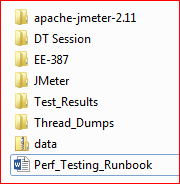
Setup details: <https://rodanandfields.atlassian.net/wiki/display/ET/Performance+Results>

Results: <https://rodanandfields.atlassian.net/wiki/display/ET/POC+Test+Results>

Box Results: <https://rodanandfields.app.box.com/files/0/f/6467044117/Perf>

Test plan: <https://rodanandfields.atlassian.net/wiki/display/ET/Performance+Testing+and+Improvement>

All scripts/test results/test data are kept on S drive under S:\Website Development\Performance



* Apache-jmeter-2.11 has all the required libraries/plugins for jdbc etc.
* DT Sessions has all the previous tests saved dynatrace sessions
* JMeter has all the jmx scripts for DB/EIS/Hybris/Solr. There are 2 versions maintained one for CLI mode and one for UI.
* Test\_Results has the jtl files for all the tests run so far on perf and poc environment.

Store Front

* All UI and CLI scripts are under S:\Website Development\Performance\JMeter\Hybris
* All test results are under S:\Website Development\Performance\Test\_Results\Hybris
* Staging envt:
* The server(LG) that needs to be used for testing on staging envt. is: 10.223.176.143

Execute script: runall\_CA\_BM.sh 100 500 1800 (Stdout is written to nohup.out) for (100 is 1/5th of US browse users. For Trans script, 20 users are hard coded in the script)

* 4 jmeter nodes are used :

|  |
| --- |
| 162.209.126.13 |
| 162.209.105.228 |
| 162.209.110.109 |
| 104.130.17.60 |

* User on all 4 nodes – ptuser/Rf!p5user
* Script location on all 4 nodes
  + Browse:
    - /home/ptuser/JMeter1
  + Transactional:
    - /home/ptuser/JMeter2
* Wrapper script for all nodes is under /home/ptuser/Run
* Commands to run script
  + Browse and transactional:
    - ./runall.sh 143 500 1800 (Stdout is written to nohup.out) for 25% traffic
      * 143 – users
      * 500 – rampup time
      * 1800 - duration
    - ./runall\_poc2.sh 286 500 1800 for 50% traffic

Simillarly for running script on prf environment can use the below command.

* + - ./runall\_prf2.sh 286 500 1800 for 50% traffic
* new Shell for full controll-

runall\_poc\_fullcontroll.sh browseTh browseRup TestDuration adhocTh stdTh exprTh pcTh rcTh txRup

* These scripts create jtl files. These files need to be scp’d to windows machine and loaded into jmeter. Save the csv output.
* Solr scripts are kept under /home/ptuser/JMeter1

Web Services

* The Boomi scripts is kept under S:\Website Development\Performance\JMeter\Hybris\UI\_Scripts\ BoomiAPI\_seperateTG
* The script is run directly from the windows machine.
* Command to create test data for the POST web service calls:
  + reassignsponsor:

Flex query to get test data:

select top 1000 p\_sponsorid,p\_rfaccountid from [dbo].[users] order by createdTS desc

Copy the data from the above command into a file boomi\_orders on linux. Run the below command on linux:

while read a b; do echo "<sponsorInfo>"; echo "<accountId>$a</accountId>"; echo "<currentSponsorId>$b</currentSponsorId>"; echo "<newSponsorId>521877</newSponsorId>" ; echo "</sponsorInfo>"; done < boomi\_orders > test.txt

* + commisionsinfo:

while read a b; do echo "<orderCommissionUpdate>"; echo "<startPeriod>2015-09-09T12:31:25.41</startPeriod>"; echo "<endPeriod>2015-09-14T12:19:55.192</endPeriod>"; echo "<rfAccountId>$a</rfAccountId>" ; echo "<distributorId>$b</distributorId>"; echo "</orderCommissionUpdate>" ; done < boomi\_orders > test1.txt

* + terminate:

while read a b; do echo "<terminatedAccount>"; echo "<accountId>$a</accountId>"; echo "<reasonId>7</reasonId>"; echo "<reasonTitle>Too expensive</reasonTitle>" ; echo "<terminationComments>This is good but too expensive</terminationComments>"; echo "<volunteerTermination>true</volunteerTermination>" ; echo "<hardTerminate>true</hardTerminate>"; echo " <hardTerminateDate>2015-08-22</hardTerminateDate>" ; echo "</terminatedAccount>" ; done < boomi\_orders > test2.txt

