Slider Makes Running Applications on YARN a Breeze

Ted Yu
Agenda

Introduction to Slider

Authoring Application Packages

Ambari View for Slider

Docker based app packaging
Fun operational problems

Placement: where to run?
Installation
Configuration & Binding
Client configuration
Lifecycle
Failure handing and recovery
Logging
Upgrading
Metrics & Monitoring

Start/Stop
Reconfigure
Scale up/down
Rolling-restart
Decommission/Recommission
Apache Slider
Deploying and managing applications on Apache Hadoop YARN

http://slider.incubator.apache.org/
YARN schedules work

- Servers run YARN Node Managers (NM)
- NM's heartbeat to Resource Manager (RM)
- RM schedules work over cluster
- RM allocates containers to apps
- NMs start containers
- NMs report container health
Client creates Slider App Master

- Client
- YARN Node Manager
  - Slider Application Master
  - HDFS
- YARN Resource Manager “The RM”
  - HDFS
Slider App Master runs the application
An application consists of

“Application Package”

JSON configuration files: YARN resources + config

Data persisted in HDFS

Placement history in HDFS
resources.json

{
  "schema": "http://example.org/specification/v2.0.0",
  "global": {
    "yarn.memory": "512"
  },
  "components": {
    "HBASE_MASTER": {
      "yarn.role.priority": "1",
      "yarn.component.instances": "1",
      "yarn.vcores": "1"
    },
    "HBASE_REGIONSERVER": {
      "yarn.role.priority": "2",
      "yarn.component.instances": "1"
    }
  }
}
configure: app_config.json

{

"application.def": "/slider/hbase_v096.zip",

"site.global.app_log_dir": "${AGENT_LOG_ROOT}/app/log",
"site.global.app_pid_dir": "${AGENT_WORK_ROOT}/app/run",

"site.global.hbase_master_heapsize": "1024m",
"site.global.ganglia_server_host": "${NN_HOST}",
"site.global.ganglia_server_port": "8667",
"site.global.ganglia_server_id": "Application1",

"site.hbase-site.hbase.tmp.dir": "${AGENT_WORK_ROOT}/work/app/tmp",
"site.hbase-site.hbase.master.info.port": "${HBASE_MASTER.ALLOCATED_PORT}"
"site.hbase-site.hbase.regionserver.port": "0",
"site.hbase-site.hbase.zookeeper.quorum": "${ZK_HOST}"

"site.core-site.fs.defaultFS": "${NN_URI}"

}
create, start, stop, destroy

$ slider create hbase1 --resources resources.json --template config.json

$ slider list

$ slider status hbase1

$ slider stop hbase1

$ slider start hbase1

$ slider destroy hbase1
Dynamic application resize

slider flex hbase1
  --component HBASE_REGIONSERVER 2
Similar to any YARN application
1. CLI starts an instance of the AM
2. AM requests containers
3. Containers activate with an Agent
4. Agent gets application definition
5. Agent registers with AM
6. AM issues commands
7. Agent reports back, status, configuration, etc.
8. AM publishes endpoints, configurations
Slider AppMaster/Agent/Client

• AppMaster
  - Common YARN interactions
  - Common *-client interactions
  - Publishing needs

• Agent
  - Configure and start
  - Re-configure and restart
  - Heartbeats & failure detection
  - Port allocations and publishing
  - Custom commands if any (e.g. graceful-stop)

• Client
  - App life cycle commands (flex, status, …)
  - Package installation
## Slider App Master

**Application: hbase1**

- **Status:** all containers allocated
- **Create time:** 28 Feb 2015 02:35:56 GMT
- **Running since:** 28 Feb 2015 02:35:56 GMT
- **Time last flexed:** N/A

Application storage path:
- hdfs://c6403.ambari.apache.org:8020/user/yarn/slide/cluster/hbase1/database

Application configuration path:
- hdfs://c6403.ambari.apache.org:8020/user/yarn/slide/cluster/hbase1/snapshots

### Component Instances

<table>
<thead>
<tr>
<th>Component</th>
<th>Desired</th>
<th>Actual</th>
<th>Outstanding Requests</th>
<th>Failed</th>
<th>Failed to start</th>
</tr>
</thead>
<tbody>
<tr>
<td>HBASE_MASTER</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HBASE_REGIONSERVER</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HBASE_REST</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HBASE_THRIFT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HBASE_THRIFT2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>slider-apmmaster</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Slider Agent Information

- http://c6403.ambari.apache.org:1025
- HBASE_REGIONSERVER Host(s)/Container(s): [c6403.ambari.apache.org/container_1424910595794_0005_01_000003]
- HBASE_MASTER Host(s)/Container(s): [c6403.ambari.apache.org/container_1424910595794_0006_01_000002]
- slider-apmmaster Host(s)/Container(s): [c6403.ambari.apache.org/container_1424910595794_0006_01_000001]
- HBASE_REST Host(s)/Container(s): [c6403.ambari.apache.org/container_1424910595794_0006_01_000004]
YARN notifies AM
AM requests replacement
Registration and Discovery

• Application must declare itself
  – URLs
  – Host/port
  – Config (client config)

• Application must be discoverable
  – Registry
  – Name-based lookups
  – Regularly updated

• Client support
  – Callback if “data” changes; thick clients
  – Configurable gateway; thin clients
Application Registry

• A common problem (not specific to Slider)
  - [https://issues.apache.org/jira/browse/YARN-913](https://issues.apache.org/jira/browse/YARN-913)

• Currently,
  - Apache Curator based
  - Register URLs pointing to actual data
  - AM doubles up as a webserver for published data

• Plan
  - Registry should be stand-alone
  - Slider is a consumer as well as publisher
  - Slider focuses on declarative solution for Applications to publish data
  - Allows integration of Applications independent of how they are hosted
App Packaging Capabilities

