

Homework 6

Md Ali

CS 550: Advanced Operating Systems

November 15, 2020

Exercise 1. What is application-aware I/O? What is its relationship with ASIC and In-Situ I/O optimization? Why do we emphasize application-aware I/O optimization?

Proof. Application-aware I/O is an intelligent network to maintain current information about applications that connect to it and optimize their functioning as well as that of other applications or system that they control. For the optimization in ASIC and In-Situ I/O optimization the data access is application dependent, it dynamic for parallelism, there is a data access pattern that allows feed back control. This also includes the integration language, memory system, and hardware/software infrastructures. We emphasize this is because application-aware I/O optimization paves the way for parallel distribution. \square

Exercise 2. Provide an overview of PFS. Briefly discuss its design goals, its strengths, and weaknesses.

Proof. Parallel File Systems offer many use cases, particularly in computational sciences. This gives us computer simulation as a tool for greater understanding of the real world. These filing systems are designed for parallelism and high performance. This also provides global share namespace for clients. The overall design goal is to provide a directory tree such that all nodes can see. Some of the strengths of a parallel file system is that you can map data across many servers and drive while coordinating access to data so certain access rules are still followed. There are weakness such that there is still the issue of find usable, efficient, portable interfaces while understanding and tuning I/O. \square

Exercise 3. Compare the file-per-process and shared-file access patterns. Pros and cons.

Proof. A file-per-process essentially lets each application task create a separate file, and writes only to that file while a share-file access creates a single file with all application tasks write to that file. A pro of having a file=per=process is that it avoid the lock contention on file systems while a con would be that it is impossible to restart an application with different number of tasks. A pro with shared-file is that this increases usability but a big con is that this can create lock contention which in turn will hinder performance. \square

Exercise 4. Describe how data distribution works in PFS.

Proof. In a parallel file system, data distribution works similar to that of other distributed files systems where there is support for a shared global namespace, except instead of having things so structured, data distribution in a parallel file system. the client system have direct access to all of the storage nodes for data transfer without having to go through single coordinating server. \square

Exercise 5. Locking in PFS. Discuss and list some of the challenges.

Proof. In parallel file systems, files are broken up into lock units. These locks are obtained by clients that will access them before I/O occurs. Some of the challenges is implementation burden because it add complexity to a file system also locks are expensive. \square

Exercise 6. Why was Hadoop platform for Big Data processing introduced and became so popular? What features made it such a widely adopted solution for data processing? What are its benefits over existing solutions? Explain briefly.

Proof. The reasoning for Hadoop platform for Big Data processing introduced was that there was a need to process huge datasets on large clusters of computers. The main popular selling popular is that it would distribute the data as it is initially stored. The application within Hadoop are writing in high-level programming languages and nodes communicate as little as possible. Data is spread among the machines in advance. The benefits is that Hadoop can handle failure and cuts cost for reliability in each application. \square

Exercise 7. What is MapReduce and what are its characteristics? List its advantages and disadvantages. Compare it with MPI-based approaches to parallelize a certain task.

Proof. Mapreduce is a method for distributing a task across multiple nodes. The advantage are that it provides automatic distribution, fault-tolerance, monitoring tools, and usually written in Java. The advantages of mapreduce are scalability, cost-effective, flexibility, fast, and secure. Some disadvantages to mapreduce is that it may not be always easy to implement and is not suitable for real-time processing. \square

Exercise 8. Provide pseudocode for sorting integers in an out-of-core fashion (i.e., integers cannot fit in memory) both in MapReduce (both mapper and reducer code) and in MPI. Think about the flow of sorting: reading the input, perform the sorting algorithm, merge all intermediate results, write the final sorted output.

Proof. Map: $(K1, V1) \rightarrow (K2, V2)$

Reduce: $(K2, \text{list}(V2)) \rightarrow \text{list}(K3, V3)$

MPI Map: $(\text{GroupID}, \text{ProcessID}) \rightarrow (\text{GroupID}, \text{ProcessID})$

MPI Reduce: $\text{MPIReduce}(\text{void}^* \text{senddata}, \text{void}^* \text{recvdata}, \text{int count}, \text{MPIDatatype datatype}, \text{MPIOp op}, \text{int root}, \text{MPI Comm communicator})$

\square

Exercise 9. Provide a table of features for Distributed File Systems (HDFS) and Parallel File Systems (PVFS). Briefly discuss similarities and differences.

