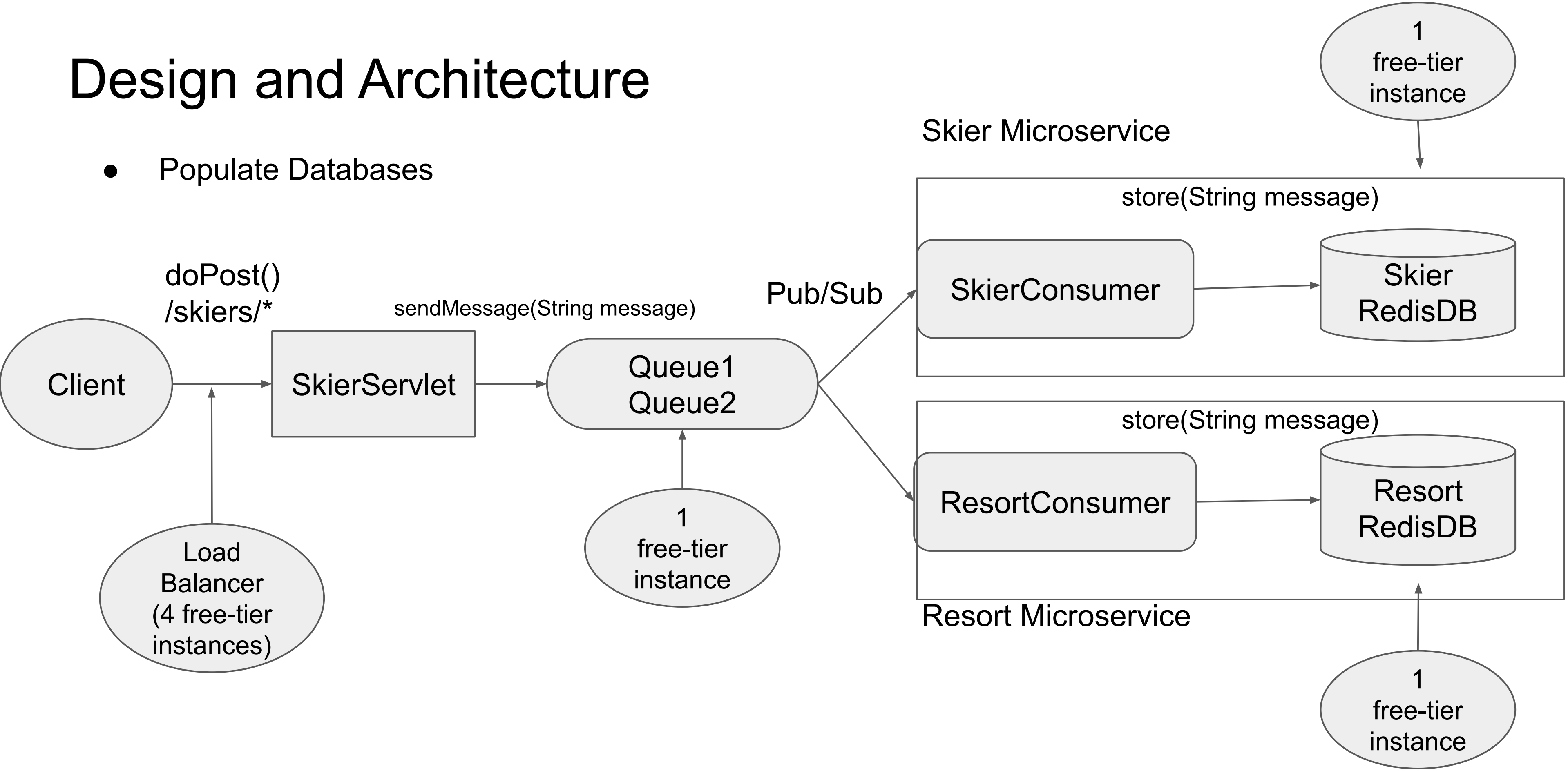
Team [in]Security Group Issues

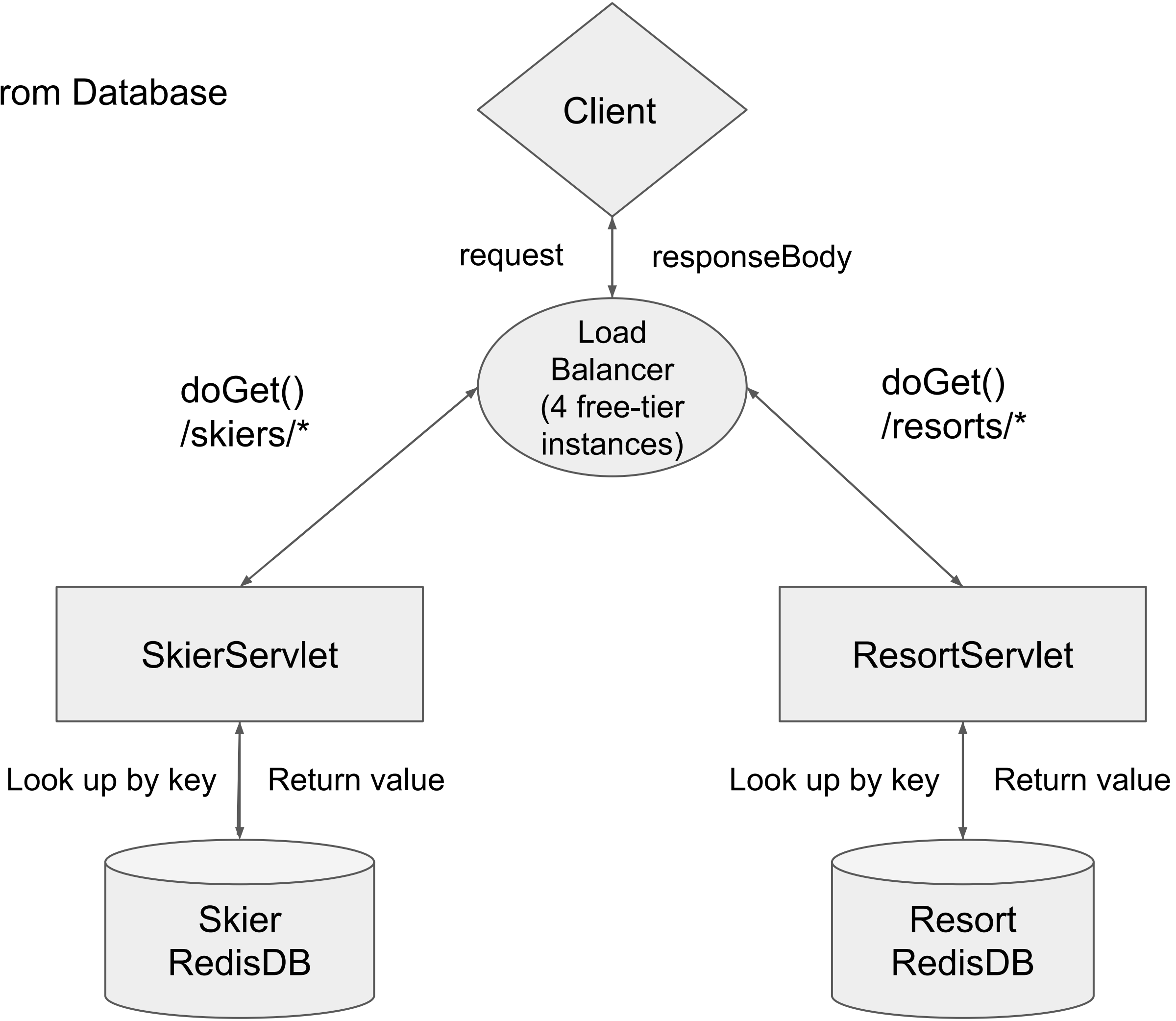
By: Sruti Keerti Munukutla & Kayla Sear

Design and Architecture

- Populate Databases



- Read from Database



Data Model

GET/skiers/{skierId}/vertical

Data stored in a hashmap: With key,field,value in below format:

Key: skierID	Field: “day-”+ dayId	Value: cumulative skier total vertical

- Get all values for skierId and add the vertical totals.
- Uses hincr() to update vertical when storing data.

