Created by Team ***Catalyst***

Final Project Presentation

Linni Cai, Shengnan You, Furong Tian

April 26, 2022

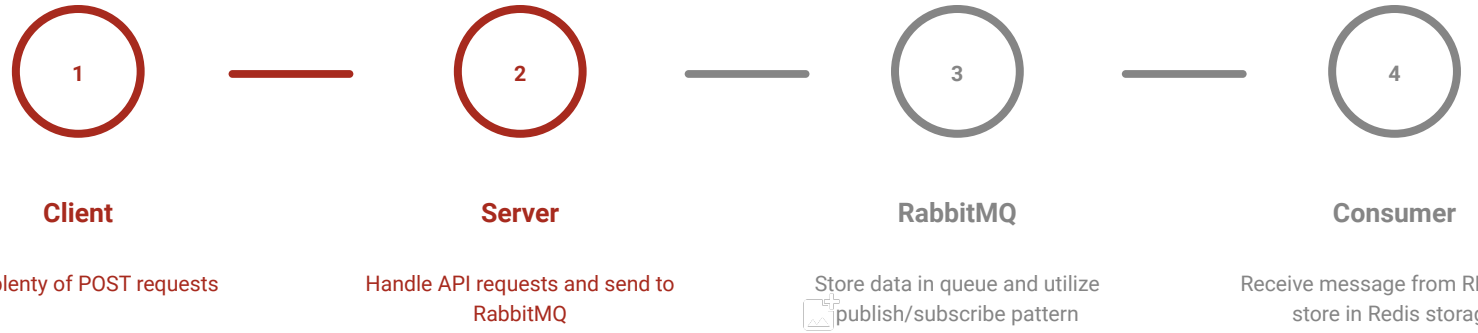
Northeastern University

Project Overview:

- Client: run POST three times with varied day id from 1, 2, 3 with the following arguments
  - num_threads : 256
  - num_skiers: 20000
  - num_lifts: 40
- Send data to server and publish data to queue
- Three consumers subscribe and receive data from queue, store the data into Redis storage
- Server implemented three GET requests

Project Deployment:

- One Linux instance for server
- One Ubuntu instance for RabbitMQ
- Three Linux instances for consumers (skier, skier2, resort) with Redis



Data Model for Redis Storage

- Use Pipeline to do transaction lock
- Resort: hash set
  - Key: a constant string NUM_SKIERS
  - Field Name: ***resortID-seasonID-dayID***
  - Field Value: the number of skiers
- Skier: hash set
  - Key: a constant string TOTAL_VERTICAL
  - Field Name: ***resortID-seasonID-dayID-skierID***
  - Field Value: the total vertical
- Skier2: hash set
  - Key: skierID
  - Field Name: ***seasonID***
  - Field Value: the total vertical

3 GET Requests:

- Skier Total Vertical:
  - GET **/skiers/{resortID}/seasons/{seasonID}/days/{dayID}/skiers/{skierID}**
  - get the total vertical for the skier for the specified ski day
- Skier Total Vertical Result List:
  - GET **/skiers/{skierID}/vertical**
  - get the total vertical for the skier for specified seasons at the specified resort
- Resort:
  - GET **/resorts/{resortID}/seasons/{seasonID}/day/{dayID}/skiers**
  - get number of unique skiers at resort/season/day

Improvement in the future:

- Add configuration automation, such as IP address synchronize
- Try different data storage instead of Redis
- Try message queue in Redis instead of RabbitMQ

Client POST Statistics

Data for Day 3

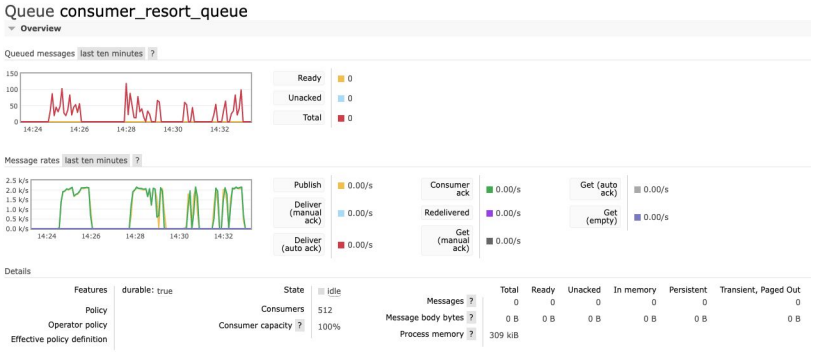
------ Client Post Statistics ------

------ PART 1 ------
number of successful requests sent: 160420
number of unsuccessful requests: 0
the total run time for all phases to complete: 96045
the total throughput in requests per second: 1000

