15440/15640 Distributed Systems

Homework 3

Q1. RAID (15 Points)

As a TA for 15-440, Eunice wants to build a DFS to store all relevant files for the course. She bought 20 used hard drives, each with 100GB capacity and an MTTF of 4 years. The read and write throughput for each disk is 100MB/s. She wants to apply the fault tolerance concepts she learned in class and build a RAID array with her disks.

Q1.1 (5 Points)

Eunice starts out by building a RAID-0 array with the disks she bought.

A. What is the effective capacity and MTTF of the RAID-0 array she built? (2 point)
   Capacity of RAID-0 array: # disks * capacity per disk
   = 20 * 100 GB = 2000 GB
   MTTF: first out of 10 disks to fail = MTTF/N = 4/20 = .2 years

B. What is the read throughput and write throughput of this RAID-0 array? (1 Point)
   Read/Write throughput = # disks * read/write throughput per disk
   = 20 * 100 MB/s = 2000 MB/s

C. Eunice notices that her disks are unreliable, so every time one disk fails, part of her entire filesystem is lost. What can she change about her system to reduce the loss of data? Explain why your solution helps. (2 Points)
   Eunice can upgrade her system to use RAID-1 instead of RAID-0, since it provides reliability (can tolerate disk failures, since there are now duplicates of the data on different disks.

   Alternative Answer: Eunice can do periodic backups of her data. That way, if a disk fails, she only loses the data written after the last backup, and not all the data she has written so far.

Q1.2 (5 Points)

Eunice is talking to Henry about the reliability issues with her RAID-0 array, and he recommends that she build a RAID-1 array instead. She decides to use her disks to build a RAID-1 array instead of a RAID-0 array.

A. What is the new effective capacity and MTTF of this RAID-1 array? (2 points)

Capacity of RAID-1 array: # disks / 2 * capacity per disk
= 20 / 2 * 100 GB = 1000 GB

MTTF: Raid 1 fails if any pair of disks fails (10 pairs total)
MTTF for a pair to fail: (MTTF for the first disk to fail + MTTF for second disk)
= MTTF / 2 + MTTF = 4 / 2 + 4 = 6
MTTF for system to fail:
MTTDL per pair / # pairs = 6 / 10 = .6 years

B. What is the read throughput and write throughput of this RAID-1 array? (1 point)
Read throughput: # disks * read throughput per disk
= 20 * 100 MB/s = 2000 MB/s
Write throughput: # disks / 2 * write throughput per disk
= 20 / 2 * 100 MB/s = 1000 MB/s

C. Eunice notices that two of her disks have potential hardware issues. Can her current setup tolerate these two disk failures? Why or why not? (2 points)
Depends: if the two disks that fail store the same data, then her system will fail if these disks fail. On the other hand, if two disks that store different data fail, then the system can tolerate this loss, since it has a replica of the data sitting on functioning disks.

Q1.3 (5 Points)

Eunice has decided that in addition to storing all the course files, she also wants to store every student submission. Her storage overhead has increased dramatically and is now quite expensive, since she needs to store a lot of files but has limited capacity.

A. Is RAID-1 still the best choice for her DFS? Why or why not? (2 points)
RAID-1 is no longer the best choice for her DFS because it has the least capacity out of any of the RAID levels. Since storage is expensive, Eunice will want to use a RAID system that maximizes her capacity, so that she has to buy less disks.

B. Eunice wants to use a new scheme (i.e. different RAID level) to decrease storage overhead, while still maintaining a reliability of at least 1. What scheme do you recommend, and what would be the new read and write throughput of this new system? (3 points)
Multiple different answers - can use RAID 4 or RAID 5
RAID 4:
Read throughput: (# disks - 1) * read throughput
= (20 - 1) * 100 MB/s = 1900 MB/s
Write throughput: (read throughput) / 2 = (100 MB/s) / 2 = 50 MB/s
RAID 5:
Read throughput: # disks * read throughput = 20 * 100 MB/s = 2000 MB/s
Write throughput: (# disks / 4) * read throughput = 20 / 4 * 100 MB/s = 500 MB/s

Q2. Mock Interview on Distributed ML (18 Points)

You are about to interview with a company that specializes in distributed ML. Crystal offers to do a mock interview with you so that you can review all concepts. Can you answer all her questions?

