Utility-Based Data Transfer Scheduling between Distributed Computing Facilities

Xin Wang,* Wei Tang,† Rajkumar Kettimuttu,† Zhiling Lan*

*Department of Computer Science, Illinois Institute of Technology
Chicago, IL 60616, USA
xwang149@hawk.iit.edu, lan@iit.edu

†Mathematics and Computer Science Division
Argonne National Laboratory, Argonne, IL 60439, USA
{wtang, kettimut}@mcs.anl.gov

***Abstract*—Today's scientific applications increasingly involve large amounts of input/output data that must be moved among multiple computing facilities via wide-area networks (WANs). The bandwidth of WANs, however, is growing at a much smaller rate and thus becoming a bottleneck. Moreover, the network bandwidth has not been viewed as a limited resource, and thus coordinated allocation is lacking. Uncoordinated scheduling of competing data transfers over shared network links results in suboptimal system performance and poor user experiences. To address these problems, we propose a data transfer scheduler to coordinate and schedule data transfers between distributed computing facilities over WANs.**

I. Introduction

Today's scientific applications increasingly involve large amounts of data. The needs for bulk data transfer between remote data centers or computing facilities are growing. Moving the increasing volume of data has imposed a heavy load on the networks between data centers. Especially when multiple data transfers compete for the limited bandwidth, coordinating and scheduling these transfers have become a challenge.

To address these problems, we designed and developed a data transfer scheduler to coordinate data transfer requests between distributed computing facilities. Our goal is to *process data transfer requests in an coordinated fashion in order to improve system performance as well as user satisfaction*. Unlike commonly used first-come, first-served (FCFS) or short-job-first (SJF) approaches, our approach quantifies user satisfaction by using a time-utility function (TUF) and schedules data transfer jobs in both temporal and spatial dimensions trying to maximize the aggregate job utility [3]. Here the utility function is a function of job turnaround time and can be used to represent the value (or utility) that the user attaches to the job completion. Maximizing aggregate job utility is consistent with an enhanced overall user satisfaction regarding job turnaround.

II. Problem Statement

We target our problem in a distributed environment where each source host has GridFTP service that handles all data transfer requests by moving data to multiple destinations over WANs.

For example, in Figure 1 a source host is connected with three destination hosts via a router. A scheduler inside the source host coordinates the data transfers between the source host and the respective destination hosts, via the links with different bandwidths. The decisions made by the scheduler include which jobs should be started now and how many TCP connections should be assigned to each job.

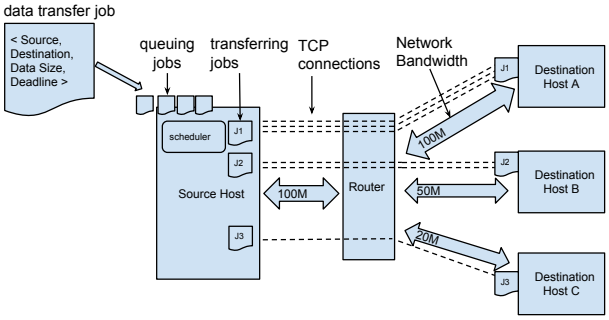
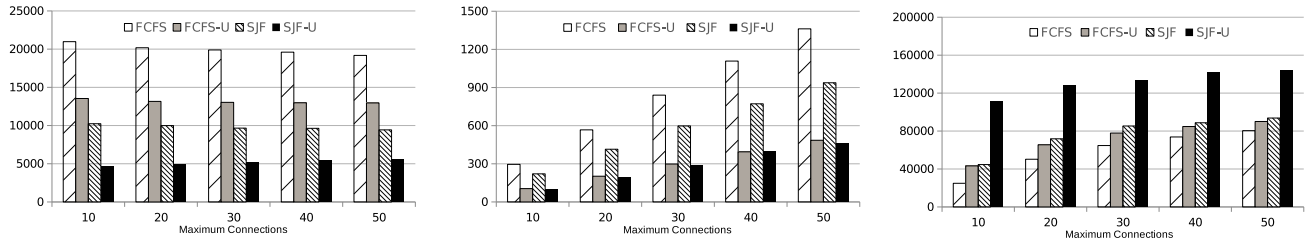


Figure 1. Example of data transfer scheduling problem. A source host is connected with three destination hosts via a router. A scheduler inside the source host will coordinate the data transfers and makes following decisions at each scheduling iteration: (1) which jobs should be started now and (2) how many TCP connections to assign to each job. The scheduling goal is to improve both system performance and user satisfaction.

