15-319 / 15-619
Cloud Computing

Recitation 8

March 1st, 2016

Overview

- **Administrative issues**
  Office Hours, Piazza guidelines
- **Last week's reflection**
  Project 3.1, OLI Unit 3, Module 13, Quiz 6
- **This week's schedule**
  - 15619 Project - Query 1 - Bonus 3/2
  - Quiz 7 - March 3rd (Unit 4, Module 14)
  - Project 3.2 - March 6th

Last Week : A Reflection

- Content, Unit 3 - Module 13:
  - Storage and Network Virtualization
  - Quiz 6 completed

- P3.1: You explored data storage solutions
  - Files
  - Databases (SQL & NoSQL)
    - MySQL
    - HBase
  - Benchmarked vertical scaling performance

This Week: Content

UNIT 4: Cloud Storage

- Module 14: Cloud Storage
  - Quiz 7 - Introduction to Cloud Storage
    - **Thursday, March 3rd**

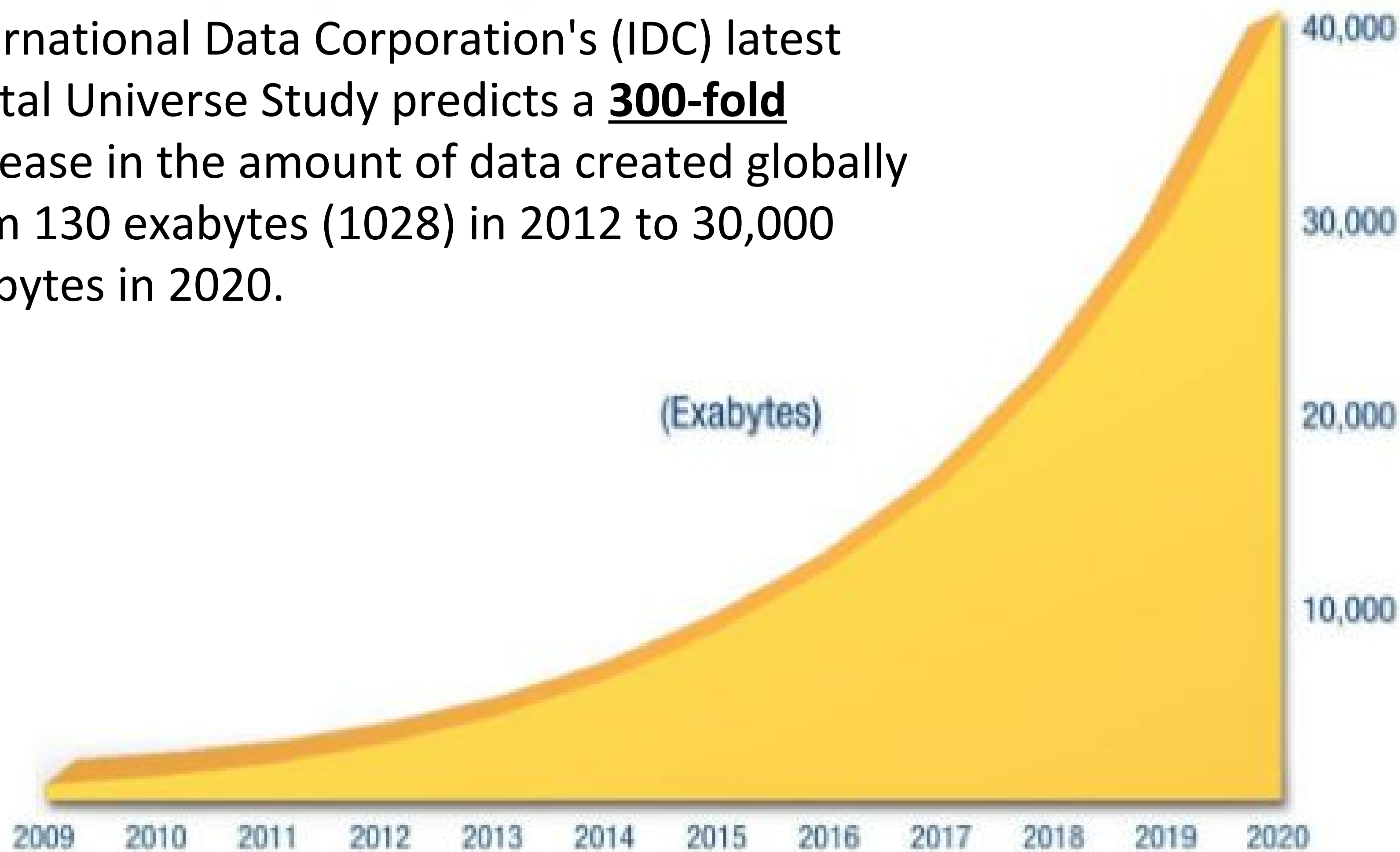
- Module 15: Case Studies: Distributed File Systems
  - Quiz 8: Distributed File Systems Checkpoint
- Module 16: Case Studies: NoSQL Databases
- Module 17: Case Studies: Cloud Object Storage
  - Quiz 9: NoSQL and Object Stores

Project 3 Weekly Modules

- P3.1: Files, SQL and NoSQL
- **P3.2: Sharding and Replication**
  - **Sunday, March 6th**
- P3.3: Consistency
- P3.4: Social network and heterogeneous back end storage
- P3.5: Data warehousing and OLAP

