Procedia Computer Science
Volume 80, 2016, Pages 472–483
ICCS 2016. The International Conference on Computational Science

Network Services and Their Compositions for Network Science Applications

Sherif Elmeligy Abdelhamid[1], Chris J. Kuhlman[2], Madhav V. Marathe[1,2], and S. S. Ravi[3]

[1] Department of Computer Science, Virginia Tech, Blacksburg, USA.
sherief@vt.edu
[2] Biocomplexity Institute of Virginia Tech, Blacksburg, USA.
{ckuhlman,mmarathe}@vbi.vt.edu
[3] Department of Computer Science, University at Albany, Albany, USA.
ravi@cs.albany.edu

Abstract
Network science is moving more and more to computing *dynamics* on networks (so-called contagion processes), in addition to computing structural network features (e.g., key players) and other parameters. Generalized contagion processes impose additional data storage and processing demands that include more generic and versatile manipulations of networked data that can be highly attributed. In this work, we describe a new network services and workflow system called MARS that supports structural network analyses and generalized network dynamics analyses. It is accessible through the internet and can serve multiple simultaneous users and software applications. In addition to managing various types of digital objects, MARS provides services that enable applications (and UIs) to add, interrogate, query, analyze, and process data. We focus on several network services and workflows of MARS. We also provide a case study using a web-based application that MARS supports, and several performance evaluations of scalability and work loads. We find that MARS efficiently processes networks of hundreds of millions of edges from many hundreds of simultaneous users.

Keywords: Network Science, Data Management, Services, Workflow Systems, Performance Evaluation

1 Introduction

1.1 Background and Motivation

Network science is the study of physical systems that are represented as networks. Various disciplines make use of graph abstractions—biology [9], sociology [13], and health sciences [17], to name a few. Many software tools exist for computing structural properties (i.e., measures) on networks (e.g., SNAP, NetworkX, Pajek). Far fewer systems exist for computing (contagion) *dynamics* on networks, particularly on large networks with 1 million or more vertices.

 Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.326

(Typically for contagion dynamics, network vertices (nodes) represent agents with behaviors, and edges represent agent interactions. States of agents change over time due to neighborhood interactions.) Running these codes requires computing expertise in scripting to generate input and to post-process output files, and are typically not accessible to those without computing skills [11]. To provide accessibility to non-computing experts, still fewer systems combine high performance computing (HPC) with an intuitive user interface (UI) as part of a web application (web app), an example being EDISON [1].

Network services are essential for these types of web apps: they provide (meta)data storage, query, and analysis facilities that are used by both UIs and (HPC) simulation engines. In network science and dynamics, these data include agent dynamics models, their properties, and validity ranges; and networks and their vertex and edge attributes. Services include query, storage and retrieval, computations of network structural properties, and provenance. A specific example task is to find all vertices with degree greater than 50 and kshell at least 25, and to those nodes, assign the threshold pair (2,6) for dynamics computations. The required properties (degree and kshell) may or may not exist for the specified network, and if they do not, then they are computed on the fly, results are returned to the requestor, and both the properties and the query results are stored in a repository to service other requests. These and other requirements motivated the development of a workflow system that we refer to as MARS.

1.2 Contributions

A summary of our major contributions follows.

1. MARS Workflow Service. MARS is a server-based workflow system developed for network science scientific computing. It operates by concurrently servicing multiple users and multiple applications and is accessed through the internet, using a well-defined API. It was designed to support both GUI functionality as well as HPC computational requirements, but the system is application-agnostic and thus not tied to either. It contains a repository along with several categories of services, but we confine ourselves to network services and their workflows in this paper. These services store networks, compute measures on them, manipulate subsets of network vertices and edges, and query data. MARS also uses external applications to support services, thereby increasing its capabilities. These applications currently run on an HPC computing cluster (MARS uses a PBS scheduler to launch executables) and are oblivious to the services that invoke them. MARS can be reconfigured to run these third-party codes on other platforms.

The services are stand-alone executing processes that can reside on different compute nodes for increased performance through data locality and for flexibility in mapping processes to hardware. As we will demonstrate, workflows are composed not only of sequences of services, but also of interleaved functions across services. Both stateless (i.e., REST-ful) and stateful (i.e., *session*) interactions with external applications are supported. All aspects of stateful interactions (hand-shaking, coordinating multiple requests by the same application for the same session) are handled completely by MARS. A customized grammar for SQL-like queries is used to support special requirements for network dynamics. While network science is the target application for this work, a large part of this system is general; e.g., a new query parser could be inserted for another application and the software used to compute measures on networks can be changed out for other applications.

2. MARS and Big Data. The MARS repository houses many types of digital objects (DOs). There are multiple categories for DOs; we focus here only on the networks that are stored for simulations because they are central for computations in this domain. We use the relational