SyncFree Technology White Paper

Configuring Distributed Erlang Systems with WombatOAM

Viktória Fördős
Erlang Solutions Ltd.

12 August 2016

Why WombatOAM?

CRDT (Conflict-free replicated data type) [7, 8] is a data type that supports conflict free resolution of concurrent, distributed updates. It is often mentioned alongside storage systems that are distributed, fault-tolerant and reliable. These are similar properties and features of Erlang/OTP systems.

What distributed Erlang/OTP systems lack, however, is a standardised way to configure multiple nodes. In order to customise Erlang nodes' behaviour, OTP middleware allows you to set configuration parameters using configuration files on a node basis, they can be updated at runtime, but will not survive a restart unless persisted in the business logic of the system. There is no widely adopted solution to address this omission.

To avoid system restarts, DevOps teams often change the configuration at runtime without updating the configuration files. This implies that the current configuration of a node cannot be determined solely on the configuration files, generating uncertainty when troubleshooting a system. This is a known problem which becomes critical in large scale, distributed Erlang systems.

In conclusion, configuring distributed Erlang systems with no down time is today a though challenge. This is a real issue as Erlang drives the telecom industry, large betting portals, messaging providers, market leading payment gateways, gaming engines and online ad services. A step towards harmonising configuration management in distributed Erlang systems would be welcomed by the industry.

Related work

Configuration management where network connectivity is unreliable and clusters span across data centers is today often limited to changing the static values listed in configuration files that are created and deployed at startup. Tools, such as CFEngine [2], BCFG2 [4], Zookeeper [6], Chef[1] and Puppet[2], all help, but come with their limitations and do not provide an Erlang centric approach to the problem. They are very generic and work only with configuration files on machine level, hence, they can only check and change the initial configurations listed in the static configuration files. The current configuration of the node cannot be retrieved or changed once deployed. Besides that, these tools expect a happy path, and rarely take network failures and consistency into consideration. As a result, issues have been known to cause outages and have left the system in an inconsistent state, with no automated means of detecting the root cause of the problem.

After studying the existing approaches and industrial best practices, we found the need for a tool that provides visibility, allows only syntactically valid changes, tracks configuration history and monitors the consistency across nodes. We committed ourselves to research a solution which would primarily focus on handling current configuration in Erlang nodes. Our solution does not compete with the general tools, it complements them by providing an Erlang centric approach to configuration management.

Erlang specific configuration management

Our results were integrated into WombatOAM[3] [3], an operation and maintenance tool for systems written in Erlang, Elixir and other programming languages running on the Beam virtual machine. Using WombatOAM's Web Dashboard, the Erlang specific configuration management [5] is available to users. Managing environment variables and automatic system checks ensuring the consistency of configuration data on both node and cluster-levels are all provided by WombatOAM. Inconsistencies and failed updates are detected and reported in the form of an alarms, and the history and status of all performed changes are logged, facilitating troubleshooting and recovery efforts. Figure 1 shows the result of changing a configuration parameter clusterwide on all nodes using the WombatOAM Web Dashboard.