P3.2: Motivation - Scale

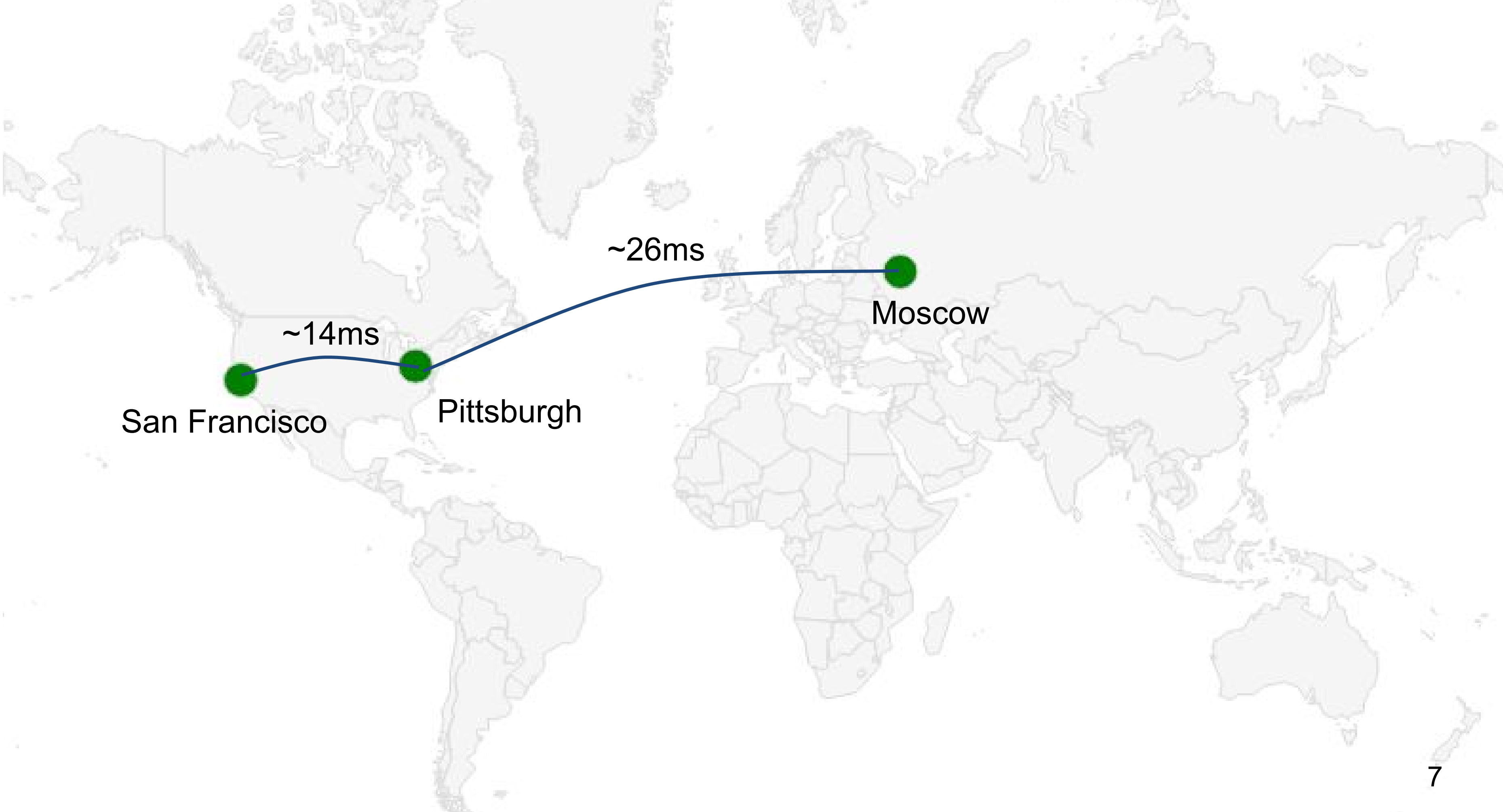
International Data Corporation's (IDC) latest Digital Universe Study predicts a **300-fold** increase in the amount of data created globally from 130 exabytes (1028) in 2012 to 30,000 exabytes in 2020.



Source: IDC's Digital Universe Study, sponsored by EMC, December 2012

P3.2: Motivation - Global Users

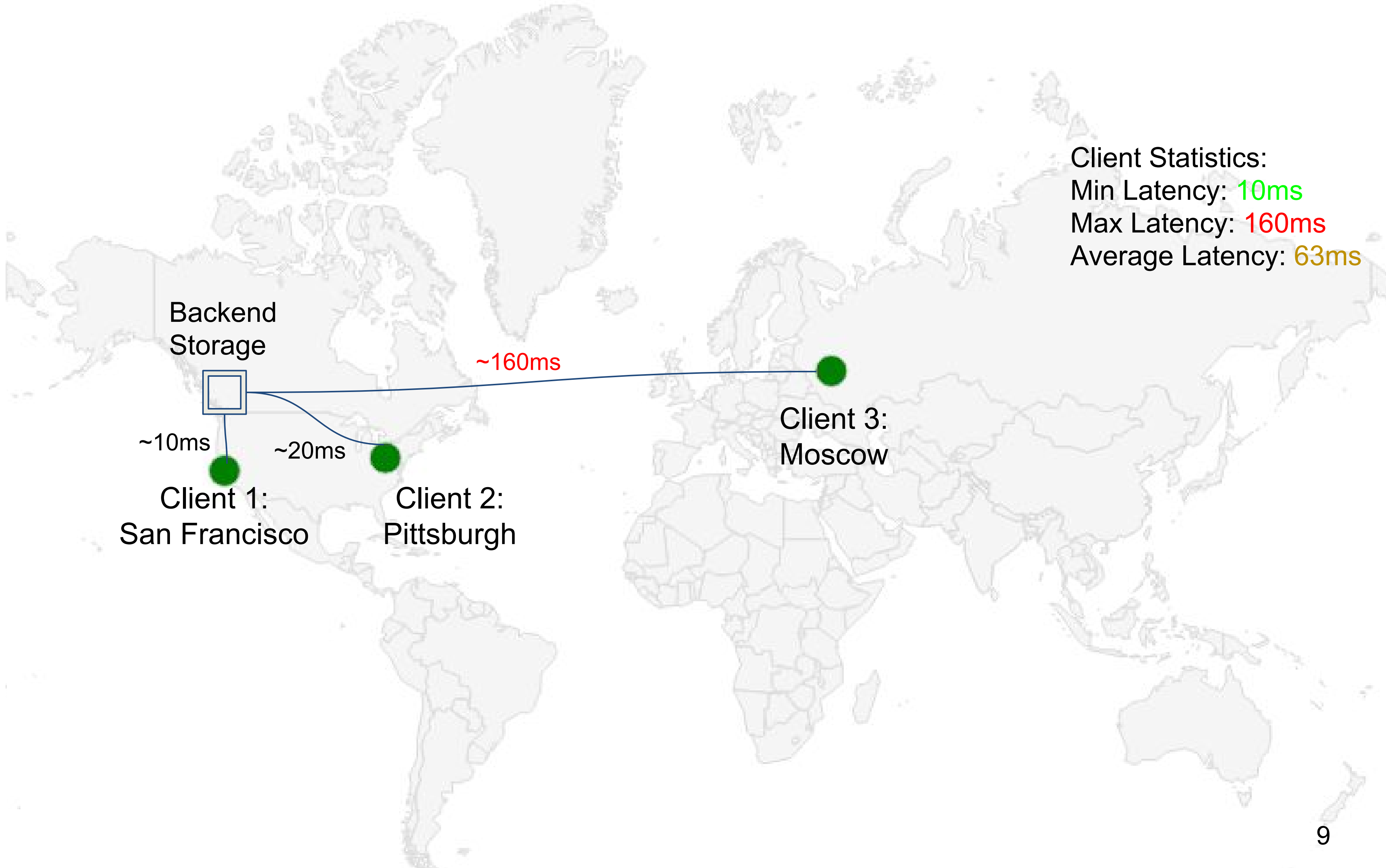
- Speed of Light (≈3.00×10^8 m/s)
- Inherent latencies



P3.2: Motivation - Requirements

- Typical end-to-end latency
  - Client request
  - Network latency (to backend)
  - Server response
    - Includes fetching and processing data from backend
  - Network latency (from backend)
  - Client response