GET/skiers/{resortID}/seasons/{seasonID}/days/{dayID}/skiers/{skierID}

- Get the total vertical for skierID for the given dayID
- Resort ID: 10, Season ID: 2022 (constant)

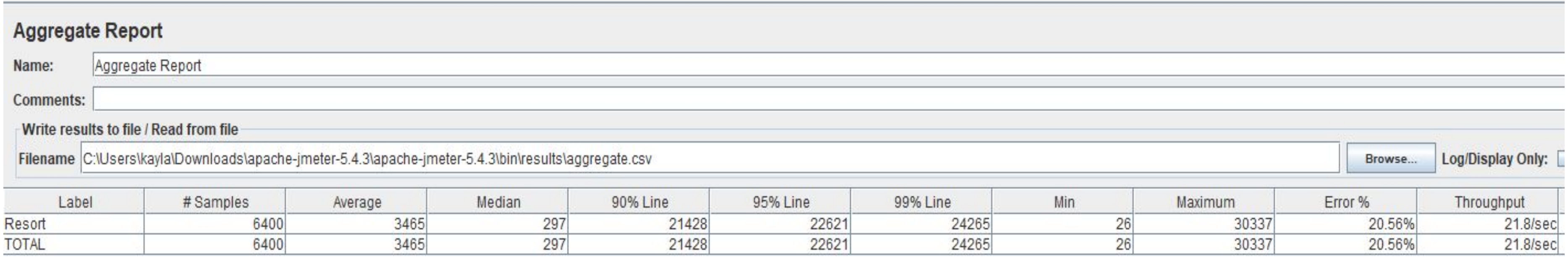
Data Model

GET/resorts/{resortID}/seasons/{seasonID}/day/{dayID}/skiers

Set name: “resort-1-day-1”	{34, 45, 243, 6543}

- To get the unique skiers for the resort on the day, get members by key using smembers()
- return the size of the result set.

Results - GET/resorts/{resortID}/seasons/{seasonID}/day/{dayID}/skiers



Aggregate Report

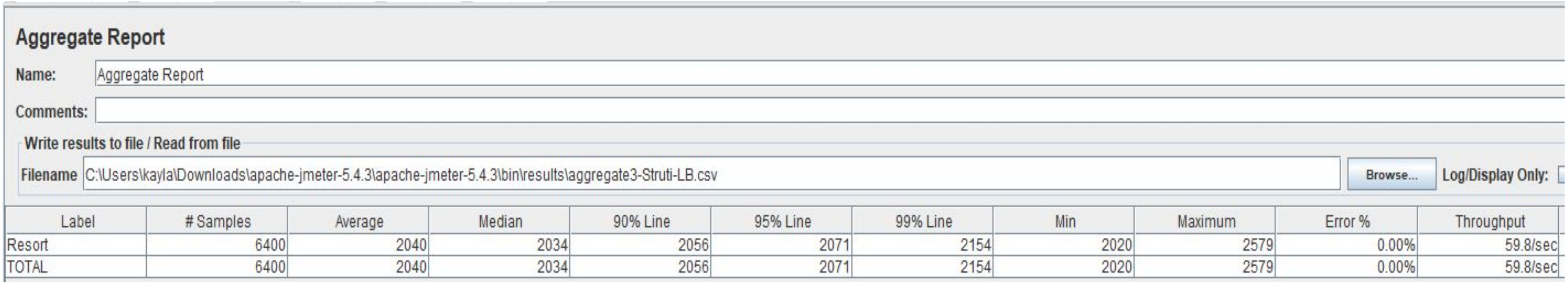
Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename C:\Users\kayla\Downloads\apache-jmeter-5.4.3\apache-jmeter-5.4.3\bin\results\aggregate.csv

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput
Resort	6400	3465	297	21428	22621	24265	26	30337	20.56%	21.8/sec
TOTAL	6400	3465	297	21428	22621	24265	26	30337	20.56%	21.8/sec



