End-to-End Parallelism for Distributed Science Workflows

Driven by the advancements in instrument and computing technologies, an increasing number of science applications started to generate large volumes of data reaching to petabytes in scale. As an example, soon-to-be-operational cosmology project Large Sky Survey Telescope will use a 3,200 megapixel camera to take pictures of the universe for 10 years and is expected to produce 15 TB of raw data every night, totaling 200 PB in its operational lifetime [2]. This fast-growing data generation rate when combined with the distributed and collaborative nature of many scientific projects has led wide-area network traffic to increase at an unprecedented rate. Building high-speed networks with 100/400 Gbps bandwidth is necessary but not sufficient as existing data transfer tools and services fall short to provide reliable high performance in these networks.

Most distributed science workflows rely on highly-skilled network operators or third-party transfer services (e.g., Globus) to tune transfer settings, typically once at the beginning of the transfer. However, optimal transfer configuration vary over time due to changing system conditions such as network and storage interference [3]. Current practice to overcome *performance variability for delay-sensitive distributed science applications* is to use over-provisioned dedicated network and computing infrastructures, but this is not a sustainable solution in the long term due to its prohibitive cost. Yet, even dedicated resources are bound to performance variations due to system noise and intra-job/inter-job resource interference [4, 8, 9]. As a result, there is a need for adaptive parallelism for distributed science workflows to provide robust, reliable end-to-end performance in the presence of performance instability.

We propose adaptive end-to-end parallelism for distributed science projects to dynamically adjust network parallelism to sustain high transfer throughput while ensuring data integrity. To achieve this goal, we will integrate adaptive network transfer optimization algorithm HARP [3] into elastic workflow schedulers. This integrated framework will periodically monitor the performance of computing and transfer tasks to identify and mitigate bottlenecks by dynamically adjusting parallelism through multithreading and multi-node execution for computing tasks and concurrent file downloads/uploads for transfer tasks. We will further innovate computation-aware transfer parallelism to control transfer speed to match computation speed to obviate the need for intermediate storage space and to increase resource utilization. HARP will accept *throughput requests* from the workflow schedulers and reconfigure transfer settings to adjust its speed to match that of computation. For example, if computation speed is slower than transfer speed, workflow scheduler may request HARP to throttle its performance to minimize the data backlog. HARP can then reconfigure transfer parameters to meet this demand.

Besides transfer throughput, integrity of data transfers is also crucial for applications that are sensitive to data manipulation, such as the Dark Energy Survey [1] and Sky Survey Simulation [7] projects. Hence, application layer end-to-end integrity verification is proposed to detect and recover from silent data corruption by using secure cryptographic hash function to check the file content at source and destination end points [5, 6]. However, its single-threaded implementation by existing transfer applications (i.e., Globus) incurs significant performance penalties at high speeds thus hampers its adoption in next-generation terabit-per-second networks. Therefore, we will introduce scalable integrity verification by taking advantage of multi-core architectures of data transfer nodes while to parallelize compute-intensive checksum compute operations and match the transfer and compute speeds.

References

[1] Dark Energy Survey, 2017. https://www.darkenergysurvey.org/.

[2] Large Synoptic Survey Telescope, 2017. https://www.lsst.org/.

[3] E. Arslan and T. Kosar. High-speed transfer optimization based on historical analysis and real-time tuning. *IEEE Transactions on Parallel and Distributed Systems*, 29(6):1303–1316, 2018.

[4] Z. Cao, V. Tarasov, H. P. Raman, D. Hildebrand, and E. Zadok. On the performance variation in modern storage stacks. In *15th {USENIX} Conference on File and Storage Technologies ({FAST} 17)*, pages 329–344, 2017.

[5] B. Charyyev, A. Alhussen, H. Sapkota, E. Pouyoul, M. H. Gunes, and E. Arslan. Towards securing data transfers against silent data corruption. In *IEEE/ACM International Symposium in Cluster, Cloud, and Grid Computing, IEEE/ACM*, 2019.

[6] B. Charyyev and E. Arslan. RIVA: Robust integrity verification algorithm for high-speed file transfers. *IEEE Transactions on Parallel and Distributed Systems*, 31(6):1387–1399, 2020.

[7] S. Habib, A. Pope, H. Finkel, N. Frontiere, K. Heitmann, D. Daniel, P. Fasel, V. Morozov, G. Zagaris, T. Peterka, et al. HACC: Simulating sky surveys on state-of-the-art supercomputing architectures. *New Astronomy*, 42:49–65, 2016.

[8] S. Herbein, D. H. Ahn, D. Lipari, T. R. Scogland, M. Stearman, M. Grondona, J. Garlick, B. Springmeyer, and M. Taufer. Scalable i/o-aware job scheduling for burst buffer enabled hpc clusters. In *Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing*, pages 69–80. ACM, 2016.

[9] S. Karki, B. Nguyen, J. Feener, K. Davis, and X. Zhang. Enforcing end-to-end i/o policies for scientific workflows using software-defined storage resource enclaves. *IEEE Transactions on Multi-Scale Computing Systems*, 4(4):662–675, 2018.