

LifeRaft: Data-Driven, Batch Processing for the Exploration of Scientific Databases

Xiaodan Wang¹, Randal Burns¹, Tanu Malik²

¹ Johns Hopkins University
{xwang, randal}@cs.jhu.edu

² Purdue University
tmalik@purdue.edu

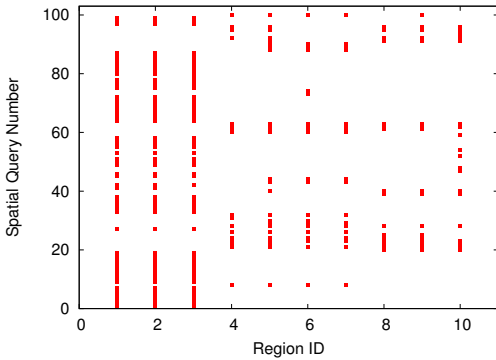


Fig. 1. Regions accessed by spatial queries in Astronomy.

I. PROJECT DESCRIPTION

Gray and Szalay [2] documented the data avalanche problem in the sciences in which improvements in physical instruments and better data pipelines lead to an exponential growth in data size. In Astronomy for example, the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS) produces tens of terabytes daily [3]. Exploring the resulting, massive amounts of data is of immense scientific value. However, as scientific repositories scale to petabyte datasets, data become too large to be stored on a single machine. Various scientific disciplines have turned to clustered or federated database environments to facilitate data exploration [5][6]. Data is typically partitioned spatially or temporarily across multiple nodes to achieve high degree of parallelism and aggregate throughput.

Increasingly, scientific discoveries are made through queries that scan large amounts of data to find correlations, mine data, and extract features from datasets that are distributed across multiple nodes. In the SkyQuery Astronomy federation, this emerging class of queries are both long running (lasting several hours or an entire day) and data intensive (performing full database scans). Moreover, SkyQuery services millions of queries each month in which multiple scan-intensive queries may execute simultaneously that place a substantial bottleneck on the disk and limit both query throughput and the scale of exploration.

We propose new query processing disciplines that maximize

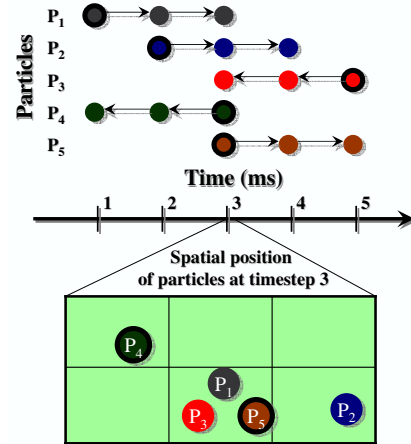


Fig. 2. Tracking the movement of particles in Turbulence both forwards and backwards in time.

throughput by alleviating contention for shared I/O resources through batch scheduling of concurrently executing queries. Our solution, LifeRaft, targets and co-schedules queries that access the same data to (1) eliminate redundant accesses to the disk and (2) amortize the cost of data access over multiple queries. Co-scheduling is possible because workloads access datasets that are indexed and partitioned spatially or temporarily such that the data requirements of queries are known prior to execution.

We envision a query processing system whereby scientists submit batch workloads consisting of long running “needle in a haystack” queries for which the user is willing to wait. LifeRaft will then reorder queries to exploit opportunities for data sharing during query execution. Batch processing is used effectively for scheduling queries over data stored on magnetic tape [7] such that tapes are accessed sequentially, and in Map-Reduce [1] to perform shared scans of the same file that is accessed by multiple map tasks. However, LifeRaft exploits data sharing even if queries only overlap partially and goes beyond reordering queries to achieve sequential data processing.

Our target application for batching are visual exploration in Astronomy [6] and multi-scale simulation in Turbulence clusters [5]. Both involve multi-terabyte datasets with scan-

