Assignment 1 Load Testing

- Download JMeter at <u>http://jmeter.apache.org/download_jmeter.cgi</u>
 Download the Binaries and extract them somewhere you won't forget.
- 2. Open up the ApacheJMeter.jar file under the bin folder
- 3. Open the A1LoadTest.jmx file.
- 4. The item called **Thread Group** sets up threads which will simulate concurrent users. Right now the settings are configuring 20 threads, with 0 ramp up and each thread will run 10 times.

🛲 A1LoadTest.jmx (C:\GAEDev\A1LoadTest.jmx) - Apache JMeter (2.7 r1342410)			
File Edit Search Run Options Help			
 Test Plan Thread Group HTTP Request ♥ View Results Tree ♥ Graph Results ₩orkBench 	Thread Group Name: Thread Group Comments: Action to be taken after a Sampler error Image: Thread Properties Number of Threads (users): 20 Ramp-Up Period (in seconds): 0 Loop Count: Forever Scheduler		

5. The **Http Request** item defines a request that is sent by each thread, right now we don't have a server setup. Assign the server to be **localhost** and the port to be **8888**.

📟 A1LoadTest.jmx (C:\GAEDev\	A1LoadTest.jmx) - Apache JMeter (2.7 r1342410)		
<u>File Edit Search Run Option</u>	ons <u>H</u> elp		
📑 실 🤒 🛃	🔏 🗊 🔁 🕂 — 🛷 🕨 👦 🔍	۵ 🔥 🗞 🙀 💥 🙈 🌜 🗐	?
Test Plan Thread Group HTTP Request View Results Tre Graph Results WorkBench	HTTP Request Name: HTTP Request Comments: Web Server Server Name or IP: localhost HTTP Request Implementation: Path: /SeatReservation Redirect Automatically Follow Redirects Usa Parameters Post Body	Port Number: 8888 Connect: Method: POST Content encoding: e KeepAlive Use multipart/form-data for POST Br	(millise
	Sen	nd Parameters With the Request:	
	Name:	Value	Encod
	FlightName		
FirstName		John	
	Detail Add A	dd from Clipboard Delete Up Down	
		Send Files With the Request:	

 So now we have each thread sending a Http Post request to localhost:8888/SeatReservation. The post data is defined under the parameters tab. The SeatReservation url points to TestSeatReservation.java. It is expecting three parameters, FlightName, FirstName, and LastName. We have two of those assigned, but FlightName is blank. To get the correct flight name, run the project and up localhost:8888.



 Click on any seat, and in the URL you will find the FlightName. In my case the URL is <u>http://localhost:8888/ReserveSeat?FlightName=aglub19hcHBfaWRyEQsSBkZsaWdodCIFRjI0OTE</u> <u>M&SeatID=10A</u>

So the Flight Name is aglub19hcHBfaWRyEQsSBkZsaWdodCIFRjI0OTEM, copy your FlightName and input it into the JMeter Post parameters. (note there are no = or & characters in the FlightName)

	AILUAUTESCIMA) - Apache Umeter (2.7 FI342410)		
<u>F</u> ile <u>E</u> dit Search <u>R</u> un <u>O</u> ptio	ons <u>H</u> elp		
	X [] [] + - 4	🔈 🗞 🚳 🆋 🌌 🙈 🏷 📰 🛛	
 	HTTP Request		
View Results Tre	Commenter		
Graph Results	Graph Results		
WorkBench Timeouts			
	Server Name or IP: localhost	Port Number: 8888 Connect:	
	HTTP Request		
	Implementation: Protocol [http]:	Method: POST Content encoding:	
	Path: /SeatReservation		
	Redirect Automatically 🔽 Follow Redirects 🗹 Use Ke	eepAlive 🛛 Use multipart/form-data for POST 🗌 Browse	
	Parameters Post Body		
	Send P	arameters With the Request:	
	Name:	Value Enco	
	FlightName agi	lub19hcHBfaWRyEQsSBkZsaWdodCIFRjI0OTEM	
	FirstName Joh	nn L	
5	I ASINALLE UN		

8. Just one more thing to do before we run the test, make sure you free up all the seats. This way you simulate more requests that end up going to the datastore. Requests to seats that are taken do not end up going to the datastore. We need to test the load with datastore interaction. So

Image: Seats Image: Seats Image: Seats Image: Seats
 C O ocalhost:8888/FreeSeats GAE MWC Azure S Easier Is Better Than S Elements Of A Viral La S Useful Ideas And Gui
🗀 GAE 🦳 MVC 🦳 Azure 🔀 Easier Is Better Than 💦 Elements Of A Viral La 🔊 Useful Ideas And Gui
Free Seats
Seats have been freed.

