Preface

Distributed systems enable the sharing, selection, and aggregation of a wide variety of distributed resources for solving large-scale, resource (computation and/or data)-intensive problems in science. Workflows represent a main programming model for the development of scientific applications. Creating scientific workflow applications, however, is still a very challenging task for scientists due to the complexity of distributed computing environments, the complex control and data flow requirements of scientific applications, and the lack of high-level languages and tool support. Particularly, sophisticated expertise in distributed computing is often required to determine the software entities to perform computations of workflow tasks, the computers on which workflow tasks are executed, the execution order of workflow tasks, and the data communications among them. The composition of an optimized scientific workflow can be even more difficult. Existing work suffers from being limited to specific implementation technologies, modeling low-level tasks, limited mechanisms to express iterative, conditional, and parallel execution, naive dataset distribution, no semantic support, no automatic workflow composition, or automatic workflow composition limited for special cases, etc.

This book presents a novel workflow language called Abstract Workflow Description Language (AWDL) and the corresponding standards-based, knowledge-enabled tool support with the aim to simplify the development of scientific workflow applications, as well as to optimize and to synthesize them. AWDL is an XML-based language for describing scientific workflow applications at a high level of abstraction. AWDL is designed such that users can concentrate on specifying scientific workflow applications without dealing with either the complexity of distributed computing environments or any specific implementation technology. A rich set of control flow constructs is provided in AWDL to simplify the specification of scientific workflow applications which includes directed acyclic graphs, conditional branches, parallel and sequential iterative constructs, alternative executions, etc. Properties and constraints can be specified in AWDL to provide additional information for workflow runtime environments to optimize and steer the execution of workflow applications. The AWDL modularization mechanism, including sub-workflows and workflow libraries, enables easy reuse and sharing of

scientific workflow applications among research groups. To streamline scientific workflow composition, a standards-based approach for modeling scientific workflow applications using the Unified Modeling Language (UML) Activity Diagram is presented, along with the corresponding graphical scientific workflow composition tool, which can automatically generate AWDL code based on graph representations of scientific workflows.

To meet the complex dataset-oriented data flow requirements of scientific workflow applications, AWDL introduces a data collection concept and the corresponding collection distribution constructs, which are inspired by High Performance Fortran (HPF). With these constructs, more fine-grained data flow can be specified at an abstract workflow language level, such as mapping a portion of a dataset to an activity, and independently distributing multiple collections onto parallel loop iterations. The use of these constructs improves the performance of scientific workflows by reducing data duplications and simplifies the effort to port scientific workflow applications onto distributed systems.

With the help of Semantic Web technologies, AWDL introduces a novel semantic based approach for scientific workflow composition. This approach features separation of concerns between data semantics and data representations and between Activity Functions (AFs) and Activity Types (ATs). It simplifies scientific workflow composition by enabling knowledge support and automatic data conversion. On the basis of this semantic approach, an Artificial Intelligence planning based algorithm for automatic control flow composition of scientific workflows using an Activity Function Data Dependence (ADD) is presented. The algorithm employs progression to create an ADD graph and regression to extract workflows, including alternative ones if available. The extracted workflows are then optimized based on data dependence analysis. Following automatic control flow composition, data flow composition is automated by semantically matching each data sink in scientific workflows against the corresponding data sources obtained through backward control flow traversing.

The newly introduced techniques and algorithms are implemented in the framework of the ASKALON development and runtime environment for workflows on distributed systems. The effectiveness of our techniques is demonstrated through a variety of experiments on a distributed computing infrastructure.

The topic covered in this book is of interest to a broad range of computer science researchers, domain scientists who are interested in applying workflow technologies in their work, and engineers who want to develop workflow systems, languages, and tools.

Munich, Germany — Jun Qin
Innsbruck, Austria — Thomas Fahringer
April 2012

http://www.springer.com/978-3-642-30714-0

Scientific Workflows
Programming, Optimization, and Synthesis with ASKALON and AWDL
Qin, J.; Fahringer, Th.
2012, XXII, 222 p., Hardcover
ISBN: 978-3-642-30714-0