P3.2: Motivation - Latency



Client Statistics:
Min Latency: 10ms
Max Latency: 160ms
Average Latency: 63ms

P3.2: Motivation - Replicas

Client Statistics:
Min Latency: 10ms
Max Latency: 20ms
Average Latency: 13.3ms

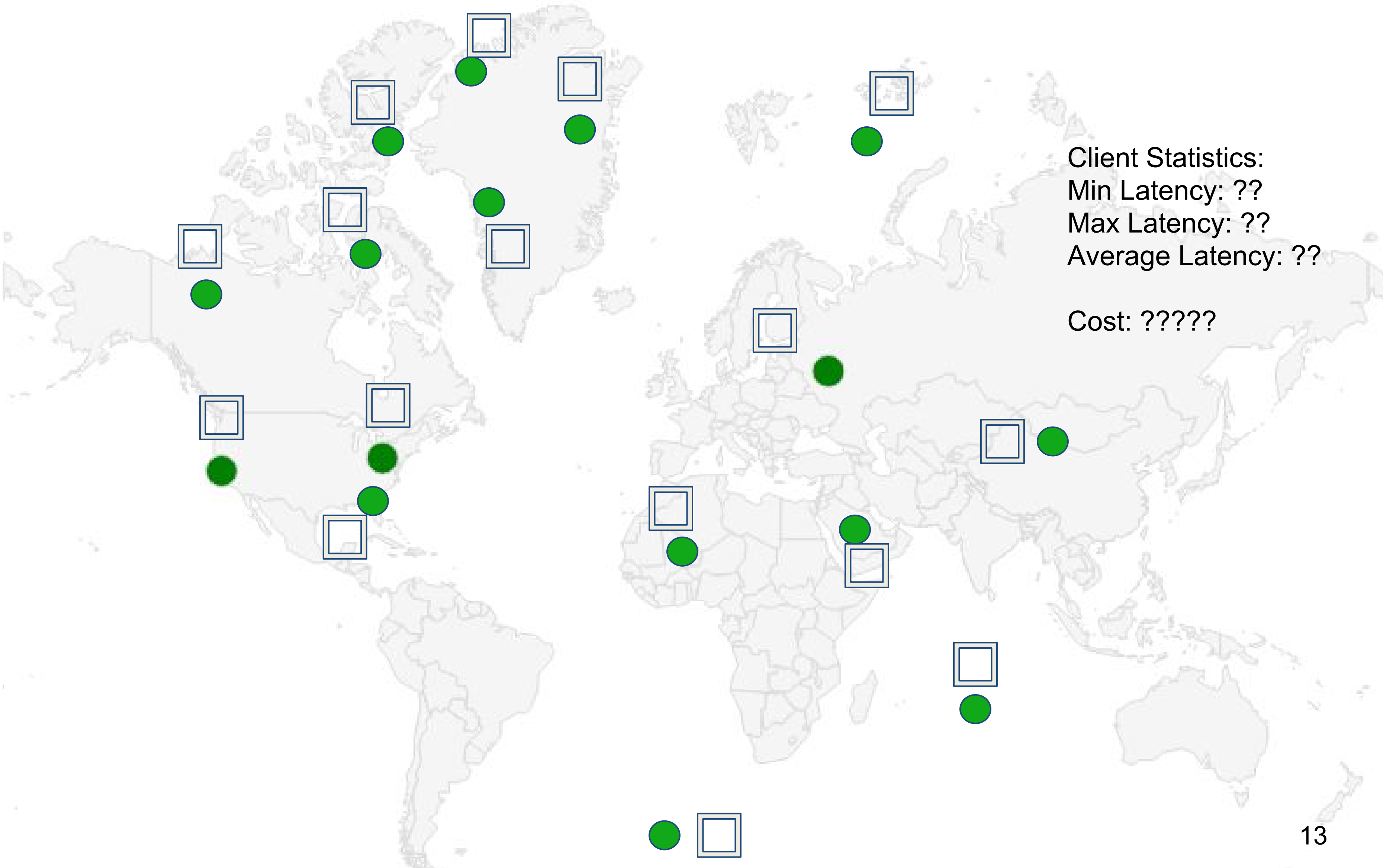
P3.2: Motivation - Replicas cont

Client Statistics:
Min Latency: 10ms
Max Latency: 10ms
Average Latency: 10ms

P3.2: Motivation - Replicas

- As you can see, by adding replicas to strategic locations in the world, we can greatly reduce the latency seen by outlying clients
- Each added datacenter decreases the average latency seen by all clients
- But at what cost?

P3.2: Motivation - Replicas?



Client Statistics:
Min Latency: ??
Max Latency: ??
Average Latency: ??

Cost: ?????

P3.2: Motivation - Replicas

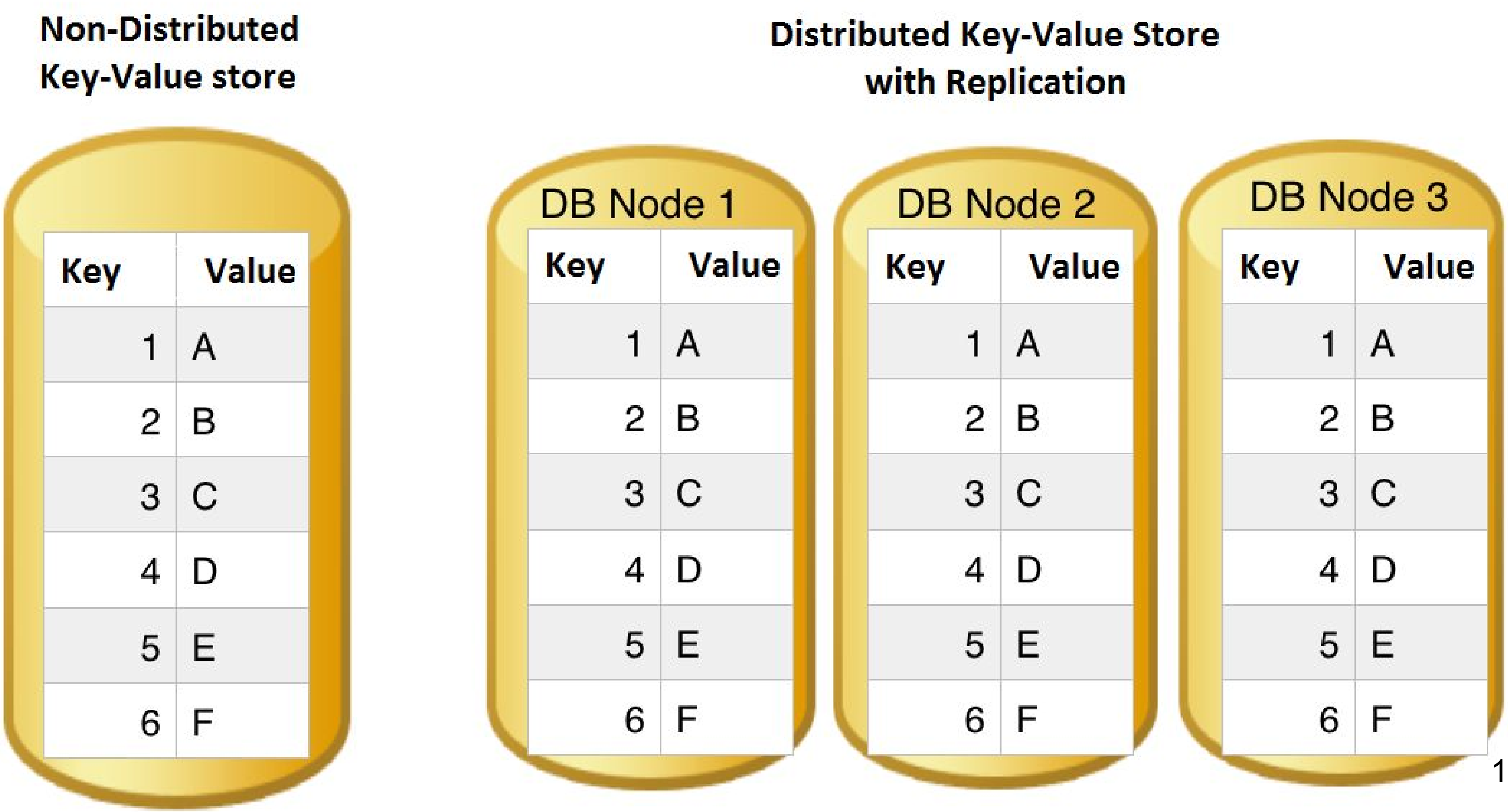
- There are downsides to adding additional replicas
  - Cost for new data center to host the replica
  - May increase latency for write requests during replication, because we need to update every single replica
- As you add more replicas throughout the world, the benefit per replica tapers off