Q2.1 (6 Points)

As a warm-up activity, tell Crystal which framework you would prefer for the following scenarios (Spark or MapReduce) with a one-sentence explanation. (3 points each, 1 for the framework, 2 for the explanation)

A. Cluster has a high rate of node failures.
   MapReduce. It persists intermediate data so it is easier to recover without having to perform a lot of re-computation.

B. Perform iterative data analytics.
   Spark. Spark allows for programs that avoid persisting intermediate data.

Q2.2 (6 Points)

MapReduce Challenges:

A. If the MapReduce framework randomly assigns map tasks to nodes, how would such random assignment impact the execution of a MapReduce job on a cluster? (2 points)
   [Acceptable answer 1:] Random assignment will not take advantage of **locality** and hence spend more time fetching data for map tasks, which can hinder performance.
   [Acceptable answer 2:] Random assignment to workers can result in uneven distribution of tasks, with some workers getting assigned more work than others, hindering performance.

B. MapReduce has a simple failure model. Briefly explain how MapReduce addresses node failure on a map task execution and explain why such failure handling is safe. (4 points)
   [Explanation we are looking for regarding what happens:]Upon node failure, any task can simply be rescheduled. (Lower level detail which student does not need to know: MapReduce will replicate the failed tasks at healthy nodes within the vicinity of the nodes that hold the data replicas on HDFS)
   [Explanation for why it is safe:] Individual Map and Reduce tasks are idempotent and have no side effects. Since no jobs depend on others, simply reschedule and re-run the job if safe.

Q2.3 (6 Points)

Spark Challenges:

A. One big term that people use when talking about Spark is “lineage”. What is lineage information used for? (2 points)
When an action is invoked, Spark will compute the needed RDDs according to their lineage.
(If the student 's answer talks about using lineage to recover the lost RDD from crash, the explanation should also get full points.)

B. “In-memory” is another keyword that people know about Spark. Can you tell Crystal why Spark is considered "in-memory"? How is this different from what MapReduce uses? (4 points, 2 for each framework)
Spark allows the application program to cache frequently using RDDs in application memory so future computation may reuse data from application memory.
MapReduce jobs always start by reading data from the file system and end by writing output to the file system.

Q3. World’s Greatest Datacenter (26 Points)

Nirav is building a datacenter, and wants to equip it with the latest and greatest in distributed storage technology. He hears about Google File System (GFS), the cluster file system design pioneered by Google. He also knows about Andrew FS (AFS) from his time at CMU. Despite forgetting his Kubernetes password every week, he remembers an interesting detail about AFS: it uses whole file caching (with callbacks) so reads/writes can be buffered locally and then flushed on close.

Q3.1 (8 Points)

For each of the following cases, help Nirav evaluate whether GFS or AFS would be a better fit for his datacenter. For the purpose of this question, assume that AFS does not perform replication, and that the server failure rate is 0% unless specified otherwise. Give a **one-line** explanation for your answer (2 points each).

A. Multiple (several hundred) clients concurrently appending entries to the same file, with the expectation that all writes should persist.
GFS. Concurrency is built in for GFS.

B. Limited server storage.
AFS. Because data is duplicated in GFS due to replication.

C. Tolerating high rate of failure of the file server (*i.e.* chunkservers for GFS).
GFS. Because it supports failure recovery using replicas, WAL, and checkpointing.

D. Repeated read accesses to the same file.
AFS. Because AFS utilizes caching.

Q3.2 (11 Points)

After carefully considering the alternatives, Nirav decides to implement his datacenter's file storage atop GFS. Recall that, in GFS, the master server is the central repository of chunk metadata, while the chunkservers store chunks of files. Help Nirav configure his GFS implementation so he can keep up with his fiercest competitor, nozamA!

A. By default, GFS implements *migration*, or periodic relocation of data chunks from a live chunkserver to another. A clever engineer suggests that disabling migration might help improve performance. Give one reason in support of the engineer's argument, and one reason against. [4 points]
For: Reduces network bandwidth utilization and saves compute cycles on the servers (master and chunkservers) by forgoing background work.

Against: Migration helps load-balance both disk and network bandwidth demand. (Also OK: this kind of wear levelling may improve the lifespan of flash-based storage media.)

B. Nirav observes that the master server's RAM utilization is consistently at 100%, and metadata for a large number of chunks is being swapped to disk, hurting performance. How might Nirav fix this bottleneck without adding any new hardware? [3 points]
The master server stores chunk metadata in RAM. Since the overhead is per chunk, high RAM (and swap) usage indicates more chunks than the GFS deployment can handle. The bottleneck can be fixed by increasing the chunk size (from 64MB to, say, 256MB).