* + updateorders:

while read a b; do echo "<orderUpdate>"; echo "<orderNumber>$a</orderNumber>"; echo "<distributorId>521877</distributorId>"; echo "<cv>125.05</cv>" ; echo "<qv>105.15</qv>"; echo "<commissionDate>2014-05-14</commissionDate>" ; echo "</orderUpdate>"; done < orders4 > test4.txt

* + consignment:

Flex query to get test data:

select {O:code}, {P:catalogNumber} from {Order as O} (NOLOCK), {Product AS P} (NOLOCK), {ORDERENTRY AS OE} (NOLOCK) where {O:pk}={OE:ORDER} AND {OE:PRODUCT}={P:PK}

Copy this output to a file orders4 on linux. Run the below linux command :

while read a b; do echo "<shipmentUpdate>"; echo "<orderNumber>$b</orderNumber>"; echo "<shipNumber>22529</shipNumber>"; echo "<sku>$a</sku>" ; echo "<quantity>1</quantity>"; echo "<status>SHIPPED</status>"; echo "<shipMethod>UPS Ground (HD)</shipMethod>"; echo "<shippingAddress>1055 WaterMark</shippingAddress>"; echo "<warehouse>belmont</warehouse>"; echo "<trackNumber>1Z91RV682111112111</trackNumber>"; echo "<statusDate>2015-10-16T00:00:00.000</statusDate>"; echo "<orderItemNumber>1</orderItemNumber>"; echo "<carrier>UPS</carrier>"; echo "<lotNumber>1</lotNumber>"; echo "</shipmentUpdate>" ; done < orders4 > test5.txt

Autoships

* Enabling faking by setting the below keys in local.properties:

fake.autoship.paymentandtaxes=true

fake.litle=true

fake.ava=true

* Below are the queries to create test data:
  + CRP:

--CRP:

DECLARE @Futuredate DATE = '2016-12-20';--

DECLARE @lastpDate DATE= '2015-12-10';--10 days before today.

DECLARE @Rundatedate DATE = '2015-12-21';--within 15 days before today.

/\* forward all scheduling date to future \*/

UPDATE ho

SET ho.p\_schedulingdate = @Futuredate ,

ho.p\_lastprocessingdate = @lastpDate

FROM Hybris..orders ho

JOIN Hybris..users u ON u.PK = ho.userpk

JOIN Hybris..vEnumerationValues v ON v.PK = u.p\_accountstatus

AND v.Value = 'Active'

WHERE ho.p\_template = 1

AND ho.p\_active = 1

AND ho.p\_ccfailurecount < 3 --Pcperks only changed to 4

AND ho.TypePkString = 8796124676178;--Crp templates

--AND ho.TypePkString=8796124741714-- Pulse Templates

---AND ho.TypePkString=8796124708946 --pcperks

IF OBJECT\_ID('tempdb..#t') IS NOT NULL

DROP TABLE #t;

SELECT TOP 27000 --Counts to get updates

ho.PK

INTO #t

FROM Hybris..orders ho

JOIN Hybris..users u ON u.PK = ho.userpk

JOIN Hybris..vEnumerationValues v ON v.PK = u.p\_accountstatus

AND v.Value = 'Active'

WHERE ho.p\_template = 1

AND ho.p\_active = 1

AND ho.p\_ccfailurecount < 3 --Pcperks only changed to 4

AND ho.TypePkString = 8796124676178;--Crp templates

--AND ho.TypePkString=8796124741714-- Pulse Templates

---AND ho.TypePkString=8796124708946 --pcperks

UPDATE ho

SET ho.p\_schedulingdate = @Rundatedate

FROM Hybris..orders ho

JOIN #t t ON t.PK = ho.PK

JOIN Hybris..users u ON u.PK = ho.userpk

JOIN Hybris..vEnumerationValues v ON v.PK = u.p\_accountstatus

AND v.Value = 'Active'

WHERE ho.p\_template = 1

AND ho.p\_active = 1

AND ho.p\_ccfailurecount < 3 --Pcperks only changed to 4

AND ho.TypePkString = 8796124676178;--Crp templates

--AND ho.TypePkString=8796124741714-- Pulse Templates

---AND ho.TypePkString=8796124708946 --pcperks

* + PCPerks --PCPerks:

DECLARE @Futuredate DATE = '2016-12-20';--

DECLARE @lastpDate DATE= '2015-12-10';--10 days before today.

