Architecture for Composable Scientific Data Analysis

Jaliya Ekanayake, Shrideep Pallickara and Geoffrey Fox
Department of Computer Science
Indiana University, Bloomington, IN 47404, USA
{jekanaya,spallick,gcf}@indiana.edu

Abstract

In this paper we present a framework for data analysis within scientific applications that possess the composable property. In our scheme we rely on the concept of moving computations to the data.

Keywords

composable applications, data analysis

1. Motivation

In recent years there has been an exponential growth in the amount of data that needs to be processed by scientific applications. Most of this data is distributed across different geographical locations. Some examples of this include the various genome, seismology, and climate related databases distributed around the world.

Efficient processing of voluminous, distributed data can be achieved by performing the analysis (either entirely or a large portion of it) in close network-proximity to the data. We expect this approach – *moving computation to the data* – will allows us to accrue significant performance benefits because it precludes the need to do voluminous data transfers.

The availability of results in near real-time is an especially critical feature in domains such as meteorology and seismology. This real-time availability is also desirable in particle physics since it allows physicists to perform multiple analyses till a desired goal is achieved.

We are interested in the class of scientific applications where the processing exhibits the composable property. Here, the processing can be split into smaller computations, and the partial-results from these computations merged after some post-processing to constitute the final result. This is distinct from the tightly-coupled parallel applications where the synchronization constraints are typically in the order of microseconds instead of the 50-200 millisecond coupling constraints in composable systems. The level of coupling between the sub-computations is higher than those in the decoupled-model such as the multiple independent sub-tasks in job processing systems such as Nimrod.

To delve a little deeper into the *composable* property, consider a common practice in many particle physics data analysis where the aim is to identify particular type of events within a collection of millions of events. The result of such analysis would be a histogram of possible events. We can easily break the initial set of events into smaller subsets and run the same analysis software on these subsets concurrently. The resulting histograms can then be combined to produce the final result.

Other domains where the composable property is applicable include genome sequencing and seismic data analysis within the scientific community. Google's *map-reduce* framework [1] is also based on the composability property.

1.1 Computation Model

In our scheme, we assume that the scientist has access to a set of machines, and that these machines are able to access the data needed for the computations. Scientists have continued to prefer this model due to its responsiveness and the ability to launch multiple concurrent experiments while enforcing security constraints. This is quite distinct from the BOINC (Berkeley Open Infrastructure for Network Computing) and computational grids model which focuses on harnessing idle CPU cycles. This typically involves moving both the data and the code comprising the computation to the harvested node. We argue that for a class of scientific data analysis "moving computation to data" is a better model to adopt.

1.2 Problem Statement

The crux of this thesis is a framework to manage the execution and subsequent

[Thesis-URL]Additional materials (Related work, manuscripts and links to software available at http://www.cs.indiana.edu/~jekanaya/research.html collaborative analysis of large scientific experiments that have the composable property. The framework will be designed to ensure that it is domain-independent, secure and resilient to failures.

2. System Architecture

The high-level architecture of the system is depicted in Figure 1. There are three phases that comprise this analysis framework viz. execute, analyze and collaborate. The execution phase manages the distribution, tracking and completion of computations on the compute clouds followed by the generation of result streams, which are routed by the content dissemination network [2], comprising partial-results from individual constituent computations of a given experiment.

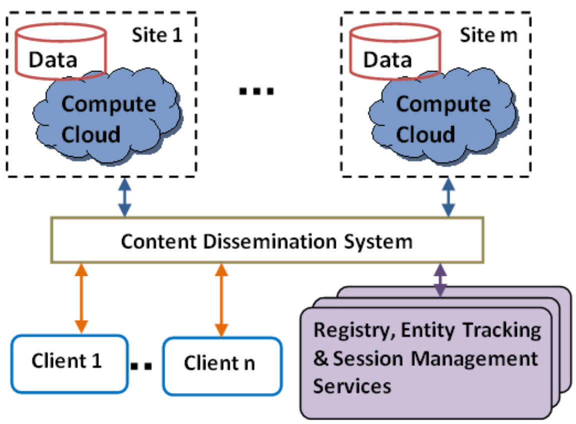

Figure1. Architecture of the proposed data analysis framework.

During the analysis phase the aforementioned result streams are consumed and post-processing related to assimilating newly computed partial results with the existing set of results for that experiment is performed. An example of post-processing, in the particle physics domain, involves updating histograms to reflect a newly computed result.

During the collaboration phase, the analysis can be performed (or visualized) by multiple concurrent users. The set of gossiping agents keep track of the computational tasks and the memberships associated with them.

3. Research so far

As part of an effort funded by the DoE we are working with the High Energy Physics group at Caltech with their particle physics analysis tool – CLARENS, which allows scientists to launch computations related to their analysis, followed by visualization of histograms produced by these computations. Using our prototype framework, which utilizes NaradaBrokering (developed by us) for distributing computations and streams, we have successfully converted this particle physics application into a collaborative one. This application was demonstrated at *Super Computing 2007*. The paper describing our results for the collaborative aspects of this framework can be found at Ref [Thesis-URL].

We will be using the lessons learned with the particle physics application towards building the generalized infrastructure that would be applicable to domains such as those where the statistical package R has been deployed. We are currently focusing on applications where the data analysis algorithms are non-hierarchical; we also plan to support to hierarchical algorithms.

4. Expected Contributions

This thesis will provision a scalable architecture for real-time composable data analysis tasks. The framework will provide faster processing because of concurrent execution and minimal data movement. The architecture will be failure resilient and enforce security constraints such as authorization, integrity and confidentiality.

The outcomes of this thesis will be domain-independent. Ultimately, we expect that for the class of applications that have the composable property, this research would make the scientific analysis process faster, collaborative and more efficient. We also expect this research to a class of applications that use a variant of workflow that satisfies real-time execution constraints [3].

5. Timelines

The first author is in the first year (after completion of Masters) of his PhD. An early domain-specific (particle physics) prototype is currently available. Currently, the student is expected to graduate in Dec 2009.

References

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in OSDI'04: 6th Symposium on Operating Systems

[2] S. Pallickara and G. Fox NaradaBrokering: A Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids. Proc of ACM/IFIP/USENIX Middleware Conference, 2003.

[3]N. Vydyanathan et.al., "Toward Optimizing Latency Under Throughput Constraints for Application Workflows on Clusters", *Euro-Par 2007 Parallel Processing*, Springer Berlin / Heidelberg, Vol. 4641,2007,pp. 173-183