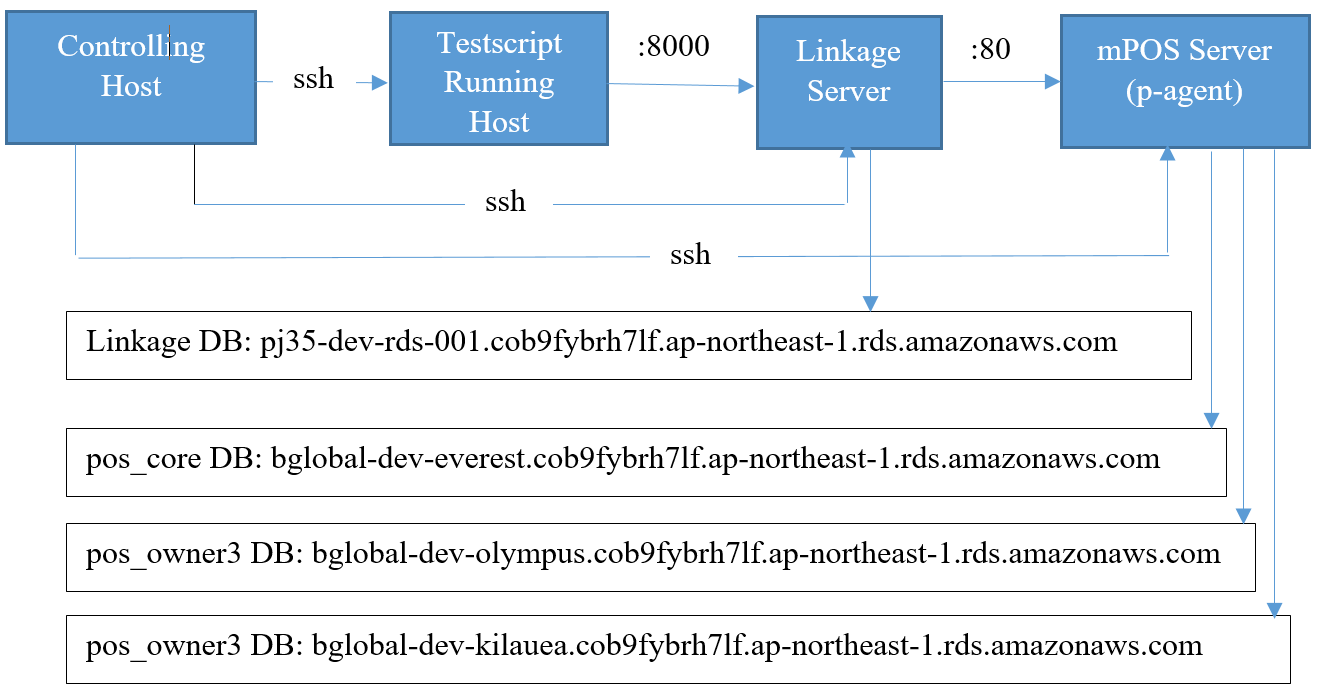
STRESS TEST DESCRIPTION

1. Measured factors

* Total orders: Number of orders Linkage received in test time
* Total successful order: Number of orders successfully excecuted
* Total unprocessed order: Number of remaining unprocessed orders in JDBC
* Average received request time of Linkage (API call): Average time linkage received request, create job and response to external service the order code.
* Average order time: Average time linkage received request until order is created. (This will include the time order stayed in JDBC)
* Max order time: Max time linkage received request until order is created to mPOS system.
* Min order time: Min time linkage received request until order is created to mPOS system.
* Total <1s: Number of orders created under 1 second.
* Total <3s: Number of orders created under 3 second.
* Ratio of orders <1s, <3s
* Average job execution time: Average time job being executed
* Average refine order p-agent: Average time Linkage send refine api to p-agent until received response from p-agent.
* Max refine order time p-agent
* Average create order p-agent: Average time Linkage send create api to p-agent until received response from p-agent.
* Max create order time p-agent
* Average job delete time: Average delete job time after completed.