DECLARE @Rundatedate DATE = '2015-12-17';--within 15 days before oftoday.

/\* forward all scheduling date to future \*/

UPDATE ho

SET ho.p\_schedulingdate = @Futuredate ,

ho.p\_lastprocessingdate = @lastpDate

FROM Hybris..orders ho

JOIN Hybris..users u ON u.PK = ho.userpk

JOIN Hybris..vEnumerationValues v ON v.PK = u.p\_accountstatus

AND v.Value = 'Active'

WHERE ho.p\_template = 1

AND ho.p\_active = 1

AND ho.p\_ccfailurecount < 4 --Pcperks only changed to 4

--AND ho.TypePkString = 8796124676178;--Crp templates

--AND ho.TypePkString=8796124741714-- Pulse Templates

AND ho.TypePkString=8796124708946 --pcperks

IF OBJECT\_ID('tempdb..#t') IS NOT NULL

DROP TABLE #t;

SELECT TOP 53000 --Counts to get updates

ho.PK

INTO #t

FROM Hybris..orders ho

JOIN Hybris..users u ON u.PK = ho.userpk

JOIN Hybris..vEnumerationValues v ON v.PK = u.p\_accountstatus

AND v.Value = 'Active'

WHERE ho.p\_template = 1

AND ho.p\_active = 1

AND ho.p\_ccfailurecount < 4 --Pcperks only changed to 4

-- AND ho.TypePkString = 8796124676178;--Crp templates

--AND ho.TypePkString=8796124741714-- Pulse Templates

AND ho.TypePkString=8796124708946 --pcperks

UPDATE ho

SET ho.p\_schedulingdate = @Rundatedate

FROM Hybris..orders ho

JOIN #t t ON t.PK = ho.PK

JOIN Hybris..users u ON u.PK = ho.userpk

JOIN Hybris..vEnumerationValues v ON v.PK = u.p\_accountstatus

AND v.Value = 'Active'

WHERE ho.p\_template = 1

AND ho.p\_active = 1

AND ho.p\_ccfailurecount < 4 --Pcperks only changed to 4

--AND ho.TypePkString = 8796124676178;--Crp templates

--AND ho.TypePkString=8796124741714-- Pulse Templates

AND ho.TypePkString=8796124708946 --pcperks

* + PulseRenew --Pulse:

DECLARE @Futuredate DATE = '2016-12-20';--

DECLARE @lastpDate DATE= '2015-12-10';--10 days before today.

DECLARE @Rundatedate DATE = '2015-12-17';--within 15 days before oftoday.

/\* forward all scheduling date to future \*/

UPDATE ho

SET ho.p\_schedulingdate = @Futuredate ,

ho.p\_lastprocessingdate = @lastpDate

FROM Hybris..orders ho

JOIN Hybris..users u ON u.PK = ho.userpk

JOIN Hybris..vEnumerationValues v ON v.PK = u.p\_accountstatus

AND v.Value = 'Active'

WHERE ho.p\_template = 1

AND ho.p\_active = 1

AND ho.p\_ccfailurecount < 3 --Pcperks only changed to 4

--AND ho.TypePkString = 8796124676178;--Crp templates

AND ho.TypePkString=8796124741714-- Pulse Templates

--AND ho.TypePkString=8796124708946 --pcperks

IF OBJECT\_ID('tempdb..#t') IS NOT NULL

DROP TABLE #t;

SELECT TOP 32000 --Counts to get updates

ho.PK

INTO #t

FROM Hybris..orders ho

JOIN Hybris..users u ON u.PK = ho.userpk

JOIN Hybris..vEnumerationValues v ON v.PK = u.p\_accountstatus

AND v.Value = 'Active'

WHERE ho.p\_template = 1

AND ho.p\_active = 1

AND ho.p\_ccfailurecount < 3 --Pcperks only changed to 4

-- AND ho.TypePkString = 8796124676178;--Crp templates

AND ho.TypePkString=8796124741714-- Pulse Templates

--AND ho.TypePkString=8796124708946 --pcperks

UPDATE ho

SET ho.p\_schedulingdate = @Rundatedate

FROM Hybris..orders ho

JOIN #t t ON t.PK = ho.PK

JOIN Hybris..users u ON u.PK = ho.userpk

JOIN Hybris..vEnumerationValues v ON v.PK = u.p\_accountstatus

AND v.Value = 'Active'