• Dynamic port allocation and sharing

• Inter-component dependency
  – Specify the start order of components

• Exports
  – Construct arbitrary name value pairs
  – E.g. URLs (org.apache.slider.monitor: http://${HBASE_MASTER_HOST}:${site.hbase-site.hbase.master.info.port}/master-status)

• Default HDFS and ZK isolation
In addition to …

• Security
  - Configured for security
  - Token renewal and/or keytabs

• High Availability
  - On a highly available cluster (NN, RM HA)
  - Itself highly available (multi-master)

• Packaging
• Configurability
• …
Slider View for Apache Ambari

http://ambari.apache.org/
### Summary

<table>
<thead>
<tr>
<th>Status</th>
<th>RUNNING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>HBASE</td>
</tr>
<tr>
<td>YARN Application ID</td>
<td>application_1424910595794_0004</td>
</tr>
<tr>
<td>Started</td>
<td>Wed, 25 Feb 2015, 17:43:07 -08:00 GMT</td>
</tr>
<tr>
<td>Finished</td>
<td>-</td>
</tr>
<tr>
<td>Diagnostics</td>
<td>-</td>
</tr>
<tr>
<td>Cluster Id</td>
<td>b51cd7ae-4a41-419b-9bc7-ac732c0dfcb7</td>
</tr>
<tr>
<td>Cluster Requests</td>
<td>33</td>
</tr>
<tr>
<td>Dead Region</td>
<td>0</td>
</tr>
<tr>
<td>Servers</td>
<td>0</td>
</tr>
<tr>
<td>Is Active Master</td>
<td>true</td>
</tr>
<tr>
<td>Master Active Time</td>
<td>16h:4m</td>
</tr>
<tr>
<td>Master Start Time</td>
<td>Thu, 26 Feb 2015 01:45:33 GMT</td>
</tr>
<tr>
<td>Metric Average</td>
<td>2.0</td>
</tr>
<tr>
<td>Load</td>
<td>1</td>
</tr>
<tr>
<td>Region Servers</td>
<td>1</td>
</tr>
<tr>
<td>Server Name</td>
<td>c6403.ambari.apache.org,52801,1424915129707</td>
</tr>
<tr>
<td>Zookeeper Quorum</td>
<td>c6403.ambari.apache.org:2181</td>
</tr>
</tbody>
</table>

### Status

- **HBASE_MASTER** component: 1 out of 1 active
- **HBASE_REGIONSERVER** component: 1 out of 1 active
- **HBASE_REST** component: 1 out of 1 active
- **slider-appmaster** component: 1 out of 1 active

### Components

- **HBASE_MASTER**: c6403.ambari.apache.org
- **HBASE_REGIONSERVER**: c6403.ambari.apache.org
- **HBASE_REST**: c6403.ambari.apache.org
- **slider-appmaster**: c6403.ambari.apache.org
Project Status

• In ASF incubation
• 0.70-incubating release - March
• Growing set of application packages
• Working in simplifying packaging
• HDP 2.2 added long-lived service support for Slider and other services
HDP2.2 features for Slider

- Labelled nodes & queues
- Log aggregation
- Container retention over Application Master restart
- Service registration & discovery
- Kerberos token renewal
- Windowed Application failure tracking

—essential for all long-lived YARN services
On-boarding Dockerized Application on YARN via Slider

- Application definition – metainfo.json (contains docker image name)
- Application instance configuration (if any) – app_config.json
- Application resource definition – resources.json
- Dockerized node.js app using dockerized redis for persistence

— No Application bundle and lifecycle scripts
metainfo.json – structure of the application

```json
{
    "schemaVersion": "2.1",
    "application": {
        "name": "NODEJS-REDIS",
        "components": [
            {
                "name": "NODEJS",
                "type": "docker",
                "dockerContainers": [
                    {
                        "name": "nodejs",
                        "image": "rsahahw/centos-node-redis",
                        "ports": [{"containerPort": "8000"}]
                    }
                ],
                "ports": [{"containerPort": "8000"}]
            },
            {
                "name": "REDIS",
                "type": "docker",
                "dockerContainers": [
                    {
                        "name": "redis",
                        "image": "tutum/redis",
                        "ports": [{"containerPort": "6379",
                                    "hostPort": "6379"}]
                    }
                ],
                "ports": [{"containerPort": "6379",
                                    "hostPort": "6379"}]
            }
        ]
    }
}
```
resources.json – resource requirement of the application

```json
{
  "schema": "http://example.org/specification/v2.0.0",
  "metadata": {},
  "global": {},
  "components": {
    "NODEJS": {
      "yarn.role.priority": "1",
      "yarn.component.instances": "1",
      "yarn.memory": "512"
    },
    "REDIS": {
      "yarn.role.priority": "2",
      "yarn.component.instances": "1",
      "yarn.memory": "512"
    }
  }
}
```
app_config.json – instance specific configuration of the application

{
    "schema": "http://example.org/specification/v2.0.0",
    "metadata": {},
    "global": {},
    "components": {
        "NODEJS": {
            "nodejs.options": "-d -e REDIS_HOST=${REDIS_HOST}",
            "nodejs.statusCommand": "/usr/bin/docker ps"
        },
        "REDIS": {
            "redis.options": "-d -e REDIS_PASS=**None**",
            "redis.statusCommand": "/usr/bin/docker ps"
        }
    }
}
Coming up

- SLIDER-799: AM-managed placement escalation
  1. Redundant container request from slider may cause high load on busy cluster

- SLIDER-787 App Upgrade/Reconfig support in Slider
  1. Allows rolling upgrade / downgrade
  2. app packages are versioned