



Doctoral Thesis

Workflow-based Services Infrastructure for Scientific Applications

Author(s):

Heinis, Thomas

Publication Date:

2009

Permanent Link:

<https://doi.org/10.3929/ethz-a-005884768> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

Diss. ETH No. 18217

Workflow-based Services: Infrastructure for Scientific Applications

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Technical Sciences

presented by
THOMAS HEINIS
Dipl. Informatik-Ing. ETH
born October 13, 1977
citizen of Therwil BL

accepted on the recommendation of
Prof. Dr. Gustavo Alonso, examiner
Prof. Dr. Donald Kossmann, co-examiner
Prof. Dr. Bertram Ludaescher, co-examiner
Prof. Dr. Cesare Pautasso, co-examiner
Dr. Bernd Wollscheid, co-examiner

2009

Abstract

The way scientists work in traditional sciences has changed drastically in recent years. Computer science is increasingly supporting them in performing and analyzing their experiments. Today, obtaining the raw data from instruments is merely the first step. No longer is the data analyzed on paper or with simple computational tools. Instead, massive amounts of data obtained raw from instruments are processed in complex and long-running computational pipelines. This trend of supporting traditional sciences with computational tools has led to a significant speedup in executing experiments and has also enabled experiments which would not have been possible before.

Scientists increasingly depend on adequate infrastructure to process experiment data using complex computational pipelines and to manage the plethora of data used and produced by them. Such computational pipelines are typically modeled as workflows and so this trend consequently challenges the current infrastructure for executing workflows as well as the infrastructure to manage the resulting data deluge. The work presented in this book addresses some of the challenges arising from this trend.

A first challenge this book addresses is the sharing of scientific computations. Publishing a methodology for others to use is no longer possible with the scientific computations or workflows people develop today. Implementing such workflows and setting up the appropriate environment is far too difficult and consequently such computations or workflows must be shared. State-of-the-art approaches use proprietary interfaces to make such workflows available as services to be used by others. Proprietary interfaces, however, make it difficult to integrate such computations and we therefore use standardized interfaces. We describe in detail how we have mapped a workflow to such a standardized interface and report on an efficient implementation.

The second challenge this book examines is how to provide adequate infrastructure for scientific workflows. One considerable advantage of scientific workflows is that they can be parametrized and one workflow can be run several thousand times for processing vast numbers of raw data sets obtained from instruments. Such large numbers of executions mandate the execution engine to be distributed in order to scale. However, configuring such a distributed engine is very difficult. Also, if it is configured optimally for just one type of workload, it may not be suitable for another. For this reason, we have added an autonomic controller to a distributed workflow execution engine. This controller monitors the behavior of the engine and adapts the configuration on the fly to serve the current workload best, adding self-configuration and self-optimization properties. The controller also heals the system in case of failures, thereby adding self-healing properties.

Finally, this book discusses the problem of data lineage tracking. Running such workflows thousandfold leads to an abundance of data products. The lineage of each such data product therefore has to be tracked, i.e., it has to be tracked what computational tools were involved in producing a given item. Without such information the data product cannot be properly contextualized and understood. Current approaches to the problem do not scale. We have developed a novel approach which can store lineage information fast, but more importantly, also allows the efficient retrieval of the highly intricate lineage information.

Kurzfassung

Die Arbeitsweise von Forschern in traditionellen Wissenschaften hat sich in den letzten Jahren fundamental verändert. Informatikmittel unterstützen sie in zunehmendem Masse bei der Durchführung und Analyse ihrer Experimente. Heutzutage ist das Erfassen der rohen Experimentdaten von Instrumenten nur noch der erste Schritt. Experimentdaten werden nicht mehr auf Papier oder mit simplen Computerwerkzeugen analysiert. Viel mehr werden die massiven Mengen von Daten die von den Instrumenten gewonnen werden mit komplexen Computerwerkzeugen und lang laufenden Softwarepipelines verarbeitet. Dieser Trend zur Unterstützung von traditionellen Wissenschaften durch Informatikmittel hat das Durchführen von einigen Experimenten substantiell beschleunigt und andere Experimente erst möglich gemacht.

Mit diesem Trend sind Wissenschaftler in einem immer grösser werdenden Masse von adäquater Infrastruktur abhängig um Experimente mit komplexen Softwarepipelines durchzuführen und um die resultierenden Datenmengen zu verwalten. Solche Softwarepipelines werden üblicherweise als Workflows modelliert und dieser Trend hinterfragt die gängige Infrastruktur für Workflows wie auch die Infrastruktur zur Handhabung der riesigen Mengen von resultierenden Daten. Diese Dissertation nimmt sich in diesem Zusammenhang drei solcher Herausforderungen an.

Eine erste Herausforderung die diskutiert wird ist das gemeinsame Benutzen von solchen Workflows. Die Idee des Publizieren einer wissenschaftlichen Methode damit sie von anderen Forschern benutzt werden kann ist mit den komplexen Workflows welche heute entwickelt werden nicht mehr möglich. Viel zu kompliziert ist das Implementieren und Bereitstellen der notwendigen Infrastruktur und konsequenterweise müssen solche Workflows als Services gemeinsam benutzt werden. Heutige Ansätze zur Lösung des Problems verwenden proprietäre Schnittstellen um diese Workflows anderen Benutzern zur Verfügung zu stellen. Solche Schnittstellen allerdings erschweren die Integration von Workflows. Wir beschreiben deswegen detailliert wie wir Workflows auf standardisierte Schnittstellen abbilden und beschreiben eine effiziente Implementierung davon.

Eine weitere Herausforderung die in der Dissertation behandelt wird ist adäquate Infrastruktur zur Ausführung von Workflows zur Verfügung zu stellen. Ein gewichtiger Vorteil von Workflows ist es, dass diese parametrisiert und tausendfach für verschiedene Datensätze ausgeführt werden können. Solche grosse Zahlen von Ausführungen erfordern aber auch eine verteilte Infrastruktur damit die Ausführung skaliert. Die Konfiguration einer solchen verteilten Infrastruktur ist allerdings sehr schwierig. Auch kann die Infrastruktur für eine bestimmte Arbeitslast ideal konfiguriert sein, für eine andere aber nicht. Aus diesem Grund haben wir die Infrastruktur mit einer autonomen Steuerung

ausgerüstet die das System zur Laufzeit neu konfiguriert und der aktuellen Arbeitslast anpasst. Die Steuerung überwacht das System und nimmt, falls notwendig, Änderungen daran vor um die aktuell ausgeführte Arbeitslast optimal auszuführen.

Schlussendlich behandelt die Dissertation auch das Problem der Datenprovenienz. Solche Workflows tausendfach auszuführen führt zu einer riesigen Menge von Daten. Die Provenienz von so erzeugten Daten muss unbedingt gespeichert werden, d.h. welche Werkzeuge & Daten zu deren Erstellung verwendet wurden. Ohne solche Provenienzinformation können solche Daten nicht korrekt kontextualisiert und interpretiert werden. Gängige Verfahren zur Speicherung solcher Daten skalieren nicht. Wir haben deswegen einen neuen Ansatz entwickelt welcher diese Information schnell speichern kann, und viel wichtiger, diese komplexe Information auch effizient abrufen kann.