WHERE ho.p\_template = 1

AND ho.p\_active = 1

AND ho.p\_ccfailurecount < 3 --Pcperks only changed to 4

--AND ho.TypePkString = 8796124676178;--Crp templates

AND ho.TypePkString=8796124741714-- Pulse Templates

-- AND ho.TypePkString=8796124708946 --pcperks

* Command to confirm the correct number of autoships are scheduled for today:

SELECT COUNT(\*) counts

FROM Hybris..orders ho

JOIN Hybris..users u ON u.PK = ho.userpk

JOIN Hybris..vEnumerationValues v ON v.PK = u.p\_accountstatus

AND v.Value = 'Active'

WHERE ho.p\_template = 1

AND ho.p\_active = 1

AND ho.p\_ccfailurecount < 3 --Pcperks only changed to 4

AND ho.TypePkString = 8796124676178 --Crp templates

--AND ho.TypePkString=8796124741714-- Pulse Templates

---AND ho.TypePkString=8796124708946 --pcperks

AND CAST(ho.p\_schedulingdate AS DATE) BETWEENCAST(DATEADD(DAY,

-30, GETDATE()) AS DATE)

AND CAST(GETDATE() AS DATE)

* Once the test data is created start autoship from HMC:
  + Hmc -> System -> cronjobs ->
  + Select Job -> autoshipCRPProcessesJob/autoshipPCPerksProcessesJob/autoshipPulseRenewalProcessesJob
* Commands to check how BP have been spawned: (Change the date)

select count (\*) FROM rfautoshipprocesslog WHERE p\_cronjobcode like 'autoshipCRPProcessesJob12232015%'

select count (\*) FROM rfautoshipprocesslog WHERE p\_cronjobcode like 'autoshipPulseRenewalProcessesJob12232015%'

select count (\*) FROM rfautoshipprocesslog WHERE p\_cronjobcode like 'autoshipPCPerksProcessesJob12232015%'

* Carry out db indexing/stats

To confirm all BP’s have ended check for the status ENDED.

* To abort autoship, stop hybris and flush the tasks, processes, processParameters, TaskLogs and TaskConditions table in hybris db.
* Carry out db indexing/stats
* Hybris db credentials
* db.username=HybrisPRDPRF^M
* db.password=yFew69Fp7^M
* Command to check the time taken for autoship to complete: (One for each type)

<http://www.corprfo.poc.rodanandfields.com/hac/console/flexsearch/>

SELECT DATEDIFF(mi, (SELECT TOP 1 logstarttime.p\_businessprocessstarttime FROM rfautoshipprocesslog logstarttime,composedtypes itemtype WHERE ( logstarttime.p\_cronjobcode =**'autoshipPCPerksProcessesJob10032015165354'**) AND (logstarttime.TypePkString=itemtype.pk AND itemtype.internalcode='RFAutoshipProcessLog') order by logstarttime.p\_businessprocessstarttime ASC), (SELECT TOP 1 logendtime.p\_businessprocessendtime FROM rfautoshipprocesslog logendtime,composedtypes itemtype WHERE ( logendtime.p\_cronjobcode =**'autoshipPCPerksProcessesJob10032015165354'**) AND (logendtime.TypePkString=itemtype.pk AND itemtype.internalcode='RFAutoshipProcessLog') order by logendtime.p\_businessprocessendtime DESC) ) AS TotalExecutionTimeInMinutes

* Command to check how many BP’s were spawned by the threads set by task.worker.max

select top 10 {bp.code},{ps.code},{ptl.ActionID},{ptl.startdate},{ptl.clusterid} from {AutoshipTemplateProcess as bp left join ProcessState as ps on {ps.pk}={bp.state} left join ProcessTaskLog as ptl on {ptl.process} ={bp.pk}} where {ptl.clusterid}='8' ORDER BY {ptl.startdate} DESC

EIS

* Configuration changes in EIS:
  + Max worker processes – 32
  + Startup/Shutdown time – 90 -> 10 secs
  + Recycle – 0
  + RFO db connections – 3 (web.config)