C. Nirav observes a peculiar trend of correlated failures: for some unknown reason, the likelihood of a Top-of-Rack (ToR) switch failing is several orders of magnitude higher than any other type of failure. Without increasing his operational costs or storage capacity requirements, what change should Nirav make to the default GFS configuration in order to improve chunk availability? Give one downside of this change. [4 points].
By default, GFS places two chunk replicas on one chunkserver, and the third replica on a different chunkserver. If ToR failure is overwhelmingly the likely failure mode (i.e. an entire rack becomes inaccessible), two of the three chunk replicas will go down simultaneously, increasing the likelihood of a chunk becoming totally unavailable.

Fix: Change replication policy so the three replicas are stored on three different racks
Downside: Write traffic has to flow through one additional rack, which might hurt performance.

Q3.3 (7 Points)

Having crowdsourced design decisions from the brilliant students in 15-440/640, Nirav finally deploys his GFS-based datacenter. His deployment consists of a single master server and three chunkservers.

A. Nirav is drafting a document describing the guarantees provided by his datacenter's storage system. State whether the following statements he has written about GFS are true or false, and provide a *one-sentence* justification of your answer. For the moment, assume no failures. [4 points].

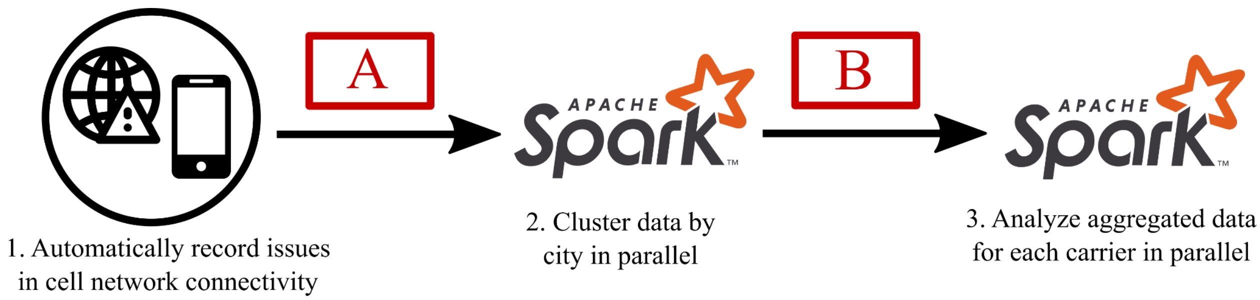
  i. *"During a WRITE operation, chunk data is transferred along a daisy-chain comprising the nearest secondary replica, the master replica, and finally the furthest secondary replica, in that order."*
  False. The chunkservers in the daisy-chain can have arbitrary order.

  ii. *"If two users, Alice and Bob, concurrently perform APPEND to the same chunk, it is possible that Alice's data gets overwritten by Bob's."*
  False. Append cannot overwrite data due to serialization through the primary chunkserver.

B. Now imagine one of the chunkservers in the deployment fails. **Name** the mechanism used to detect this failure, and state the next steps performed by the master once it has detected the failure. [3 points].
Heartbeats. Decrement count of replica for all chunks on dead chunkserver. Re-replicate chunks missing replicas in background.

Q4. WeLoveNetworkCarriersSoMuch (6 Points)

A new startup, WeLoveNetworkCarriersSoMuch, is pitching an analytics dashboard to allow network carriers to quickly identify and triage city-wide connectivity issues in real time. They achieve this using a smartphone application that records cell phone connectivity events (*e.g.* dropped calls). Their application records and periodically sends data about its location, carrier, and any recent connectivity issue to WLNCSM's central servers. They have data-centers that they want to dedicate to two different tasks: clustering location data by city, and analyzing the data for each carrier. The system pipeline is depicted in the figure below.



Q4.1 (2 Points)

WeLoveNetworkCarriersSoMuch considers using RPCs to communicate between the application and its computing servers. Is this a good idea? Give a one-sentence justification for your answer. [2 pts].

RPC's are bad because the system will have large numbers of failures and many nodes working in parallel. Handling Replication with RPCs is hard.

Q4.2 (2 Points)

What framework(s) would we apply in this situation instead? Name the frameworks that you would use at A and B, and give a one-sentence justification for your answer. [2 pts].

