Geophysical Research Abstracts
Vol. 14, EGU2012-11323, 2012
EGU General Assembly 2012

An Alternative for In-Depth Monitoring of Tsunami Early Warning Systems

L. Lindner, S. Gensch, R. Henneberger, M. Lendholt, and M. Hammitzsch
GFZ German Research Centre for Geosciences, Potsdam, Germany (lars.lindner@gmail.com)

This article presents a concept for in-depth monitoring of Tsunami Early Warning Systems (TEWS) developed in the projects German Indonesian Tsunami Early Warning System (GITEWS) , Distant Early Warning System (DEWS) and Collaborative, Complex, and Critical Decision-Support in Evolving Crises (TRIDEC). Traditionally such systems are monitored using existing standard software solutions such as Nagios with different levels of customization.

This talk discusses the operational shortcomings of monitoring larger heterogeneous, loosely coupled infrastructures that are not integrated within a service oriented architecture (SOA) framework providing sufficient monitoring utilities. Such a service-level framework could provide the proper means to monitor service interactions. Without such a framework e.g. in case of a loosely coupled component architecture, the means for a proper component-level supervision do not exist. This manifest in creating Nagios-based "service" checks for such an architecture.

Service monitoring using Nagios employs active and passive service checks mostly via ICMP, SSH. The control flow is pulling status from the supervised hosts. When monitoring components, interacting via interfaces not necessarily exposed to a service-level monitoring, pulling the component status becomes inefficient. Nagios administrators bypass this by implementing host-local inspection checks that simulate testable service endpoints to Nagios. We believe that such checks fundamentally fail the service concept of Nagios.

The main problems with this approach are (1) duplicating component code by implementation white box tests based on the internal component control flow or/and data structure and (2) an inevitable divergence of component behaviour and check assumptions over the component development life-cycle. This amounts to a continuously needed administrative action to update the service checks and to inspect whether recent errors reported by the monitoring are real errors or caused by a component deployment.

This paper suggests to move pseudo "service" checks from Nagios to a push based component level monitoring implemented by the monitoring tool SpurTracer. In a push based monitoring components must be migrated to emit notifications to the monitoring server responsible for the correlation of received notifications and thus tracing the control flow to detect errors and timeouts. By correlating notifications of two interacting components the interface between these components is supervised indirectly .

In the projects mentioned sensor data is continuously harvested and fed into the sensor integration platform Tsunami Service Bus (TSB). For this, data is retrieved using a retriever proxy collecting data of a plenty of different sensor stations distributed around the world. Using Nagios it is difficult to monitor if data of a specific sensor has been imported successfully. With Nagios supervision is only possible by implementing a service check keeping track of the sensor data packages and their processing status by inspecting the internal state of the two processing components.

As a proof of concept SpurTracer is integrated for both components, the retriever proxies and the TSB, allowing to monitor the interaction. With interface invocation notifications SpurTracer can automatically detect (1) interface timeouts, (2) component timeouts, and (3) errors reported by the components. By tracking statistical data for the involved hosts, the hosted components, component instances, interfaces, and interface instances SpurTracer allows actively identifying root causes of sudden system failures. Thus SpurTracer assures a component-level operational monitoring additionally to the host and service monitoring provided by Nagios.