Proof. Below is a table of features for Distributed File Systems and Parallel File Systems.

| HDFS | PVFS |
|------------------------------------|-------------------------------------|
| Write Once, Read Many Times | Explicit State Machine System |
| Namespace in RAM | Stateless Servers and Clients |
| Persistent Checkpoint | Stateless System without locks |
| Critical Files Replication Factor | Clusterwide consistent name space |
| Hierarchy of Files and Directories | Deleted File is Removed Immediately |
| File Lease | Buffered Messaging Interface |
| File Content Replication | Two step permission checking |

A parallel file system is a type of distribution file system. So there are some similarities between the two, hence both of them can spread data across multiple storage servers, scale to accommodate lots of data, and both also support high bandwidth.

A distributed file system used a standard network file access protocol while a parallel file system requires an installation of client based software driver to access the share storage. Another distinction is that a distributed file system essentially stores a file on a single storage node while a parallel file system will break up the file and take away the data blocks to multiple storage nodes. A last distinction to note, is that distributed file systems tend to target data-heavy application while parallel file system focus in high performance workloads that that do coordinated I/O access and use significant bandwidth.

□

Exercise 10. How does data distribution works on HDFS? Who is responsible for distributing data? What would you optimize in the distribution policies to make the system faster and more reliable?

Proof. The architecture for HDFS operates on top of an existing filesystem and files are stored as blocks. There is also no data caching due to the large data sets that it will handle. For the data distribution, we have NameNode's that will manage file system namespace and information about DataNodes. Now the main issue here is that if the NameNode some how goes down and gets removed, then there must be a place where it can be replicated or some sort of fault tolerance in place for such issues. I would prioritize some sort of policy for have replication in the system as a priority to avoid any issues.

□

Exercise 11. Discuss how the fault tolerance features in HDFS work? Compare it with hardware based (RAID) approaches. What other approaches for fault tolerance are out there (think erasure coding)?

Proof. In a HDFS, the file system replicates each piece of data multiple times and distributes the copies to individual nodes, placing at least one copy on a different server rack than the others. There are also DataNodes that send heartbeats to the NameNode and the NameNode uses heartbeats to detect DataNode failure. This insure that nothing will be lost and can be

recovered. Instead, RAID will copy the data known as disk mirroring and this is different because you can't chunk the data instead it's just somewhere else. Other fault tolerance methods include recovery blocks, n-version programming, and self-checking software. \square

Exercise 12. Provide a description for the ideal workload for HDFS. What about the worst-case scenario?

Proof. The ideal work load for HDFS is ideal for application with large data sets and needs to be providing streaming access to file system data. The worst case scenario would be if the NameNode keep failing all the time. \square

Exercise 13. Could you describe the basic idea and method behind the I/O coordination algorithm we introduced in Lecture 21 and 22.

Proof. I believe from the lectures, the methods that were described were for accessing files in certain file systems where we have sequential, random, and keyed access. The coordination will depend on the file system we are trying to implement. \square

Exercise 14. Would it be safe to join message 3 and 4 in the authentication protocol shown in Fig. 9.6, into $KA, B(RB, RA)$?

Proof. It would be safe to join message 3 and 4 because there are no issues with Alice's message to Bob that can't be in the same message. \square

Exercise 15. Why it is not necessary in Fig. 9-9 for the KDC to know for sure it was talking to Alice when it receives a request for a secret key that Alice can share with Bob?

Proof. Since Alice is the only other person holding a secret key $K_{A,KDC}$ and if any other person tries to send a message they won't be able to decrypt it unless they are Alice, who has the secret key to decrypt it. \square

Exercise 16. Devise a simple authentication protocol using signatures in a public-key cryptosystem.

Proof. Taking Alice and Bob communicating with each other and Alice wants to make sure she is sending messages to Bob, she can send Bob a challenge R . From there Bob will be requested to place his signature under the challenge. Once the signature has been received by Alice, she can then check if it really is Bob and decrypt the response and send it back over to Bob. \square