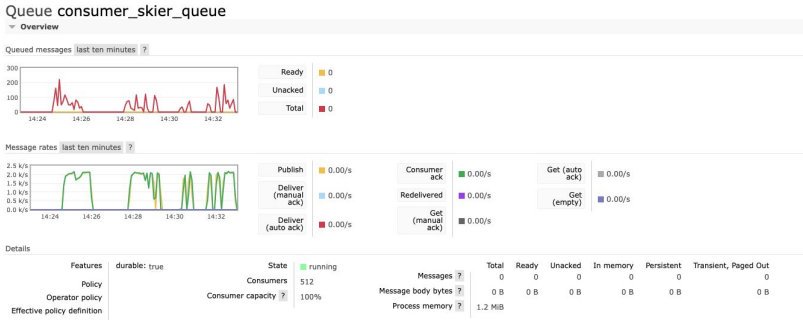
------ PART 2 ------
mean response time (millisecs): 115
median response time (millisecs): 32
throughput: 1000
p99 (99th percentile) response time: 2021
min response time (millisecs): 12
max response time (millisecs): 9601

Skier Total Vertical Result List Queue

Skier Total Vertical Queue

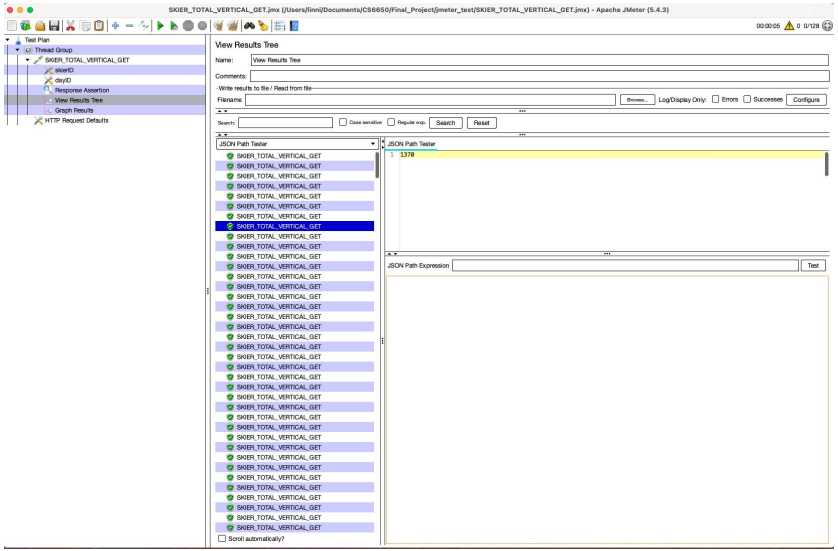


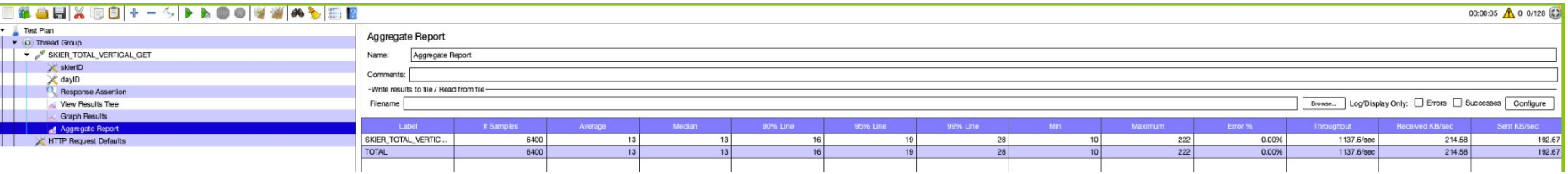