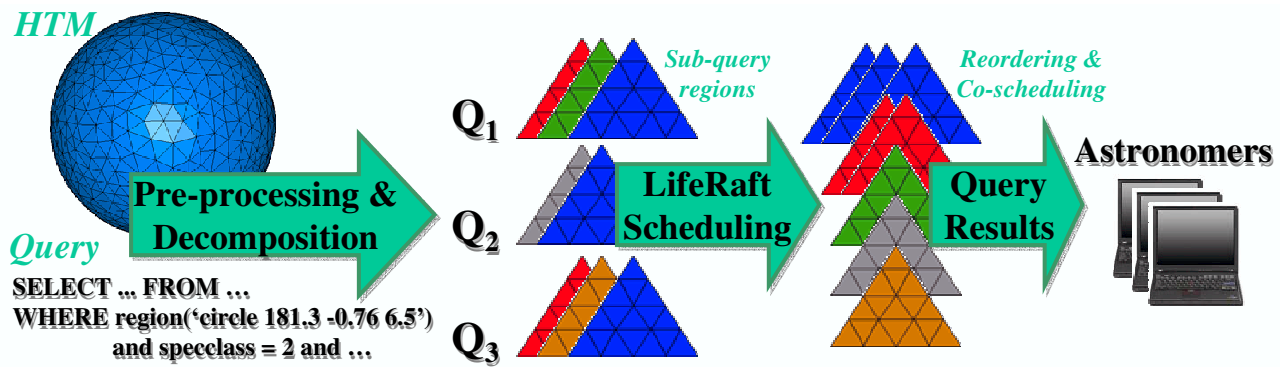


Fig. 3. Query processing procedure in LifeRaft batch scheduling.

intensive queries that benefit from data sharing. Figure 1 illustrates a sequence of queries of the form: retrieve astronomical objects with certain spectral properties from a spatial region. Each point in the graph denotes that the queried region overlaps a specific region in the sky. Clearly, effective query co-scheduling can avoid multiple, unnecessary passes over the same data. In Turbulence simulations, the movement of particles are tracked over multiple time steps. Queries exhibit temporal dependency (computation at the current timestep depends on the result of a prior timestep), which constrains when different queries can be co-scheduled. Figure 2 illustrates the tracking of particles moving both forwards and backwards in time along with the spatial position of particles at timestep three. Two tracked particles can share data access only if they are co-scheduled during the same timestep.

II. DESIGN

We put forth LifeRaft: a data-driven scheduler that relaxes in-order scheduling to achieve large improvements in query throughput without imposing undue wait times on individual queries. This is accomplished by processing queries in three stages to exploit opportunities for data sharing. Figure 3 illustrates this for Astronomy in which the sky is partitioned using a Hierarchical Triangular Mesh [4] (a regular quad-tree decomposition of a spherical surface into triangles). First each query is pre-processed to identify its data access requirements (*i.e.* determining the spatial regions of interest). Next, query decomposition sub-divides each query into sub-queries such that each sub-query operates on a small data region and the original query is answered by combining the sub-query results. Finally, rather than execute queries in order of arrival, sub-queries are reordered and co-scheduled based on contention for shared data to maximize throughput.

We have evaluated LifeRaft for Astronomy workloads in a single system environment, which demonstrates two-fold improvement in query throughput. We are extending the scheduler to multi-node clustered environments and queries with data dependencies such as particle tracking experiments in Turbulence simulation. Thus, our poster will detail LifeRaft scheduling techniques along with performance evaluation in both Astronomy and Turbulence applications.

REFERENCES

- [1] P. Agrawal, D. Kifer, and C. Olston. Scheduling Shared Scans of Large Data Files. In *VLDB*, 2008.
- [2] J. Gray and A. Szalay. Where the Rubber Meets the Sky: Bridging the Gap Between Databases and Science. *IEEE Data Engineering Bulletin*, 27(4):3–11, 2004.
- [3] N. Kaiser, H. Aussel, B. Burke, H. Boesgaard, K. Chambers, M. Chun, J. Heasley, K. Hodapp, B. Hunt, R. Jedicke, D. Jewitt, R. Kudritzki, G. Luppino, M. Maberry, E. Magnier, D. Monet, P. Onaka, A. Pickles, P. H. Rhoads, T. Simon, A. Szalay, I. Szapudi, D. Tholen, J. Tonry, M. Waterson, and J. Wick. Pan-STARRS: A Large Synoptic Survey Telescope Array. In *SPIE*, 2002.
- [4] P. Kunszt, A. Szalay, I. Csabai, and A. Thakar. The Indexing of the SDSS Science Archive. In *ADASS*, 2000.
- [5] Y. Li, E. Perlman, M. Wan, Y. Yang, C. Meneveau, R. Burns, S. Chen, A. Szalay, and G. Eyink. A Public Turbulence Database Cluster and Applications to Study Lagrangian Evolution of Velocity Increments in Turbulence. *Journal of Turbulence*, 9(31):1–29, 2008.
- [6] A. Szalay, J. Gray, A. Thakar, P. Kuntz, T. Malik, J. Raddick, C. Stoughton, and J. Vandenberg. The SDSS SkyServer - Public Access to the Sloan Digital Sky Server Data. In *SIGMOD*, 2002.
- [7] J.-B. Yu and D. J. DeWitt. Query Pre-Execution and Batching in Paradise: A Two-Pronged Approach to the Efficient Processing of Queries on Tape-Resident Raster Images. In *SSDBM*, 1997.