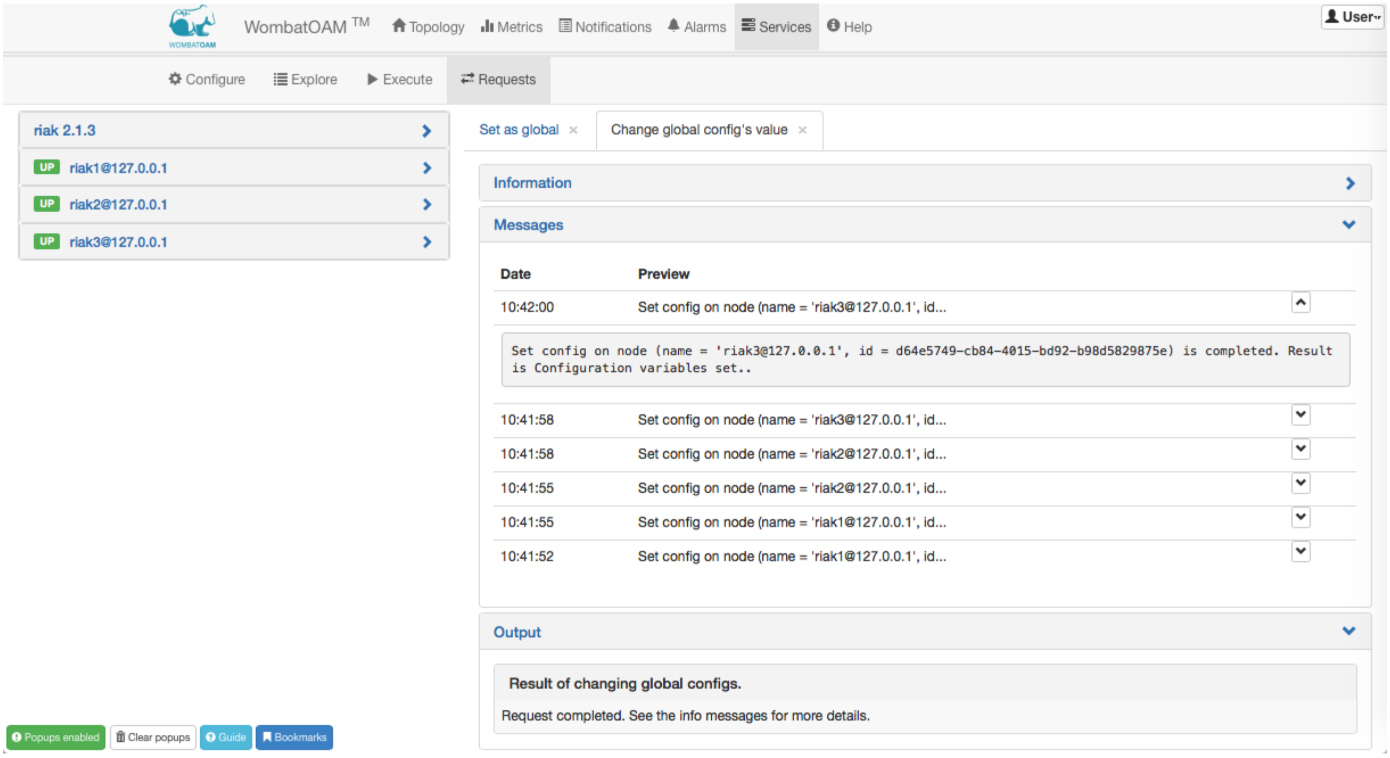
Behind the scenes

Our configuration management approach is designed for distributed Erlang/OTP systems, which are systems where network issues are likely to happen. Ergo, the challenge is how to guarantee that configuration changes will eventually be completed and persisted, even in the presence of errors. Operation based CRDTs and causal consistency can be an enabler for successfully carrying out operations in large scale, distributed Erlang systems. Hence, WombatOAM's configuration management feature builds on the

[1] https://www.chef.io/solutions/ configuration-management/

[2] https://puppet.com/solutions/configuration-management

[3] https://www.erlang-solutions.com/products/wombat-oam.html



Date	Preview
10:42:00	Set config on node (name = 'riak3@127.0.0.1', id...

```
Set config on node (name = 'riak3@127.0.0.1', id = d64e5749-cb84-4015-bd92-b98d5829875e) is completed. Result
is Configuration variables set..
```

Date	Preview
10:41:58	Set config on node (name = 'riak3@127.0.0.1', id...
10:41:58	Set config on node (name = 'riak2@127.0.0.1', id...
10:41:55	Set config on node (name = 'riak2@127.0.0.1', id...
10:41:55	Set config on node (name = 'riak1@127.0.0.1', id...
10:41:52	Set config on node (name = 'riak1@127.0.0.1', id...

Result of changing global configs.

Request completed. See the info messages for more details.

Figure 1: Change a configuration cluster-wide

CRDTs and the consistency model provided by Antidote [1].

A stronger guarantee in an unreliable network can be provided by removing synchronisation and centralisation, thus, the configuration updates can be carried out in an asynchronous and decentralised manner. The key idea is to separate two functionalities of server nodes: (1) introducing new configuration and (2) controlling consistent update on the managed nodes. That is the WombatOAM Server node remains the point of introducing new configurations to the system (and a single point of control for users), while the "controlling" functionality that manages consistent configuration update on the managed nodes moves to an intermediate layer, which is an Antidote cluster in this case that consists of a set of Antidote nodes. Using the intermediate layer the Node Configurator Plugins ensure that all connected and re-joining nodes are updated with the latest configuration. Figure ?? shows an overview depicting the main components of the CRDT based approach.

As a result, the CRDT based approach is less sensitive to network issues, and can help in managing systems with high availability by minimising propagation failures caused by network issues.

References

[1] Deepthi Devaki Akkoorath, Alejandro Z. Tomsic, Manuel Bravo, Zhongmiao Li, Tyler Crain, Annette Bieniusa, Nuno Preguiça, and Marc Shapiro. Cure: Strong semantics meets high availability and low latency. In *International Conference on Distributed Computing Systems (ICDCS 2016)*, Nara, Japan, June 2016. IEEE.

[2] Mark Burgess. On the theory of system administration. *Science of Computer Programming*, 49(1):1 – 46, 2003.

[3] Natalia Chechina, Kenneth MacKenzie, Simon Thompson, Phil Trinder, Olivier Boudeville, Viktória Fördős, Amir Ghaffari, Csaba Hoch, and Mario Moro Hernandez. Evaluating Scalable Distributed Erlang for Scalability and Reliability. *Submitted to IEEE Transactions on Parallel and Distributed Systems*, April 2016.

[4] Narayan Desai, Rick Bradshaw, Joey Hagedorn, and Cory Lueninghoener. Directing change using bcfg2. In *Proceedings of 20th Large Installation System Administration Conference*, LISA'06, dec 2006.

[5] Viktória Fördős and Francesco Cesarini. CRDTs for the Configuration of Distributed Erlang Systems. In *Proceedings of the 15th ACM SIGPLAN Erlang Workshop*, Erlang'16, Nara, Japan, September 2016. ACM.

[6] Patrick Hunt, Mahadev Konar, Flavio P. Junqueira, and Benjamin Reed. Zookeeper: Wait-free coordination for internet-scale systems. In *usenix-atc*, pages 11–11, Boston, MA, USA, 2010.

[7] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. A comprehensive study of Convergent and Commutative Replicated Data Types. Research Report RR-7506, Inria – Centre Paris-Rocquencourt ; INRIA, January 2011.

[8] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. Conflict-free replicated data types. In *Proceedings of the 13th International Conference on Stabilization, Safety, and Security of Distributed Systems*, SSS'11, pages 386–400, Berlin, Heidelberg, 2011. Springer-Verlag.