Tim Shaker
CST 334 Online
April 21, 2026

Working on this project gave me a better understanding of how large-scale systems are designed under real-world constraints, which can be very different from what is typically assumed in a local-computing environment. In a distributed system like GFS, failures are expected rather than exceptional, and consistency is more approximate than precise. Despite the challenges inherent in distributed systems, the authors demonstrated that GFS effectively solved the problems they were facing, serving as a model for systems that followed, like its successor Colossus, and other distributed systems like Hadoop.

One of the most interesting aspects of GFS is how the authors made intentional choices about trade-offs. Instead of trying to build a general-purpose file system that satisfies all possible use cases, the authors focused on their specific needs, optimizing for large-scale data processing with huge files, sequential reads, and append-heavy workloads. The relaxed consistency model is a clear example of this; rather than guaranteeing that every replica is identical, GFS shifted data validation to consumer applications, allowing the overall system to improve throughput and scalability, which was more important for their use case than strict consistency across chunkservers.

Another aspect worth noting was the simplicity of the authors' core system design. Although the actual implementation involved significant complexity, their single-master design with multiple chunkservers is intuitive and easy to reason about. As the authors mentioned, simplicity was a central concern while designing GFS, and even though the single master seems like it could be a bottleneck, other design choices they made prevented it from becoming an issue in practice. This is a useful insight for system design: start with the simplest solution possible and then address any issues that arise.

Finally, reading about the evaluation methods used in the paper helped me to understand how large systems are validated in practice. The authors used both micro-benchmarks from their lab testing and real-world cluster data to demonstrate scalability and reliability. This approach showed me how both lab benchmarks and real-world performance data are essential when evaluating distributed systems. Overall, this paper provides many useful insights on distributed system design, emphasizing that real-world engineering is about understanding constraints and making trade-offs to optimize for the specific things that matter for the problem being solved, rather than designing a system that is perfect in every aspect. It also showed me how theoretical concepts from class, like designing for concurrent access, are applied in practice at a massive scale.