A and B should both be using Kafka (Pub-Sub). Any explained advantages of Kafka over RPC: High-throughput, Low Latency, Fault-Tolerant, Durability.

Q4.3 (2 Points)

A and B will probably require some sort of data grouping. In one sentence each, describe the best way to group data for both A and B. Please frame your answer in the context of the framework being used. [2 pts].

Since we are using PubSub we wanted to see some sort of description of topics. We accept a topic-based or content-based approach as long as it is explained well. For example:

A: Content-based, with content being the phone's location

B: Topic-based, with topic being the network carrier

Q5. Let's buy her a gift! (16 Points)

For this question, please upload a PDF file as your final answer. You are responsible for uploading the correct document to this question. The course staff will not help upload your solution after the hw deadline.

Eunice's birthday is approaching, and Emma wants to buy her a gift. She decides to look on gifts.emmazon.com, her own proprietary online gift store, for a suitable gift. To direct her to the website, her local DNS server performs an iterative lookup. The diagram below shows some of the DNS records contained in each DNS server. Note that DNS responses are cached in the local DNS server.

localdns.cmu.edu (S1)

Record Number	Name	Value	Type	TTL
R1	c.root.net	190.40.4.8	A	24 hours
R2	.	c.root.net	NS	24 hours

c.root.net (S2)

Record Number	Name	Value	Type	TTL
R3	b.gltd.net	190.31.1.9	A	12 hours
R4	.	b.gltd.net	NS	12 hours

b.gltd.net (S3)

Record Number	Name	Value	Type	TTL
R5	emmazon.com	ns-9.emmazon.com	NS	4 hours
R6	ns-9.emmazon.com	83.102.188.3	A	4 hours

ns-9.emmazon.com (S4)

Record Number	Name	Value	Type	TTL
R7	gifts.emmazon.com	83.102.188.4	A	30 minutes
R8	gyfts.emmazon.com	83.102.188.5	A	30 minutes
R9	gifts.ammazon.com	83.102.188.6	A	30 minutes
R10	gefts.amazon.com	83.102.188.7	A	30 minutes

Q5.1 (4 Points)

Fill in the following table to indicate the sequence of queries and responses exchanged among the servers.

	Sender	Receiver	Type (Query/Response)	Data
1	Emma's PC	S1	Query	gifts.emmazon.com
2	S1	S2	Query	gifts.emmazon.com
3	S2	S1	Response	R3, R4
4	S1	S3	Query	gifts.emmazon.com
5	S3	S1	Response	R5, R6
6	S1	S4	Query	gifts.emmazon.com
7	S4	S1	Response	R7
8	S1	Emma's PC	Response	R7

Q5.2 (4 Points)

Fill in any new DNS records in the local DNS server right after the sequence of queries and responses in (a). Label the new records with record numbers starting at R11.

Record Number	Name	Value	Type	TTL
R1	c.root.net	190.40.4.8	A	24 hours
R2	.	c.root.net	NS	24 hours
R11	b.gltd.net	190.31.1.9	A	12 hours
R12	.	b.gltd.net	NS	12 hours
R13	emmazon.com	ns-9.emmazon.com	NS	4 hours
R14	ns-9.emmazon.com	83.102.188.3	A	4 hours
R15	gifts.emmazon.com	83.102.188.4	A	30 minutes

Q5.3 (4 Points)

Emma takes a break from browsing to hold office hours. After 4 hours, Emma is done, and looks at gifts.emmazon.com again, to see if there are any updates of super cool gifts on the site. Fill in the DNS records in the local DNS server right before any queries and responses are performed for her second request.

Record Number	Name	Value	Type	TTL
R1	c.root.net	190.40.4.8	A	20 hours
R2	.	c.root.net	NS	20 hours
R11	b.gltd.net	190.31.1.9	A	8 hours
R12	.	b.gltd.net	NS	8 hours

Q5.4 (4 Points)

Once again, fill in the following table to indicate the sequence of queries and responses exchanged among the servers for Emma's second request.

	Sender	Receiver	Type (Query/Response)	Data
1	Emma's PC	S1	Query	gifts.emmazon.com
2	S1	S3	Query	gifts.emmazon.com
3	S3	S1	Response	R5, R6
4	S1	S4	Query	gifts.emmazon.com
5	S4	S1	Response	R7
6	S1	Emma's PC	Response	R7