Distributed data using the clusterer module

Răzvan Crainea
OpenSIPS Project
razvan@opensips.org

May 10 – 11
Overview

1. Introduction

2. Share data in OpenSIPS 1.*

3. Share data in OpenSIPS 2.2

4. Workshop
Multiple instances?

- Multiple nodes
- Same configuration/behavior
- Different servers
- *Ideally* different geographical locations
Why use multiple instances?

- Increase Capacity
- Balance load
- Failover
- High Availability
all machines can answer to any request, anytime
Active – Active

- all machines can answer to any request, anytime

- All nodes must share all the data!
● Only one machine is active
● A’ is a hot backup
Active – Backup

- A crashes
- A’ becomes active
Active – Backup

- A crashes
- A’ becomes active

- A’ must know all A’s data!
How to share data

- **Centralized**
  - data is stored in a central place
  - single point of failure
  - hard to manage stale data

- **Decentralized**
  - data is stored on each node
  - nodes replicate data
  - each node has its own copy of the data
Share data in OpenSIPS - SQL

- MySQL
  - replication
- Percona
- Galera
- PostgreSQL
Share data in OpenSIPS - NoSQL

- Cassandra
- Couchbase
- Memcached
- MongoDB
- Redis
Share data in OpenSIPS - bin interface

- UDP

- compact binary format

- optimized data
  - send only useful information
  - aggregate information
Share data in OpenSIPS - bin interface

- UDP
- compact binary format
- optimized data
  - send only useful information
  - aggregate information

Downside

UDP does not guarantee the order of the messages, nor that the messages are actually delivered
TCP
- solves the UDP issues

Very Efficient
- built on top of the Transport Interface
- uses OpenSIPS 2.2 TCP stack

Easy to use (programmatically)
Who uses proto_bin?

- ratelimit
  - limits the CPS
Who uses proto_bin?

- ratelimit
  - limits the CPS

- dialog
  - shares dialogs for HA purposes
  - shares dialog profiles
Who uses proto_bin?

- **ratelimit**
  - limits the CPS

- **dialog**
  - shares dialogs for HA purposes
  - shares dialog profiles

- **usrloc**
  - shares user location
Cluster

- Group of nodes/instances with the same function
- They need to communicate between them
- They need to know/learn about each other
Clusterer

- Interface used to group multiple nodes in clusters
- A node is defined by a connector (URL)
  - currently a proto_bin interface
- Groups/clusters can have different purposes
- A node can be part of multiple clusters
Clusterer Management

- Provisioned in database
- Nodes are cached in OpenSIPS’ memory
- Query the clusters status (using MI commands)
- Granular (per node) control
  - Node timeouts, data flow
Clusterer Node

- Communicates with all other nodes
- Has all the information in the cluster
- If a node does not respond for a specific duration – **timeout**
  - data is not yet discarded
- If another timeout occurs – data from that node is discarded
Future Work

- **Ongoing Work**
  - Dynamically learn the network topology
  - Communicate through alternative routes
  - Distribute user location data to different nodes
    - Chord-like algorithm

- **Future Work**
  - Add replication support to other modules
  - Distribute dialogs similar to user location
Take-Away Message

If you want to build an efficient, distributed and highly available platform, all you have to do is OpenSIPS 2.2!

Răzvan Crainea
razvan@opensips.org