Chapter V
Data–Aware Distributed Batch Scheduling

Tevfik Kosar
Louisiana State University, USA

ABSTRACT

As the data requirements of scientific distributed applications increase, the access to remote data becomes the main performance bottleneck for these applications. Traditional distributed computing systems closely couple data placement and computation, and consider data placement as a side effect of computation. Data placement is either embedded in the computation and causes the computation to delay, or performed as simple scripts which do not have the privileges of a job. The insufficiency of the traditional systems and existing CPU-oriented schedulers in dealing with the complex data handling problem has yielded a new emerging era: the data-aware schedulers. In this chapter, we discuss the challenges in this area as well as future trends, with a focus on Stork case study.

INTRODUCTION

Modern scientific applications and experiments become increasingly data intensive. Large experiments, such as high-energy physics simulations, genome mapping, and climate modeling generate data volumes reaching hundreds of terabytes (Hey, 2003). Similarly, data collected from remote sensors and satellites are also producing extremely large amounts of data for scientists (Tummala and Kosar, 2007; Ceyhan & Kosar, 2007). In order to process these data, scientists are turning towards distributed resources owned by the collaborating parties to provide them the computing power and
storage capacity needed to push their research forward. But the use of distributed resources imposes new challenges (Kosar, 2006). Even simply sharing and disseminating subsets of the data to the scientists’ home institutions is difficult. The systems managing these resources must provide robust scheduling and allocation of storage resources, as well as the efficient management of data movement.

One key benefit of distributed resources is that it allows institutions and organizations to gain access to resources needed for large-scale applications that they would not otherwise have. But in order to facilitate the sharing of compute, storage, and network resources between collaborating parties, middleware is needed for planning, scheduling, and management of the tasks as well as the resources. The majority of existing research has been on the management of compute tasks and resources, as they are widely considered to be the most expensive. As scientific applications become more data intensive, however, the management of storage resources and data movement between the storage and compute resources is becoming the main bottleneck. Many jobs executing in distributed environments are failed or are inhibited by overloaded storage servers. These failures prevent scientists from making progress in their research.

According to the ‘Strategic Plan for the US Climate Change Science Program (CCSP)’, one of the main objectives of the future research programs should be “Enhancing the data management infrastructure”, since “The users should be able to focus their attention on the information content of the data, rather than how to discover, access, and use it.” (CCSP, 2003). This statement by CCSP summarizes the goal of many cyberinfrastructure efforts initiated by DOE, NSF and other federal agencies, as well the research direction of several leading academic institutions.

NSF’s ‘Cyberinfrastructure Vision for 21st Century’ states that “The national data framework must provide for reliable preservation, access, analysis, interoperability, and data movement” (NSF, 2006). The same report also says: “NSF will ensure that its efforts take advantage of innovation in large data management and distribution activities sponsored by other agencies and international efforts as well.” According to the NSF report on ‘Research Challenges in Distributed Computing Systems’, “Data storage is a fundamental challenge for large-scale distributed systems, and advances in storage research promise to enable a range of new high-impact applications and capabilities” (NSF, 2005).

It would not be too bold to claim that the research and development in the computation-oriented distributed computing has reached its maturity, and now there is an obvious shift of focus towards data-oriented distributed computing. This is mainly due to the fact that existing solutions work very well for computationally-intense applications, but inadequately address applications which access, create, and move large amounts of data over wide-area networks.

Accessing and transferring widely distributed data can be extremely inefficient and can introduce unreliability. For instance, an application may suffer from insufficient storage space when staging-in the input data, generating the output, and staging-out the generated data to a remote storage. This can lead to trashing of the storage server and subsequent timeout due to too many concurrent read data transfers—ultimately causing server crashes due to an overload of write data transfers. Other third party data transfers may stall indefinitely due to loss of acknowledgment. And even if transfer is performed efficiently, faulty hardware involved in staging and hosting can cause data corruption. Furthermore, remote access will suffer from unforeseeable contingencies such as performance degradation due to unplanned data transfers, and intermittent network outages.

Traditional distributed computing systems closely couple data handling and computation. They consider data resources as second class entities, and access to data as a side effect of
Related Content

Statically Optimal Binary Search Tree Computation Using Non-Serial Polyadic Dynamic Programming on GPU’s
www.igi-global.com/article/statically-optimal-binary-search-tree-computation-using-non-serial-polyadic-dynamic-programming-on-gpus/216481?camid=4v1a

A Comparative Study and Algorithmic Analysis of Workflow Decomposition in Distributed Systems
www.igi-global.com/article/a-comparative-study-and-algorithmic-analysis-of-workflow-decomposition-in-distributed-systems/216482?camid=4v1a

A Novel System Oriented Scheduler for Avoiding Haste Problem in Computational Grids
www.igi-global.com/chapter/novel-system-oriented-scheduler-avoiding/69026?camid=4v1a

Location Update Improvement Using Fuzzy Logic Optimization in Location Based Routing Protocols in MANET
www.igi-global.com/article/location-update-improvement-using-fuzzy/56352?camid=4v1a