To generalize the problem, we need a scheduler that coordinates the data transfer jobs by prioritizing jobs in the queue and allocates bandwidths to jobs by assigning TCP connections. The goal of the scheduler is to achieve improved system performance and user satisfaction. Improved system performance can be measured by reduced average job turnaround and network contentions. User satisfaction here can be represented by aggregate job utility.

III. Methodology

Our scheduling algorithm comprises three major steps. First, we try to allocate bandwidths for jobs with different destination with certain fairness guarantees. Second, for the jobs with same destination, we prioritize these jobs and select candidate jobs based on a window size. Third,



(a) avg. job response time (sec) (b) avg. transfer time (sec) (c) aggregate utility

Figure 2. Performance evaluation for different scheduling policies under various maximum TCP connection configurations. The legends of (a) through (c) represent different scheduling policies. The x-axis is the number of maximum TCP connections.

we apply a utility optimization algorithm to assign TCP connections to selected jobs.

1) Bandwidth Allocation: To allocate the shared bandwidths fairly, we use max-min fairness [4], a technique widely used in practice. The sharing of a network resource is "max-min fair" when the following conditions hold: the lowest demand on data rate is maximized; only after the lowest demand on the network resource has been satisfied will the second-lowest demand on the network resource be maximized; and so on.

2) Job Prioritizing and Selection: Once we have assigned available TCP connections to each job queue, we then iterate with each of the queues and select a set of candidate jobs that can be started at the current scheduling iteration. That is, the jobs will be sorted with certain policies, and the jobs at the head of the queue will be selected.

3) Utility-Based Connection Allocation: Once we select a set of candidate jobs for each queue, we want to decide how many TCP connections for each job to use for data transfers. The more connections a job uses, the more bandwidth it can occupy and the faster the data transfer. Therefore, we can use the connection assignment to favor certain important jobs—jobs that deliver more utility—to achieve aggregate user satisfaction.

IV. Experiments

We apply our new scheduler to a real case study where data movements are conducted between multiple XSEDE [2] sites. To evaluate our approach, we developed an open-source event-driven data transfer scheduling simulator named Dsim [1].

Using real job traces from multiple XSEDE sites, we demonstrated that our utility-based data transfer scheduler can achieve enhanced system performance and user satisfaction compared with traditional FCFS and SJF approaches in terms of reduced average job response time, less network contention, and aggregate job utility.

Figure 2 illustrates the overall results. The x-axis indicates different value of maximum connections. The y-axis shows the average performance results among all the jobs.

Figure 2(a) shows the results for average response time. We first observe the utility-based scheduling improves the average response time for the respective queuing policies. Figure 2(b) shows the results for average transfer time. As shown in the figure, using utility-based scheduling improves the data transfer time significantly, no matter what queuing policy is used. Figure 2(c) shows that using utility-based scheduling can significantly improve aggregate job utility, especially when the queuing policy is SJF.

V. Conclusion

In this work, we have presented a utility-based data transfer scheduler to coordinate and schedule bulk data transfers among distributed computing facilities via wide-area networks. We implemented our algorithms in an open-source data transfer scheduling simulator and conducted trace-based simulations using real job traces collected from production computing facilities. The experimental results demonstrate that our approach considerably improves the system performance and user satisfaction compared with traditional scheduling methods FCFS and SJF. Our utility optimization model improves job response time, transfer time and aggregate utility.

References

[1] Dsim project repo: *https://github.com/xwang149/Dsim*

[2] XSEDE: Extreme Science and Engineering Discovery Environment. https://www.xsede.org

[3] C. B. Lee, and A. E. Snavely, "Precise and realistic utility functions for user-centric performance analysis of schedulers," in *Proc. of 16th international symposium on High Performance Distributed Computing*, 2007.

[4] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica,"Dominant resource fairness: fair allocation of multiple resource types," in *Proc. of NSDI'11*, pp. 323-336, 2011