P3.2: Motivation - Requirements

- Inherent dynamicity of data
- Large scale data intensive systems
  - Many reads/ writes impact throughput performance
    - Hot records
  - Global reach creates latency issues
  - Expectation to be 100% available
- Possible Solutions
  - Vertically scale storage backend
  - Replication
  - Sharding

P3.2: Replication (1)

- Duplicate the data across multiple instances and/or locations



P3.2: Replication (2)

- Duplicate the data across multiple instances
- Advantages
  - Fetching data can be faster
    - Data can be retrieved from any datastore
    - System can handle failures of nodes
- Disadvantages
  - Requires more storage capacity
  - Updates are slower
    - Changes must reflect on all datastores

P3.2: Consistency

- Data consistency across replicas is important (More on consistency models in P3.3 next week)
  - Strict, Strong (Linearizability), Sequential, Causal and Eventual Consistency
- This Week - Strong Consistency
  - All datastores must return the same value for a key
  - The order in which the values are updated must be preserved

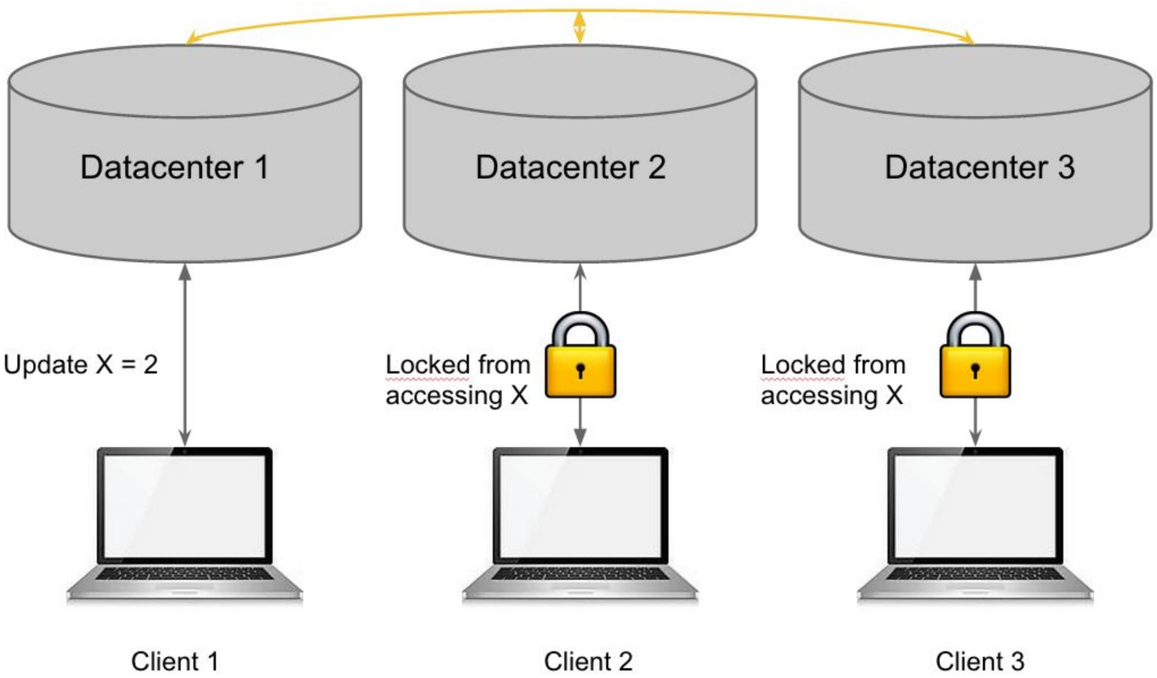
P3.2: Strong Consistency

Single PUT request for key 'X'

- Block all GET for key 'X' until all datastores are updated
- GET requests for a different key 'Y' must be allowed

Multiple PUT requests for 'X'

- Resolved in order of their arrival
- Any GET request in between 2 PUTs must return the last PUT value

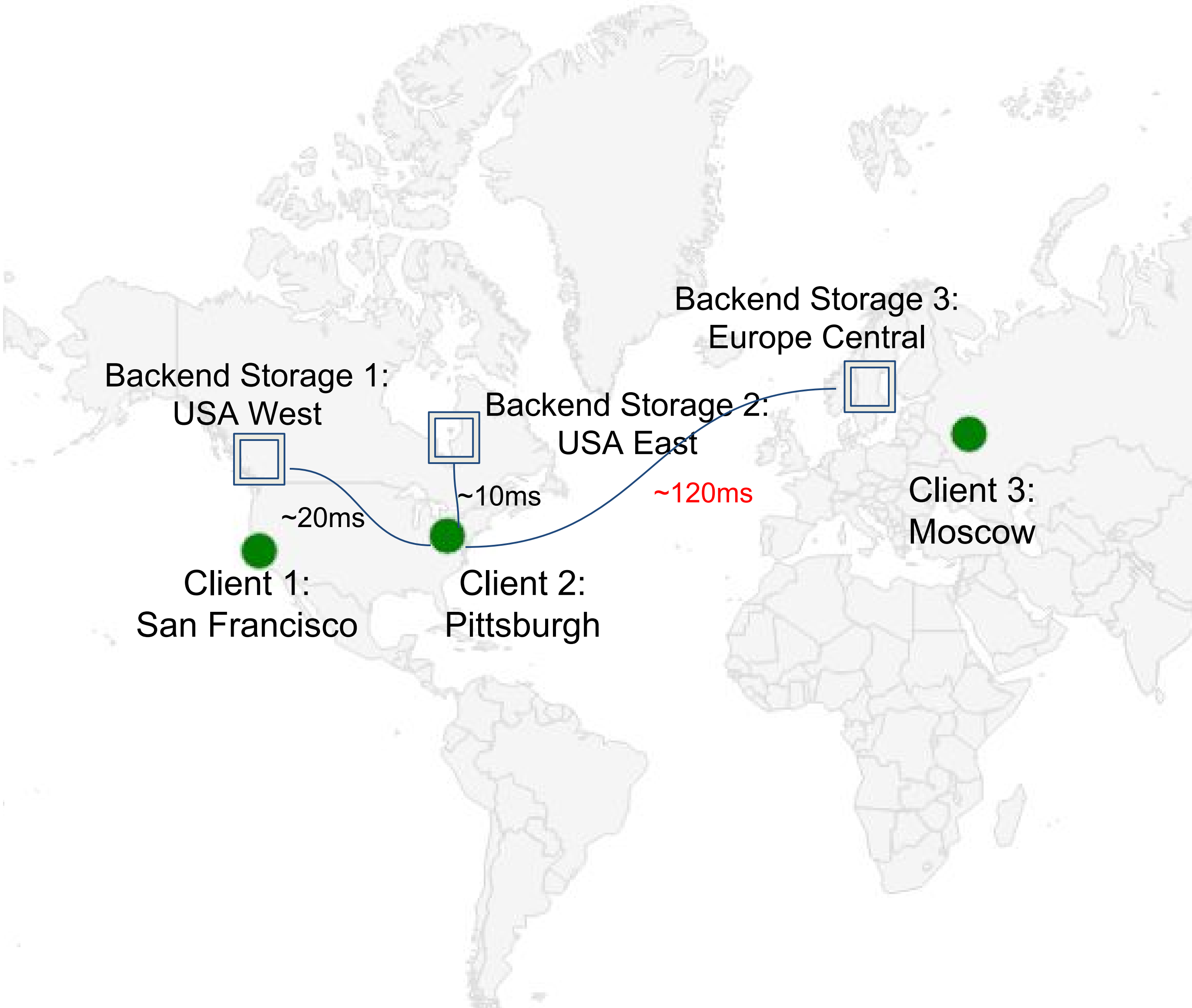


P3.2: Motivation - Replication READ

Read Operation:

Min Latency: 10ms
Max Latency: 10ms
Average Latency: 10ms

P3.2: Motivation - Replication WRITE



Write Operation:

Latency for Client 2 =
MAX(10ms, 20ms, 120ms)
= 120ms

Same across all clients

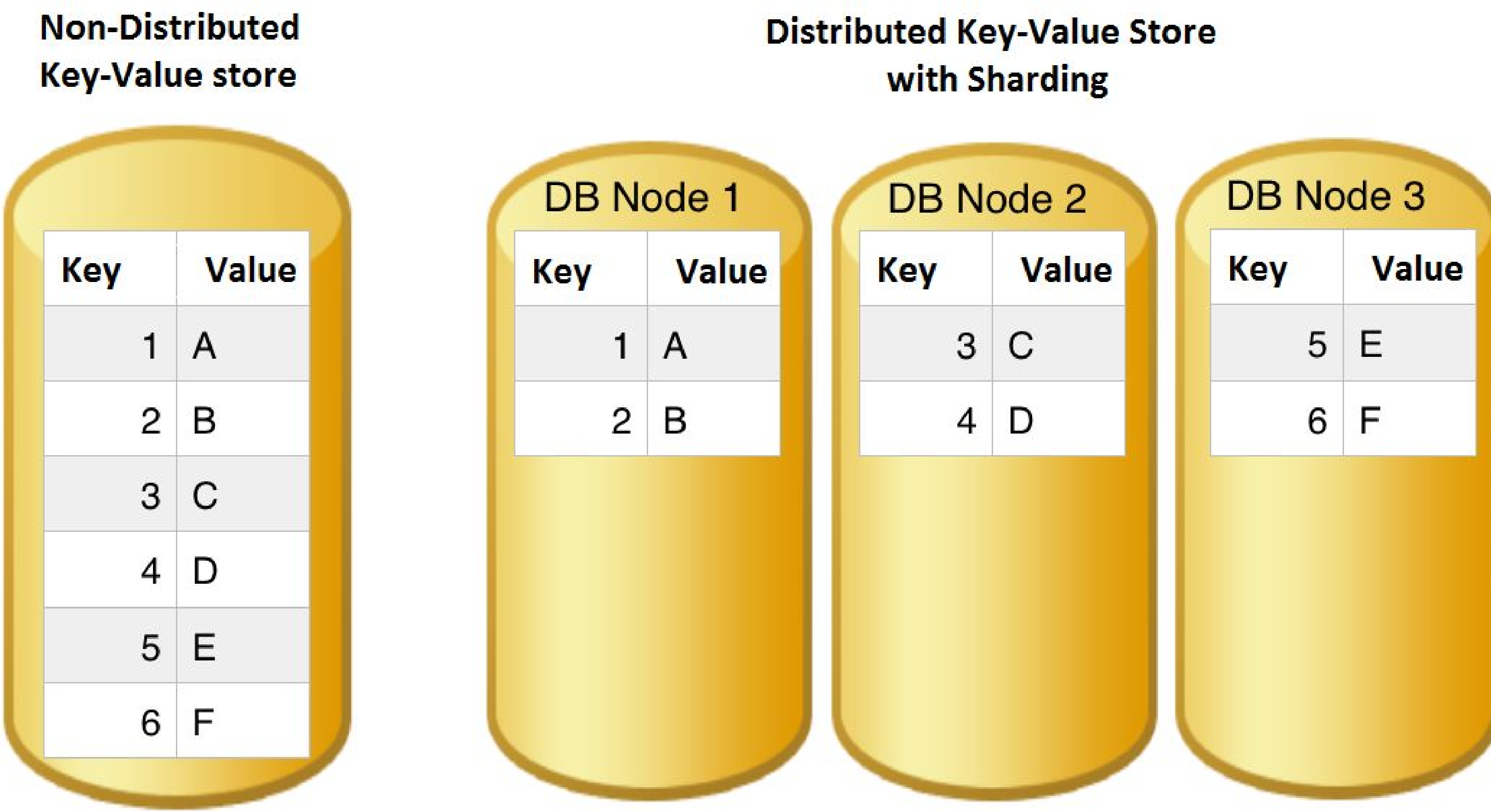
Even worse if the operations block each other!

P3.2: Motivation - Replication

- Read requests are very fast!
  - All clients have a replica close to them to access
- Write requests are quite slow
  - Instead of updating a single data center, all write requests must now update all 3 replicas
  - If write requests for a certain key is created, then they all have to wait for each other to finish

P3.2: Sharding (1)

- Partition data across multiple instances



P3.2: Sharding (2)

- Partition data across multiple instances
  - Needs a mechanism to partition the data
- Advantages
  - Updates are faster
- Only need to update one datastore
- Disadvantages
  - Fetching data is slower
- Overhead in identifying the location of data
  - Requires more storage capacity
  - More susceptible to failure

