Mind-Alliance Performance Test Report

Date: March 23, 2012

SVN Revision: 3848

Hardware and Software details:

Environment	Parameter	Value
Channels	Hardware	Processor: Intel C2D@2.66Ghz Memory: 8194MiB Motherboard: DG31G Hard Disk: 250 GB
	Software	OS – Ubuntu Server 10.10
Load Machine Details	Hardware	Processor: Intel Core i3 CPU@3.02Ghz Memory: 4096 MB RAM Motherboard: INSPIRON N5110 Hard Disk: 500 GB
	Software	OS – Windows 7 Professional

Test Overview:

Test was conducted using JMeter for following scenarios with load details.

Sr. No	Scenario	Load (Thread)
1	About Plan-Details:-Planner A Updates the details of the Plan and if Planner B will update the same plan then the lock will be generated for the plan	20
2	About Plan-All Events:- Planner A will add event to the Plan and if Planner B will add event in the same plan then the lock will be generated for the plan	20
3	About Plan-All Organizations:- Planner A will add All Organizations to the Plan and if Planner B will add All Organizations in the same plan then the lock will be generated for the plan	20
4	About Plan Segment-Details:- Planner A Updates the details of the About Plan Segment and if Planner B will update the same Segment then the lock will be generated for the plan	20
5	About Plan Segment-Goal:- Planner A adds the Goal in the Segment and if Planner B will add goal the same Segment then the lock will be generated for the plan	20

6	About Plan - Update Task Details: Lock should be generated for user B for updating same task of about plan	20
7	Update Flow : Lock should be generated for user B for same updating flow	20
8	Phase: Lock should be generated for user B for same updating phase	20
9	Agents: Lock should be generated for user B For same updating Agent	20
10	Places: Lock should be generated for user B For same updating Places	20
11	Transmission Media: Lock should be generated for user B for same updating Transmission Media	20
12	Organization: Lock should be generated for user B for same updating Organizations	20
13	Roles: Lock should be generated for user B for same updating Roles	20

Note: As all the scenarios are running simultaneously hence whenever lock is release from the channel other user can perform its task.

E.g.:
Suppose all 9 scenarios are executing using 10 threads (users) per scenario.
Hence if Planner A of scenario 1 is updating the details of the plan then all other planners which are logged in to the channels will not access the plan till Planner A completes its task. It may happened that Planner A of scenario 1 release the lock when Planner 5 of scenario 2 is start executing, so it will add events for Planner 5 but for Planner 1,2,3 and 4 it will not add the events because of lock. So it is depending upon the Planner when it releases the lock and which one grabs it based on requests.

Similarly sometimes Scenario 3 grabs the lock and so on…

Load Details:
Total No of Threads (Users): 260

JVM size (Heap):

Parameter	**Ubuntu server**	**Load machine (Increased for Jmeter)**
Min	4096m	4096m
Max	4096m	4096m

Summary:
Refer attached files with this report for summary of tests conducted.

Observation:

- During test CPU utilization was between 80-85%.
- Observed that for some requests Std. deviation was above 2 sec (Refer SummaryReportFor260threads.ods file)

Test area	Actual Result	Expected Result	Comments
Update Plan	Not updated	Updated	Due to Load some components were not updated
Add Event	Not added	Added	
Update Media	Not updated	Updated	
Update Agent	Not updated	Updated	
Update Places	Not updated	Updated	
Update Organization	Updated	Updated	
Add Goal	Not added	Added	
Add Role	Updated	Updated	
Update Task	Not updated	Updated	
Add Flow	Not added	Added	
Update Segment	Updated	Updated	
Update Phase	Updated	Updated	

Please refer following files for more statistics.

Sr. No.	File Name	Comments
1	SummaryReportFor260threads.ods	Contains information about response time, stddev and throughput etc.
2	ResultStatusFor260Threads.ods	Status of each request with respect to thread (user).
3	Channels.log	Channels log
3	PerformanceCounterReportFor260Threads.ods	CPU, Memory and IO etc. performance counter.

It is observed that with increase in threads no. (User load) operations (request) were not completed