Datagridflows: Managing Long-Run Processes on Datagrids

Arun Jagatheesan[1,2], Jonathan Weinberg[1], Reena Mathew[1], Allen Ding[1], Erik Vandekieft[1], Daniel Moore[1,3], Reagan Moore[1], Lucas Gilbert[1], Mark Tran[1], and Jeffrey Kuramoto[1]

[1] San Diego Supercomputer Center,
University of California, San Diego,
9500 Gilman Drive, MC0505, La Jolla, CA 92093
{arun, jonw, alding, moore, iktome, kuramoto}@sdsc.edu
{rmathew, evandeki}@cs.ucsd.edu
mixx@umail.ucsb.edu
mdtran@ucsd.edu
[2] Institute of High Energy Physics and Astrophysics,
University of Florida,
Gainesville, FL 32611
[3] Department of Computer Science,
University of California, Santa Barbara, CA93106

Abstract. This paper is an introduction to *Datagridflows*. Until recently, datagrids were generally considered over-hyped and the associated technologies not widely embraced in the academic community. Today, datagrids have become a reality and an important technology for managing large, unstructured data and storage resources distributed over autonomous administrative domains. The datagrids that are operating in production provide us an idea of new requirements and challenges that will be faced in future datagrid environments. One such requirement is the coordinated execution of long-run data management processes in datagrids. We term these processes as "datagridflows". This new area provides exciting opportunities and challenges to researchers in distributed computing and distributed databases. This paper is intended to introduce these challenges to other researchers, including those new to grid computing. We provide motivation through discussion of datagridflow requirements and real production scenarios. We introduce current work on datagridflow technologies including the *Datagrid Language (DGL)* for describing datagridflows in datagrids.

1 Introduction

Datagrid technology is currently used for managing very large, unstructured data storage resources [1, 2, 3]. The need for long-run data management processes on top of datagrid environments is seen as a common emerging requirement in most datagrid deployments. Examples of these long-run processes include datagrid information lifecycle management, datagrid triggers, and data-intensive computational workflows. These long-run processes could be considered "datagrid workflows" and are discussed later in this paper. We refer to these long-run datagrid processes as datagridflows.

J.-M. Pierson (Ed.): VLDB DMG 2005, LNCS 3836, pp. 113 – 128, 2005.

In the following section, we introduce some fundamental concepts in datagrids for the benefit of those new to grid computing. In section 2, we describe three motivating scenarios for datagridflows and our work on the Data Grid Language. We discuss the requirements and components of a system to manage datagridflows in section 3. In section 4, we provide some overview of our work on the Data Grid Language as part of the SRB Matrix Project. Related and future works to this paper are presented in section 5.

1.1 Data Grid Landscape

In this section, we introduce datagrids, associated concepts and relevant terminology to prepare the reader for the problem statement discussed in the following sections.

Grid Computing. We describe a "grid" as a coordinated distributed computing infrastructure, formed by combining heterogeneous resources from autonomous administrative domains. Grids provide the infrastructure that is used for large-scale, resource-intensive, and distributed applications. The definition of a Grid is continually evolving as different people have different perspectives of the same technology. The commonality that is observed in the different perspectives of the "Grid" is the formation of a logical infrastructure as a single ensemble, by dynamically combining independently managed resources.

Datagrid. A datagrid is a logical unified view of a grid's data storage infrastructure. Data storage middleware create a federated, location independent, logical infrastructure namespace that dynamically spreads across the grid's administrative domains. Datagrids support sharing data collections and storage resources between autonomous administrative domains. A shared collection is a logical aggregation of digital entities, (e.g.) files, which are physically distributed in multiple physical storage resources that are owned by multiple administrative domains. A shared resource allows users from multiple administrative domains to share data storage space. The core concept behind the success for datagrid software is the concept of "data virtualization".

Data Virtualization. Data Virtualization is the concept of bringing together different heterogeneous data and storage resources into one or more *logical views* so that the distributed and replicated data appear as a single logical data source managed by a single data management system. This logical view is simple for users and applications as it hides the complexity of working with distributed and heterogeneous systems. The logical view is provided on top of a logical resource namespace, allowing high levels of flexibility for distributed computing and migration of data storage resources. Data and resource names are logical and can be physically changed or migrated without affecting the applications. The underlying concept behind the datagrids and data virtualization is the same as the concept behind relational databases: *to isolate physical organization of the data from logical schema.* In data virtualization, we go one step further. Instead of completely hiding the physical organization of the storage resources where the data resides, another logical namespace of storage resources is provided to the applications. Applications now have the added capability to perform distributed data management operations on the combined logical data namespace