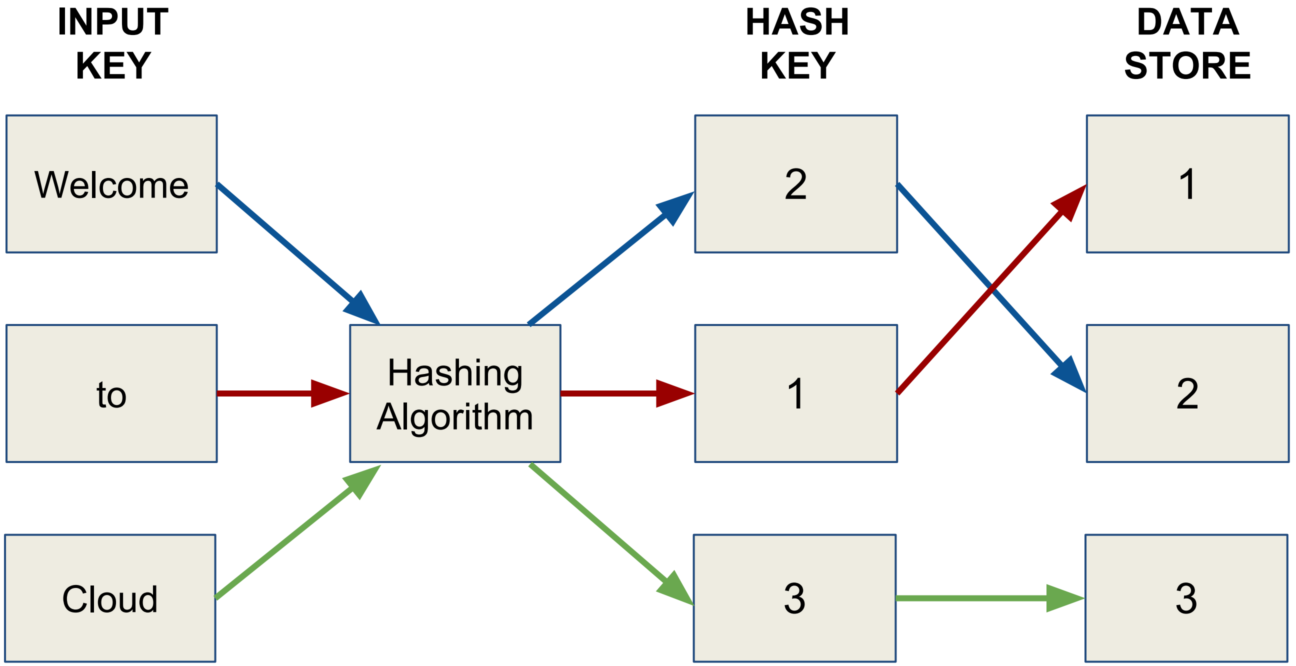
P3.2: Hashing

Hashing

- Map data (key) of variable length to a value of fixed length.

Consistent Hashing

- The hashing algorithm must return the same value for the same key at all times.
- Used to "remember" in which datastore a particular key is stored.

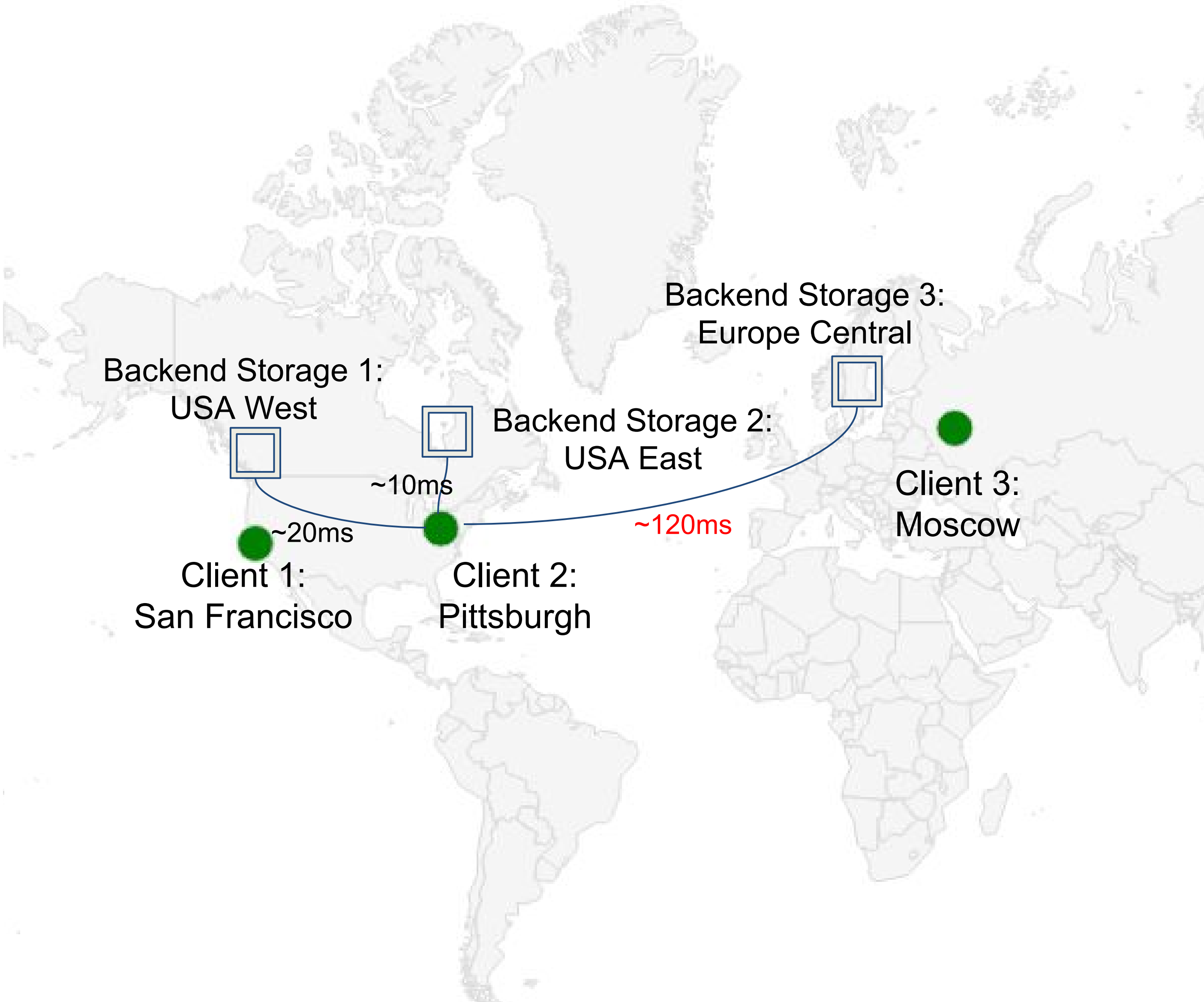


P3.2: Motivation - Sharding READ

Read Operation from DC2:

Min Latency: 10ms
Max Latency: 120ms
Average Latency: 50ms

P3.2: Motivation - Sharding WRITE



Write Operation:

Latency for Client 2:

Best Case- update DC2
= 10ms

Worst Case- update DC3
= 120ms

Expected Latency:
= 50ms

Can be very good if keys are properly distributed!

P3.2: Motivation - Sharding

- Read requests are rather slow
  - Once again, some clients may have high latencies due to being geographically further away from the appropriate data center
- Write requests are fast*
  - Even though there are 3 shards, only one ever needs to be updated!
  - Fast* performance derives from the ability to strategically distribute “hot” records to appropriate shards

P3.2: Your Task

- You are provided with 3 Key-Value datastore instances that support:
  - PUT & GET
- Implement a coordinator that supports:
  - Replication for PUT & GET
    - Strong consistency
  - Sharding for PUT & GET
    - Uses a consistent hashing algorithm
    - Bonus: even distribution

