University of South Carolina  
Department of Computer Science and Engineering  
Ph.d. Dissertation Defense  
Candidate: Annamaria V. Kish

Dissertation Title: Efficient Partitioning and Allocation of Data for Workflow Compositions  
Advisor: Dr. Csilla Farkas  
Date: April 14, 2016  
Time: 12:00PM  
Place: Swearingen 3A75

ABSTRACT

Our aim is to provide efficient partitioning and allocation of data for web service compositions. Web service compositions are represented as partial order database transactions. We accommodate a variety of transaction types (such as read-only and write-oriented transactions) to support workloads in cloud environments. We introduce an approach that partitions and allocates small units of data, called micropartitions, to multiple database nodes. Each database node stores only the data needed to support a specific workload. Transactions are routed directly to the appropriate data nodes. Our approach guarantees serializability and efficient execution.

In Phase 1, we cluster transactions based on data requirements. We associate each cluster with an abstract query definition. An abstract query represents the minimal data requirement that would satisfy all the queries that belong to a given cluster. A micropartition is generated by executing the abstract query on the original database.

We show that our abstract query definition is complete and minimal. Intuitively, completeness means that all queries of the corresponding cluster can be correctly answered using the micropartition generated from the abstract query. The minimality property means that no smaller partition of the data can satisfy all of the queries in the cluster.

We also aim to support efficient web services execution. Our approach reduces the number of data accesses to distributed data. We also aim to limit the number of replica updates. Our empirical results show that the partitioning approach improves data access efficiency over standard partitioning of data.

In Phase 2, we investigate the performance improvement via parallel execution. Based on the data allocation achieved in Phase I, we develop a scheduling approach. Our approach guarantees serializability while efficiently exploiting parallel execution of web services.

We achieve conflict serializability by scheduling conflicting operations in a predefined order. This order is based on the calculation of a minimal delay requirement. We use this delay to schedule services to preserve serializability without the traditional locking mechanisms.