9. Now run the JMeter test, click on the toolbar. Once the test is complete you should see something like this on the **Graph Results**

📟 A1LoadTest.jmx (C:\GAEDev\A1LoadTest.jmx) - Ај	pache JMeter (2.7 r1342410)	<u>×</u>
<u>File Edit Search Run Options Help</u>		
	+ - 4	126 🔔 🛛 🛛 🖓
P ▲ Test Plan P Im Thread Group	Graph Results	
HTTP Request	Name: Graph Results	
View Results Tree	Comments:	
Graph Results	Write results to file / Read from file	
	Filename Browse Log/Display Only: Errors Succes	ses Configure
	Graphs to Display 🗹 Data 🔽 Average 🗹 Median 🔽 Deviation 🔽 Throughput	
	638 ms	
	and the second se	
	They and man all the second	
	1 San Anis Martine	
	a state of the second stat	
	and the second sec	
	- Martin - Ma	
	0 ms (•
	No of Samples 200 Latest Sample 11 Average 311 Deviation 222 Throughput 3.356.643(minute Median 266	
	in oughput 3,330,943/minute Metian 200	

10. So my test ran with 126 errors, and at a throughput rate of 3356 requests per minute. If you look under **View Results Tree.** You will see each request. The failed requests give you this

🖴 A1LoadTest.jmx (C:\GAEDev\A1LoadTest.jmx) - Apache JMeter (2.7 r1342410)			
Elle Edit Search Run Options Help			
♀ ▲ Test Plan ♀ ■ Thread Group	View Results Tree		
HTTP Request	Name: View Results Tree		
View Results Tree	Comments:		
- 🔄 Graph Results	Write results to file / Read from file		
VvorkBench	Filename Browse Log/Display Only: Errors Successes Configure		
	HTTP Request Sampler result Request Response data		
	HTTP Request Inread Name: Inread Group 1-17 Comple Chat 2010 (5 12 10:20:20 RDT		
	A TIT Request		
	A HTTP Request		
	Size in bytes: 5342		
	Headers size in bytes: 222		
	Body size in bytes: 5120		
	A HTTP Request Sample Count: 1		
	Error Count 1		
	Response code: 500		
	HTTP Request HTTP Request HTTP and the sponse message: too much contention on these datastore entities, please try again.		
	- A HTTP Request		
	Content-Type: text/html: charset=iso-8859-1		
	Cache-Control: must-revalidate.no-cache.no-store		
	Content-Length: 5120		
	A HTTP Request Server: Jetty(6.1.x)		
	A HTTP Request		
	HTTP Request		
	HTTP Request HTTPSampleResult fields:		
	A HITP Request Content type: tex/thtml; charset=iso-8859-1		
	Latencoding: Iso-8859-1		
	Text		
	Scroll automatically? Raw Parsed		

response message, "too much contention on these datastore entities"

11. To run this test with the project running on Google, modify the HTTP Request to point to ubc-417-{chosenID}-a1.appspot.com, don't specify a port, and make sure you have the correct FlightName. Also make sure you've created the data on the server by going to <u>http://ubc-417-{chosenID}.a1.appspot.com/Prep</u>, also make sure the seats are free by going to <u>http://ubc-417-{chosenID}.ai.appspot.com/FreeSeats</u>

HTTP Request			
Name: HTTP Request			
Comments:			
r Web Server r Timeouts (milliseconds)			
Server Name or IP: ubc-417-123456789-a1.appspot.com Port Number: Connect: Response:			
□ HTTP Request			
Implementation: Protocol [http]: Method: POST Content encoding:			
Path: //SeatReservation			
🗌 Redirect Automatically 📝 Follow Redirects 📝 Use KeepAlive 🗌 Use multipart/form-data for POST 🔲 Browser-compatible head			
Parameters Post Body			
Send Parameters With the Request:			
Name:	Value	Encode? Include Equal	
FlightName	aglub19hcHBfaWRyEQsSBkZsaWdodCIFRjI0OTEM		
FirstName John 🗌 🖌			
LastName Doe 🗌 🔽			
Detail Add Add from Clipboard Delete Up Down			

12. Please note that running these tests targeted to your project on Google will use up your quotas. Make sure you project works locally before testing it on Google. You can deplete your quotas by running this test too many times and you will not be able to continue until the following day.