P3.2 Architecture

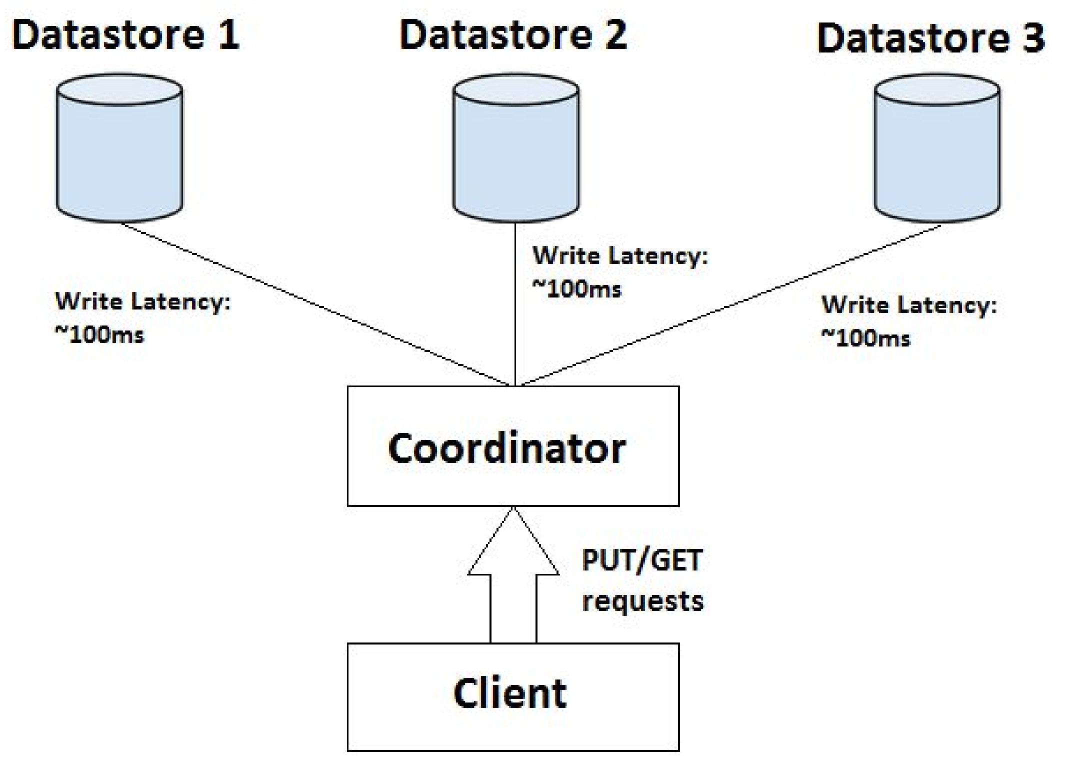
Coordinator receives PUT / GET requests from Client

Replication:

- PUT - All datastore must update
- GET - Retrieve from the datastore specified
- Strong Consistency

Sharding:

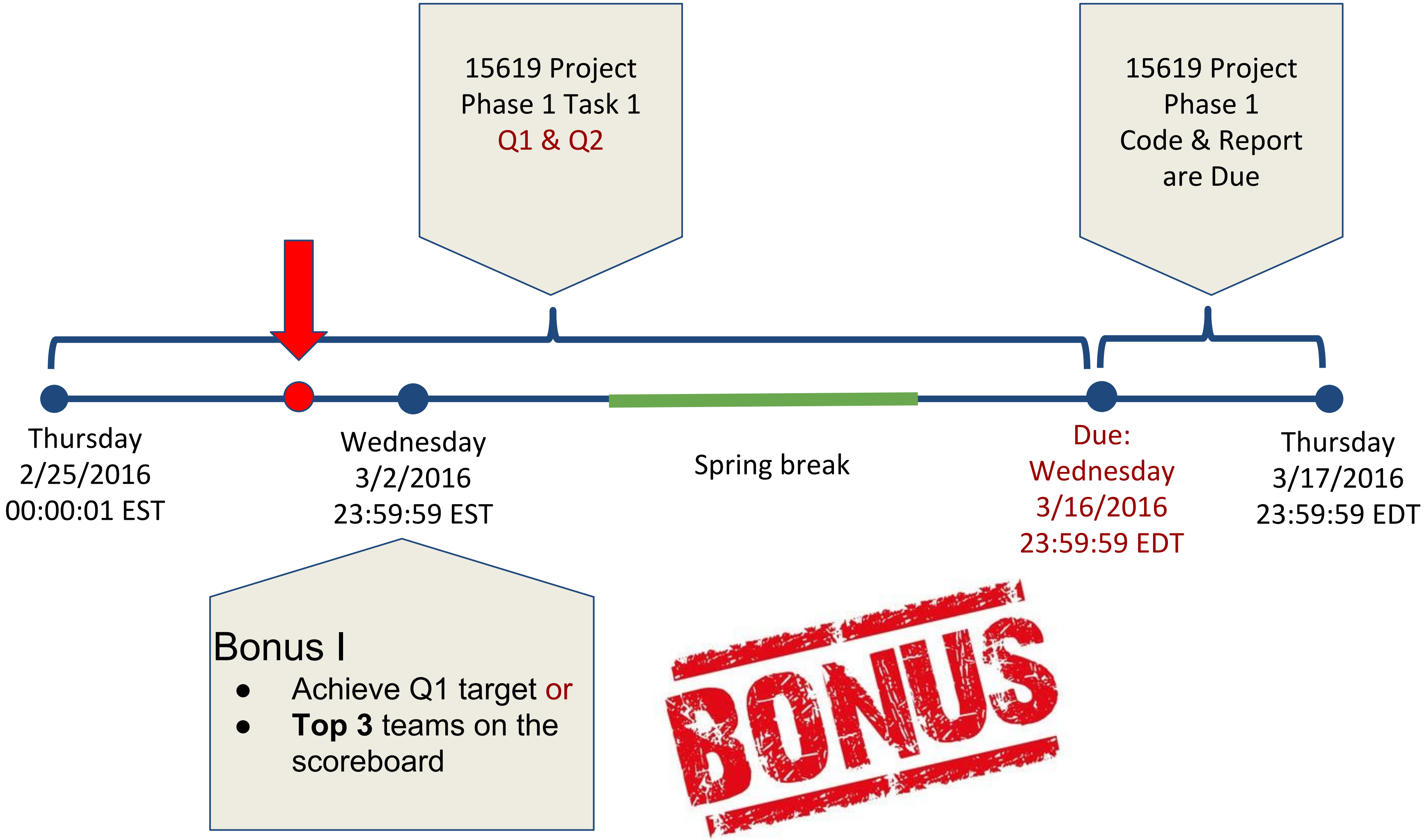
- PUT - Update a single datastore based on hash
- GET - Retrieve from the same datastore as PUT
- Bonus - Even distribution of data on all nodes