Aggregate Report

Name: Aggregate Report

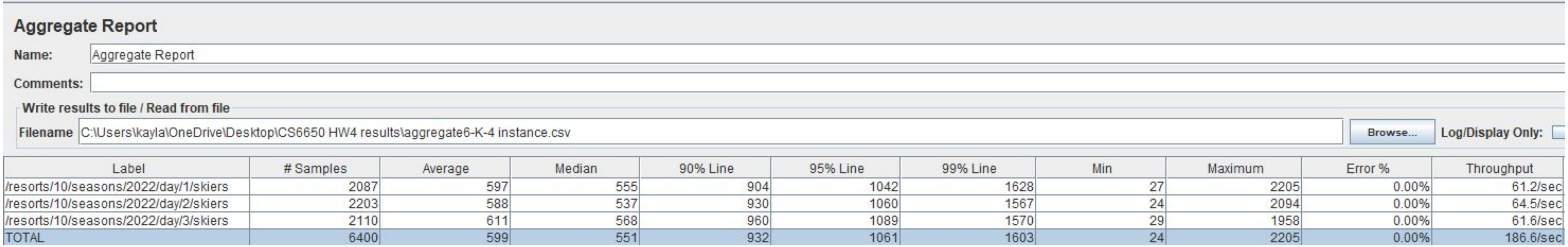
Comments:

Write results to file / Read from file

Filename C:\Users\kayla\Downloads\apache-jmeter-5.4.3\apache-jmeter-5.4.3\bin\results\aggregate3-Struti-LB.csv

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput
Resort	6400	2040	2034	2056	2071	2154	2020	2579	0.00%	59.8/sec
TOTAL	6400	2040	2034	2056	2071	2154	2020	2579	0.00%	59.8/sec

Results - GET/resorts/{resortID}/seasons/{seasonID}/day/{dayID}/skiers



Aggregate Report

Name: Aggregate Report

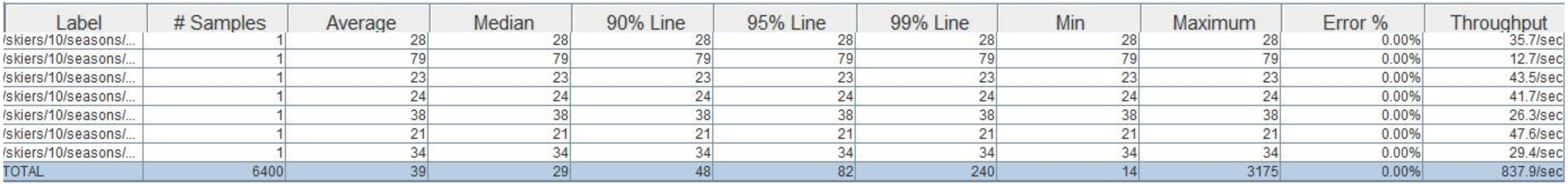
Comments:

Write results to file / Read from file

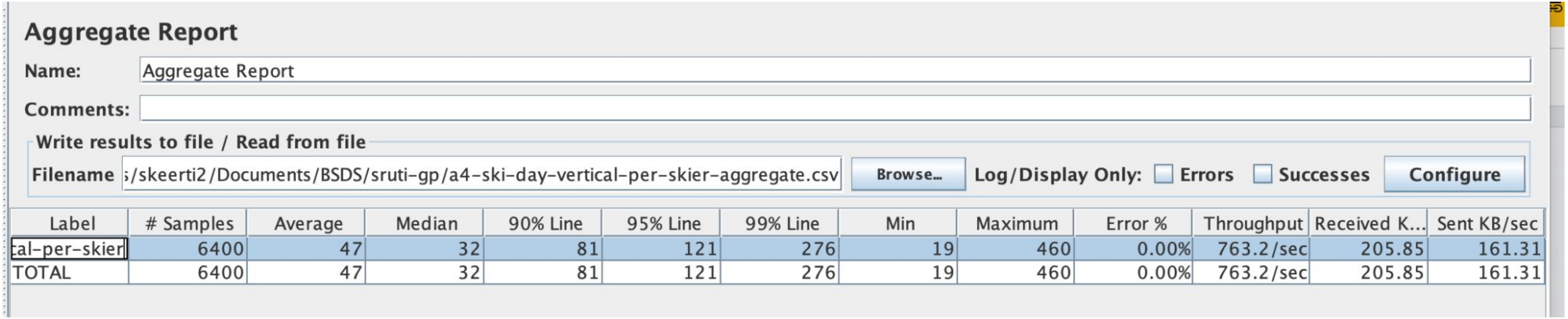
Filename C:\Users\kayla\OneDrive\Desktop\CS6650 HW4 results\aggregate6-K-4 instance.csv