Resort Queue



JMeter Test Result

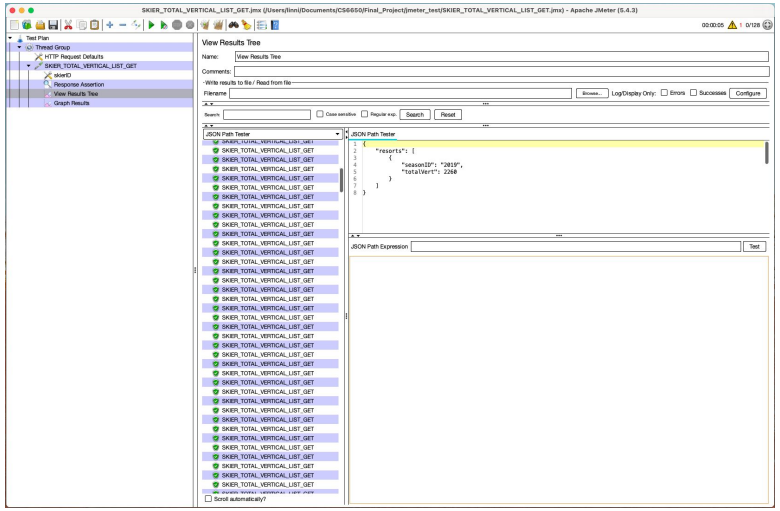
Skier Total Vertical GET

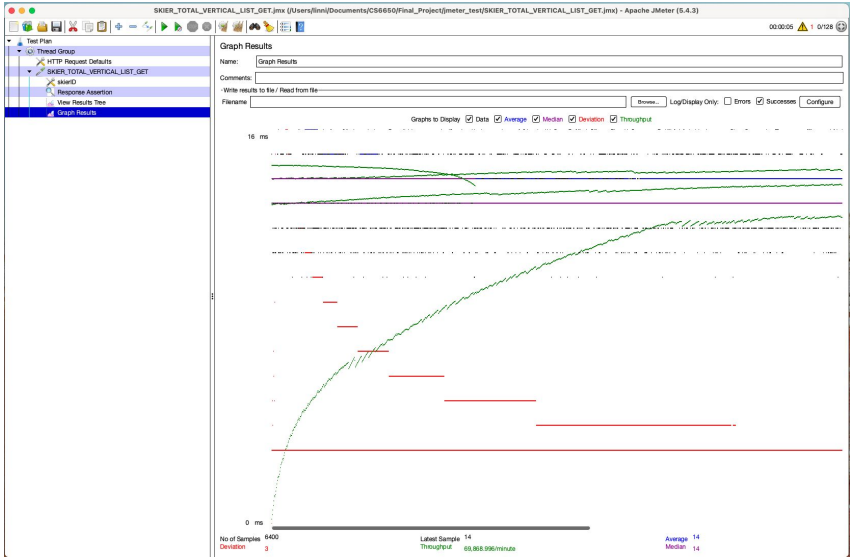


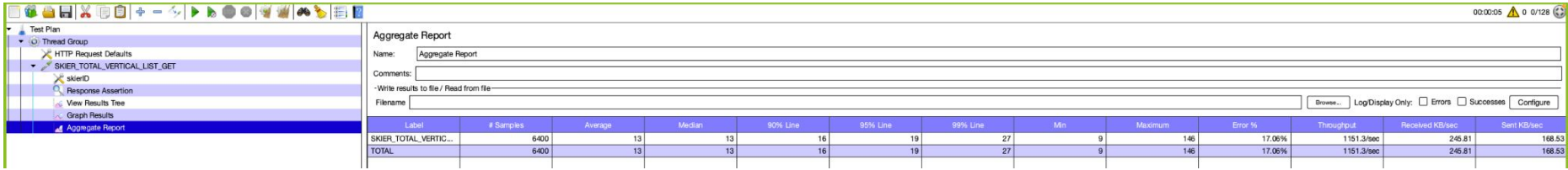


Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
SKIER_TOTAL_VERTIC...	6400	13	13	16	19	28	10	222	0.00%	1137.6/sec	214.58	192.67
TOTAL	6400	13	13	16	19	28	10	222	0.00%	1137.6/sec	214.58	192.67