1. Test instruction
2. **Setup for stress test**



1. **Linkage Server:**

* Target undergoing stress test
* Public static IP address (such as: **13.231.186.22**) or Public domain name (such as: ec2-13-231-186-22.ap-northeast-1.compute.amazonaws.com)
* Running service that is listening on port **8000** (access can be restricted to only one or some pre-defined hosts)
* Must have access to Linkage Service DB, such as pj35-dev-rds-001.cob9fybrh7lf.ap-northeast-1.rds.amazonaws.com, the database **linkage** resides here.

1. **mPOS Server:**

* Running new modified p-agent, that is handling requests coming from Linkage Server
* Public static IP address (such as: **54.238.253.71**) or Public domain name (such as: ec2-54-238-253-71.ap-northeast-1.compute.amazonaws.com)
* Must have access to mPOS DB system, pos\_core3, pos\_owner3… such as:
  + bglobal-dev-everest.cob9fybrh7lf.ap-northeast-1.rds.amazonaws.com: **POS\_CORE\_MASTER\_HOST**, **POS\_CORE\_SLAVE\_HOST** in /home/blaynregi/BLAYNREGI/webdata/conf/database\_config.inc and **url** in /home/blaynregi/java/v0.4.x/agent/resources/jdbc.properties and **sql\_hostname** in expshopitem.py (see below)
  + bglobal-dev-olympus.cob9fybrh7lf.ap-northeast-1.rds.amazonaws.com: **POS\_STORE\_HOST** in /home/blaynregi/BLAYNREGI/webdata/conf/database\_config.inc and **sql\_hostname\_s1** in expshopitem.py (see below)
  + bglobal-dev-kilauea.cob9fybrh7lf.ap-northeast-1.rds.amazonaws.com: **sql\_hostname\_s2** in expshopitem.py (see below)

1. **Testscript Running Host:**

* Any host that can run python3 and also act as ssh server
* A Windows host can be used but a Linux is a better choice
* EC2 instance is highly recommended for minimizing network latencies (here: **13.112.109.57** or ec2-13-112-109-57.ap-northeast-1.compute.amazonaws.com)
* Must have access to Linkage Server via port 8000 as descibed above

1. **Controlling Host:**

* Used for synchronizing tasks between them during stress test
* Running Linux with python3 support
* Can connect to three main components above (Linkage Server, mPOS Server and Testscript Running Host) via ssh with unattended authentication by public key
* Current working directory contains the following files:
  + bco.py: python script, that is to be copied to **Testscript Running Host** via scp (target diretory: ~/pj35)
  + expshopitem.py: to extract shop login credentials and items data for bco.py to run. Also the generated files (shops.tsv and items.tsv) have to be copied to **Testscript Running Host** via scp (target diretory: ~/pj35 same as bco.py)
  + genscript.sh: to generate a shell script for a stress test
  + go.sh: the wrapper script that runs all other scripts in the right sequence

1. **Run stress test**
   * Prepare the empty database for LS and export to linkage.sql. It will be used to restore for fresh DB each test.
   * Run go.sh to execute script of stress test
2. **Analysis log data and report**
   * Download all log from LS
   * Execute runallawk.sh to analysis log
   * Get result in results folder and copy data to temple file: Stress test report.xlsx
3. Scenario
4. **Scenario 1: Linkage use quartz**

* **Flow**: Script test send request create order to Linkage with speed X orders/s -> Linkage create job and save to Quartz JDBC -> Quartz with Y threads get jobs and from JDBC and execute -> A job excecution will send 2 request to p-agent to create order -> Quartz delete job in JDBC after completed.
* **Test case**: Run script for 10 mins with 1 pair of params X/Y from { 5/10, 10/10, 25/10, 50/10, 100/10, 500/10, 5/20, 10/20, 25/20, 50/20, 100/20, 500/20, 5/50, 10/50, 25/50, 50/50, 100/50, 500/50, 5/100, 10/100, 25/100, 50/100, 100/100/ 500/100 } for each test case.
* **Test process**: Run 24 testcases corresponding with 24 pairs of params. Repeat 5 times to get average result.

1. **Scenario 2: Linkage without quartz**

* **Flow**: Script test send request create order to Linkage with speed X orders/s -> For each order, Linkage will send 2 requests to p-agent.
* **Test case:** Run script for 10 mins with param X from {5, 10, 25, 100, 500} each test case.