Cost-based LCR for OpenSIPS using CGRateS

Dan Christian Bogos
dan.bogos@itsyscom.com

OpenSIPS Summit Amsterdam 2016
Our Background

Located in Bavaria/Germany, over 9 years of experience with architecting server side solutions in VoIP environment

Platform implementations covering both wholesale and retail business categories

Responsibly understanding real-time processing constrains and the seriousness of live system outages

Cost-based LCR for OpenSIPS using CGRateS
OpenSIPS Summit Amsterdam 2016
About CGRateS

Charging/Billing engine
Plug-able into existing billing infrastructure
Accommodate new components into ISP/ITSP network (eg: add new VoIP switch, SMS Service, Data stream)
Non-intrusive into existing setups

Modular architecture
Easy to enhance by rewriting specific components - JSON/HTTP/GOB RPC API

Performance Oriented
Built-in transactional cache system (data ageing, live counters)
Asynchronous processing with micro-threads

Feature-rich
Multi-tenancy, derived charging, account bundles, LCR, CDRStats, rates history, etc
Agile in developing new features

Test driven development
Aprox. 1200 tests as part of the build system

Cost-based LCR for OpenSIPS using CGRateS
OpenSIPS Summit Amsterdam 2016
In a Nutshell, cgrates...

... has had 4,413 commits made by 14 contributors representing 116,074 lines of code

... is mostly written in Go with an average number of source code comments

... has a codebase with a long source history maintained by a large development team with increasing Y-O-Y commits

... took an estimated 30 years of effort (COCOMO model) starting with its first commit in January, 2012 ending with its most recent commit 1 day ago
Fast and ... very fast
<table>
<thead>
<tr>
<th>CGRateS subsystems</th>
<th>Functionality</th>
</tr>
</thead>
</table>
| RATE/ACCOUNT/LCR            | • calculate costs for events  
                              | • maintain accounts           
                              | • compute LCR                
                              | • real-time fraud mitigation  |
| CDR SERVER                  | • centralized CDR server  
                              | • CDR replication             
                              | • forward to CDRStats        |
| SESSION MANAGER             | • maintain/disconnect sessions  
                              | • balance reservation         
                              | • balance refunds             |
| DIAMETER AGENT              | • call control via diameter interface (rfc 4006).                            |
| CDR STATS                   | • compute real-time CDR stats  
                              | • real-time fraud mitigation  |
| ALIASING SERVER             | • alias request/reply information using predefined rules                     |
| USER SERVER                 | • maintain user profiles (LDAP similarity)                                  |
| PUBSUB SERVER               | • expose internal events to subscribed external components                  |
| HISTORY SERVER              | • archive rate changes using GIT in human readable JSON format               |
CGR-RALs (Rating)

Highly configurable rating
Connect fees, rate units, rate increments, rates grouping, a-number rating, various rounding methods, configurable decimals in costs, maximum cost per destination with hit strategy, rating profile scheduling

Multiple TypeOfRecord support

Multiple Category filters for same TOR
(eg: calls, premium_calls, inbound_calls)

Derived Charging
Reseller/distributors chaining or inbound/outbound traffic charging

Cost-based LCR for OpenSIPS using CGRateS
OpenSIPS Summit Amsterdam 2016
CGR-RALs (Accounting)

Prepaid, Postpaid, Pseudo-prepaid controller

Unlimited Balances per Account
Balance selection prioritisation through weights
Unlimited bundle combinations

Shared/Group Balances

Balance lifetime controls
Eg: balance expires or balance is active on specific time intervals

Concurrent sessions per account
Balance reservation in chunks of debit interval
Balance refunds
Debit sleep when needed

Cost-based LCR for OpenSIPS using CGRateS
OpenSIPS Summit Amsterdam 2016
FRAUD MITIGATION

Part of Accounting
Tightly integrated, balance operations cannot avoid it
Minimum & maximum balance monitors
Minimum & maximum counter monitors

Part of CDRStats
Multiple metrics and stat queues thresholds

Scheduler integration
One-time, recurrent triggers

Synchronous & Asynchronous Actions
Realtime CDR Server
Accessible Internal, GOB, JSON, HTTP-JSON, HTTP-REST interfaces

Offline CDR Import (csv, xml, fwv)
Automated via Linux inotify or scheduled
Simultaneous folders monitored with multiple import templates per folder

Zero configuration CDR Sources
FreeSWITCH
Kamailio
OpenSIPS

Cost-based LCR for OpenSIPS using CGRateS
OpenSIPS Summit Amsterdam 2016
Derived Charging support

Real-time CDR replication
Raw or Rated CDRs

CDR Exporter
CSV, Fixed Length Fields, Combined Export templates

Cost-based LCR for OpenSIPS using CGRateS
OpenSIPS Summit Amsterdam 2016
CDR STATS

**Standalone component**
Internally or remotely accessible
Performance oriented

**RawCDR and RatedCDR sources**

**Multiple Stats Queues**
Per server and individually configurable stat queues for same CDR

**Highly configurable Stats Queues**
QueueLength, TimeWindow, Metrics
CDR Field Filters

**Individually configured ActionTriggers**
One-time, recurrent triggers
Synchronous & Asynchronous Actions executed
Part of the Fraud Detection mechanism