Skier Total Vertical Result List GET

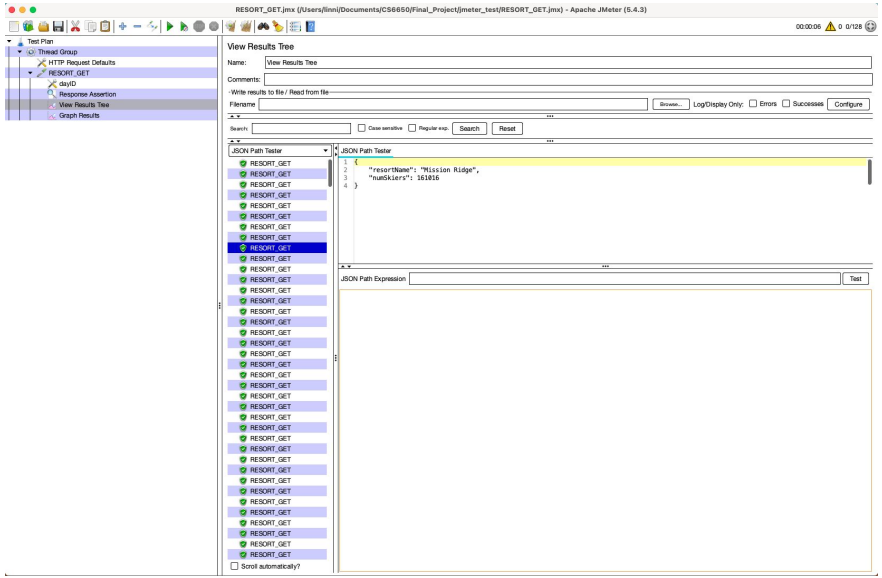


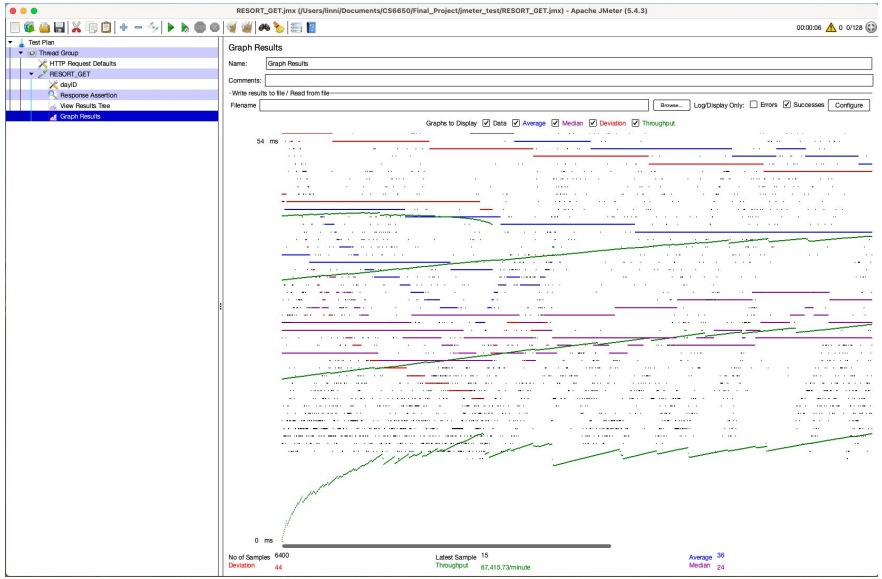


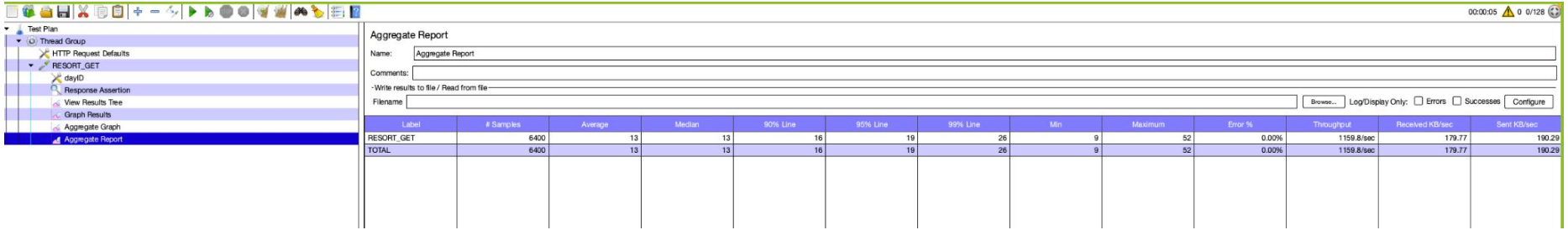


Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
SKIER_TOTAL_VERTIC...	6400	13	13	16	19	27	9	146	17.06%	1151.3/sec	245.81	168.53
TOTAL	6400	13	13	16	19	27	9	146	17.06%	1151.3/sec	245.81	168.53

Resort GET







Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
RESORT_GET	6400	13	13	16	19	26	9	52	0.00%	1159.8/sec	179.77	190.29
TOTAL	6400	13	13	16	19	26	9	52	0.00%	1159.8/sec	179.77	190.29


Thank you