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput
/resorts/10/seasons/2022/day/1/skiers	2087	597	555	904	1042	1628	27	2205	0.00%	61.2/sec
/resorts/10/seasons/2022/day/2/skiers	2203	588	537	930	1060	1567	24	2094	0.00%	64.5/sec
/resorts/10/seasons/2022/day/3/skiers	2110	611	568	960	1089	1570	29	1958	0.00%	61.6/sec
TOTAL	6400	599	551	932	1061	1603	24	2205	0.00%	186.6/sec

Results - GET/skiers/{resortID}/seasons/{seasonID}/days/{dayID}/skiers/{skierID}



Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput
/skiers/10/seasons/...	1	28	28	28	28	28	28	28	0.00%	35.7/sec
/skiers/10/seasons/...	1	79	79	79	79	79	79	79	0.00%	12.7/sec
/skiers/10/seasons/...	1	23	23	23	23	23	23	23	0.00%	43.5/sec
/skiers/10/seasons/...	1	24	24	24	24	24	24	24	0.00%	41.7/sec
/skiers/10/seasons/...	1	38	38	38	38	38	38	38	0.00%	26.3/sec
/skiers/10/seasons/...	1	21	21	21	21	21	21	21	0.00%	47.6/sec
/skiers/10/seasons/...	1	34	34	34	34	34	34	34	0.00%	29.4/sec
TOTAL	6400	39	29	48	82	240	14	3175	0.00%	837.9/sec



Aggregate Report

Name: Aggregate Report

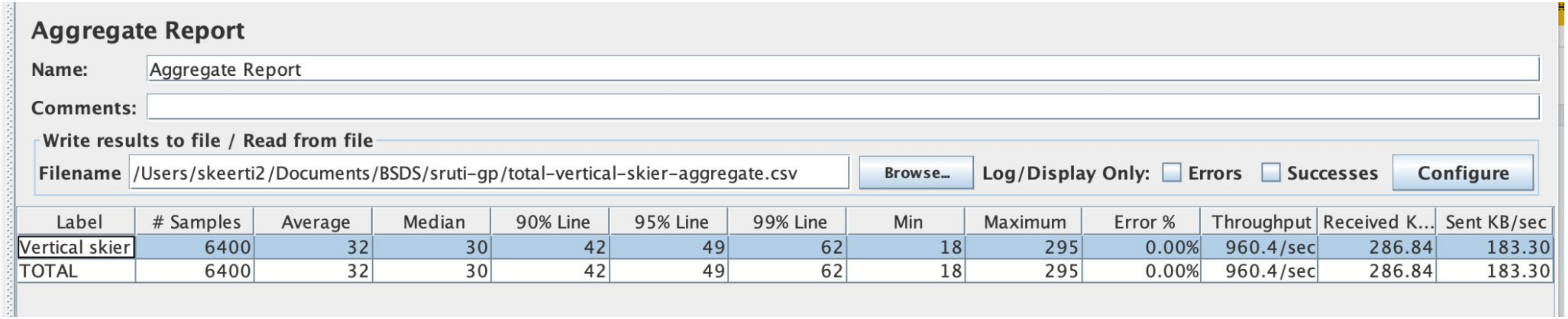
Comments:

Write results to file / Read from file

Filename s/skeerti2/Documents/BSDS/sruti-gp/a4-ski-day-vertical-per-skier-aggregate.csv

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received K...	Sent KB/sec
tal-per-skier	6400	47	32	81	121	276	19	460	0.00%	763.2/sec	205.85	161.31
TOTAL	6400	47	32	81	121	276	19	460	0.00%	763.2/sec	205.85	161.31

Results - GET/skiers/{skierID}/vertical



Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename /Users/skeerti2/Documents/BSDS/sruti-gp/total-vertical-skier-aggregate.csv

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received K...	Sent KB/sec
Vertical skier	6400	32	30	42	49	62	18	295	0.00%	960.4/sec	286.84	183.30
TOTAL	6400	32	30	42	49	62	18	295	0.00%	960.4/sec	286.84	183.30

Future Improvements

- Change data structure of the Resort data model for faster reads - increase throughput.
- Change data models to include resortId and seasonId
- Use bigger instances to increase throughput, however cost $$ - add another free instance to LB?
- Implement a cache for frequent reads

Thank you!