**Cost-based LCR for OpenSIPS using CGRateS**
OpenSIPS Summit Amsterdam 2016
CGR-RALs (LCR)

Core component logic
Internally or remotely accessible through APIer or RATER components
Non-intrusive, injects supplier information into Telecom Switch

Tightly coupled with ACCOUNTING subsystem
Provides LCR over bundles

Integrates traffic patterns
Computes LCR for specific call duration

Advanced profile selection mechanism
Filter on Direction, Tenant, Category, Account, Subject/CLI prefix, Destination
Weight based prioritization
Activation time

Extended functionality through multiple strategies
Flexible strategy parameters

Cost-based LCR for OpenSIPS using CGRateS
OpenSIPS Summit Amsterdam 2016
**static**
Classic way of LCR, suppliers ordered based on configured rule parameters

```
*out,cgrates.org,call,1001,*any,DST_1002,lcr_profile1,*static,suppl2;suppl1,2014-01-14T00:00:00Z,10
```

LCR Strategies (1)
**lowest_cost**
Use supplier with least cost

```
*out,cgrates.org,call,*any,*any,*any,lcr_profile1,*lowest_cost,,2014-01-14T00:00:00Z,10```

```
root@CgrDev1:~/cgrates# cgr-console 'lcr Account="1005" Destination="1002"'
{
  "DestinationId": "DST_1002",
  "RPCategory": "lcr_profile2",
  "Strategy": "*lowest_cost",
  "Suppliers": [
    {
      "Supplier": "suppl3",
      "Cost": 0.01,
      "QOS": null
    },
    {
      "Supplier": "suppl1",
      "Cost": 0.6,
      "QOS": null
    },
    {
      "Supplier": "suppl2",
      "Cost": 1.2,
      "QOS": null
    }
  ]
}
```

**LCR Strategies (2)**
*highest_cost*
Use supplier with highest cost

```
"*out,cgrates.org,call,1002,*any,DST_1002,lcr_profile1,*highest_cost,,2014-01-14T00:00:00Z,10"
```

```json
root@CgrDev1:~/cgrates# cgr-console 'lcr Account="1002" Destination="1002"'
{
    "DestinationId": "DST_1002",
    "RPCategory": "lcr_profile1",
    "Strategy": "*highest_cost",
    "Suppliers": [
    {
        "Supplier": "suppl1",
        "Cost": 1.2,
        "QOS": null
    },
    {
        "Supplier": "suppl2",
        "Cost": 0.6,
        "QOS": null
    }
    ]
}
```

LCR Strategies (3)
**qos_threshold**
Supplier with lowest cost, matching QoS thresholds min/max for ASR, ACD, TCD, ACC, TCC
"*out,cgrates.org,call,1003,*any,DST_1002,lcr_profile1,*qos_threshold,20;;2
m;;;;;;,2014-01-14T00:00:00Z,10"

```
root@CgrDev1:~/cgrates/general_tests# cgr-console 'lcr Account="1003" Destination="1002"'
{
    "DestinationId": "DST_1002",
    "RPCategory": "lcr_profile1",
    "Strategy": "*qos_threshold",
    "Suppliers": [{
        "Name": "CGRATES", "Engine": "active",
        "GetMetrics": (StatsQueueId: "CDRST_1003"), {statMetrics}: err.Error()
    }, {"Supplier": "suppl1", err.Error()}]
}
```

**LCR Strategies (4)**
*qos
Supplier with best quality, independent on cost
"*out,cgrates.org,call,1002,*any,*any,lcr_profile1,*qos,,2014-01-14T00:00:00Z,10"

LCR Strategies (5)
**load_distribution**

Load based results resulted, configurable supplier ratios, individually or server defaults

```
"out":cgrates.org,"call":1004,"any":DST_1002,"lcr_profile1","load_distribution","supplier1:5;supplier2:3;","default":1,2014-01-14T00:00:00Z,10"
```

**LCR Strategies (6)**
Multiple integration mechanisms
Based on traffic profile
Shared data through pseudovariables

REST_CLIENT for call authorization, LCR
HTTP-JSÖN RPC Request/Answer

EVI ACC_ACCOUNTING
*prepaid, *pseudoprepaid, *postpaid, *rated

EVI E_ACC_CDR
*pseudoprepaid, *postpaid, *rated
MI_DATAGRAM
Ability to tear-down calls in real-time via dlg_end_dlg

CDR.csv processing (db_flatfile)
*pseudoprepaid, *postpaid, *rated
caching module for CDR split over multiple files

Integration tutorial available
Questions?

Website
http://www.cgrates.org

Documentation
http://cgrates.readthedocs.org

Code + issues tracker
https://github.com/cgrates/cgrates

Support
Google group: CGRateS
IRC Freenode: #cgrates

Cost-based LCR for OpenSIPS using CGRateS
OpenSIPS Summit Amsterdam 2016