tr>
<tr>
<td>new</td>
<td>major</td>
<td>2014-05-26 14:08:15</td>
<td>cluster-node2@10.100.0.132</td>
<td>disk_capacity_major</td>
<td></td>
</tr>
<tr>
<td>new</td>
<td>indeterminate</td>
<td>2014-05-26 14:08:15</td>
<td>cluster-node2@10.100.0.132</td>
<td>[disk_almost_full,&quot;/boot&quot;]</td>
<td></td>
</tr>
<tr>
<td>new</td>
<td>minor</td>
<td>2014-05-26 14:08:15</td>
<td>cluster-node1@10.100.0.132</td>
<td>old_code_minor</td>
<td></td>
</tr>
<tr>
<td>new</td>
<td>major</td>
<td>2014-05-26 14:08:15</td>
<td>cluster-node1@10.100.0.132</td>
<td>disk_capacity_major</td>
<td></td>
</tr>
<tr>
<td>new</td>
<td>indeterminate</td>
<td>2014-05-26 14:08:15</td>
<td>cluster-node1@10.100.0.132</td>
<td>[disk_almost_full,&quot;/boot&quot;]</td>
<td></td>
</tr>
</tbody>
</table>
DEPLOYMENT

- Register infrastructure providers
- Upload tarball releases
- Define node families
- Deploy applications
What has Wombat ever done for us?
%% Spawn some processes
\[ W = \text{fun()} \rightarrow \text{receive } \_ \rightarrow \text{ok end end.} \]
\[ P = [\text{spawn}(W) \mid \_ \leftarrow \text{lists:seq(1, 1000)}]. \]

%% Stop them
\[ [\text{Pid} ! \text{stop} \mid \mid \text{Pid} \leftarrow P]. \]
LEAKING MESSAGES

```prolog
[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 capability
wo_test_crash_srv:please_crash().
DETECTING OLD CODE

1(wo_test_crash_srv).
UNEVEN LOAD-BALANCING
FUTURE PLANS

WHATEVER YOU DO ALWAYS GIVE 100% UNLESS YOU'RE DONATING BLOOD

VIA 9GAG.COM
CONTACT DETAILS

http://erlang-solutions.com
http://erlangcentral.org