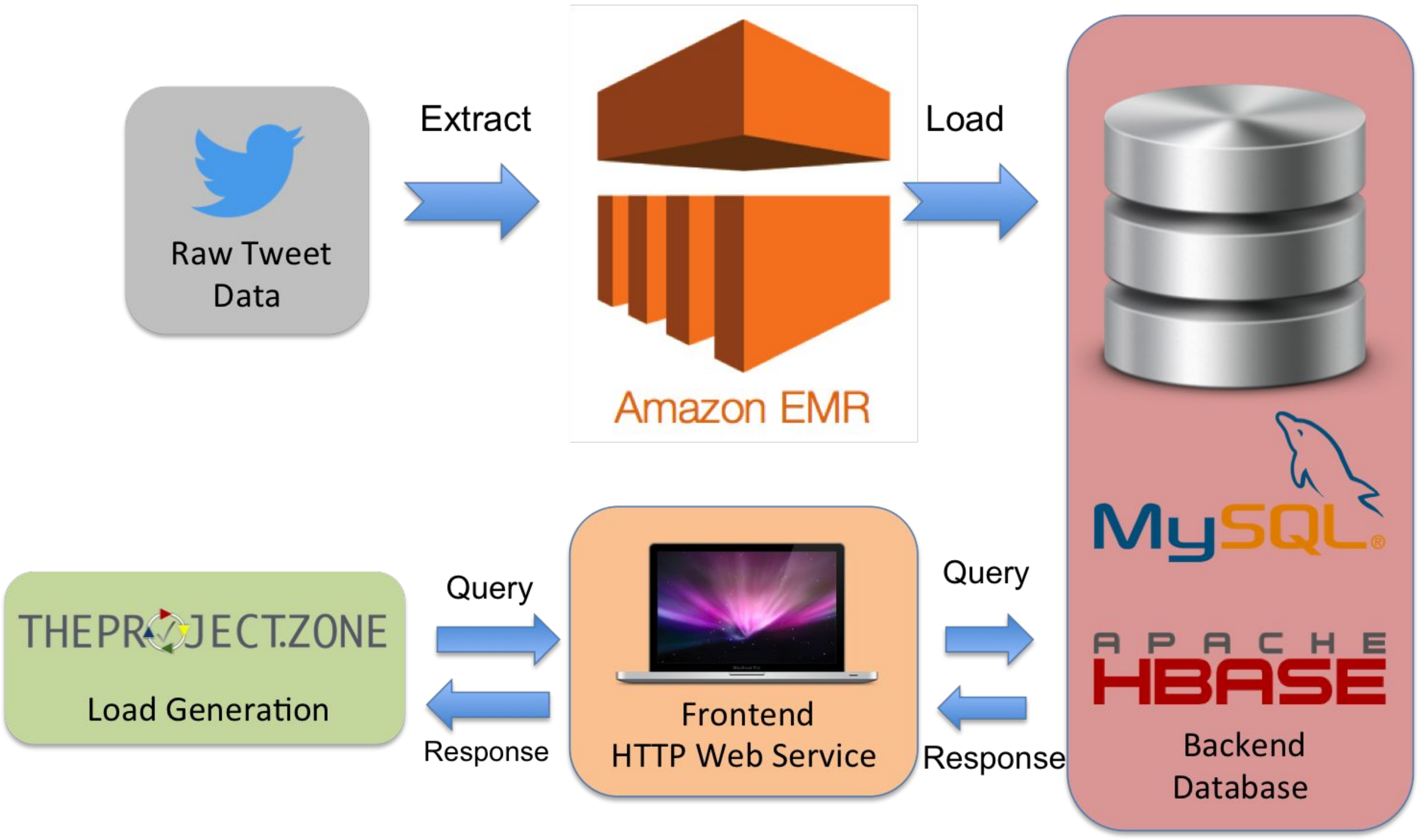
15619Project

TWITTER DATA ANALYTICS: 15619 PROJECT

15619 Project Phase 1 Deadlines



15619 Project System Architecture



Q1 : Heartbeat and Authentication

- Task
  - Greatest Common Divisor (GCD) for big integers
  - Decryption
  - Compare at least two frameworks (+ report)
- Tips
  - Understand how the frameworks work
  - Use multiple front-end instances

Current Submission Status

Rank	Nickname	Time	Total	Effective Throughput	Q1 Score	Q1 Effective Throughput
1	MyLittlePony	02/29/2016 00:43 -0500	100	33405.3	100	33405.3
2	QuackingDoge	02/28/2016 08:35 -0500	100	31556.3	100	31556.3
3	silverlining1	02/29/2016 13:45 -0500	100	31471.7	100	31471.7
4	ccsome	02/26/2016 19:17 -0500	100	30833.5	100	30833.5
5	ccfighter	02/28/2016 18:55 -0500	100	30775.2	100	30775.2
6	Steins;Gate	02/29/2016 02:32 -0500	100	30342	100	30342
7	YouKnowNothingJonSnow	02/28/2016 15:07 -0500	100	29344.4	100	29344.4
8	ThreeKings	02/29/2016 00:27 -0500	100	28998.1	100	28998.1
9	0X3D03	03/01/2016 02:38 -0500	100	27192.8	100	27192.8
10	AXIS	03/01/2016 04:17 -0500	100	25404.6	100	25404.6
11	Aplus	02/29/2016 13:30 -0500	100	25113.79	100	25113.79
12	Dimly Impugn	02/28/2016 17:19 -0500	88.99	22246.7	88.99	22246.7
13	Awesomething	03/01/2016 07:07 -0500	77.8	19449.8	77.8	19449.8
14	cc15619_fall16	03/01/2016 04:37 -0500	68.32	17079.1	68.32	17079.1
15	Shaw Eleven Lang	02/29/2016 23:50 -0500	62.9	15724.2	62.9	15724.2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Q2 : Text Cleaning And Analysis

- ETL Task
  - Sentiment Density Calculation
  - Text Censoring
  - Reference Data Set
- Request
  - Userid
  - Hashtag
- Return
  - Tweet ID, Tweet Time, Sentiment Density, Censored text

Q2: Sentiment Density

Amazingly, despite the nice cloudy weather, the BEST Hope for us to enjoy is to study CLOUD COMPUTING. Cloud is supper interesting.

Sentiment Density: ??

Word	**Score**		**Word**	**Score**	
amazing	4		interesting	3	
best	3		enjoy	1	
nice	2		super	7	
hope	2		study	-100	

Q2: Sentiment Density

Amazingly, despite ~~the~~ **nice** cloudy weather, ~~the~~ **BEST Hope** ~~for~~ us ~~to~~ **enjoy** ~~is~~ ~~to~~ **study** CLOUD COMPUTING. Cloud ~~is~~ supper **interesting**.

Effective Word: 15

Sentiment score: -89

SD: -89/15 = -5.933 (rounding)

Word	**Score**		**Word**	**Score**	
amazing	4		interesting	3	
best	3		enjoy	1	
nice	2		super	7	
hope	2		study	-100	

Q2: Text Censorship

Rot13ed Banned words
pybhq
vagrerfgvat

ROT13 →

Banned words
cloud
interesting

Amazingly, despite the nice, cloudy weather, the BEST Hope for us to enjoy is to study CLOUD COMPUTING. Cloud is supper-interesting.

↓

Amazingly, despite the nice,cloudy weather, the BEST Hope for us to enjoy is to study C***D COMPUTING. Cloud is supper-i*********g.

Q2: Other issues

- Unicode

  الحوسبة السحابية

  बादल कंप्यूटिंग

  云计算

  クラウドコ

  ンピューティング

  ಕ್ಲೌಡ್ ಕಂಪ್ಯೂಟಿಂ

  ווָאלקן קאַמפּיוטינ

  ג облачных вычислений

- Multiple tweets at the same time for a single user

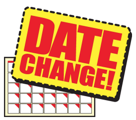
On Piazza

- Scoreboard Issues and Bonus
    - bonus top 3 by effective throughput of Q1 and Q2
    - score by effective throughput of Q1 and Q2
- Cost
    - EC2, EBS, ELB, EMR
- Spot instance limit / Accounts not verified
    - issue AWS support ticket / call

Start early!

Upcoming Deadlines

- Quiz 7: Unit 4 - Module 14 - Cloud Storage



Due: **<u>Thursday,</u>** 3/3/2016 11:59PM Pittsburgh

- Project 3.2: Partitioning (Sharding) and Replication

Due: 3/6/2016 11:59PM Pittsburgh

- 15619Project: Phase 1, Bonus (This week!)

Due: 3/2/2016 11:59PM Pittsburgh

- 15619Project: Phase 1

Due: 3/16/2016 11:59PM Pittsburgh