



A novel general framework for automatic and cost-effective handling of recoverable temporal violations in scientific workflow systems

Xiao Liu^{a,*}, Zhiwei Ni^b, Zhangjun Wu^{b,a}, Dong Yuan^a, Jinjun Chen^a, Yun Yang^a

^a Faculty of Information and Communication Technologies, Swinburne University of Technology, Hawthorn, Melbourne 3122, Australia

^b Institute of Intelligent Management, School of Management, Hefei University of Technology, Hefei, Anhui 230009, China

ARTICLE INFO

Article history:

Received 14 June 2010

Received in revised form 20 October 2010

Accepted 20 October 2010

Available online 30 October 2010

Keywords:

Temporal violation

Exception handling

Scientific workflow

Workflow rescheduling

Workflow QoS

Temporal verification

ABSTRACT

Due to the complex nature of scientific workflow environments, temporal violations often take place and may severely reduce the timeliness of the execution's results. To handle temporal violations in an automatic and cost-effective fashion, two interdependent fundamental issues viz. the definition of fine-grained recoverable temporal violations and the design of light-weight effective exception handling strategies need to be resolved. However, most existing works study them separately without defining a comprehensive framework. To address such a problem, with the probability based temporal consistency model which defines the range of recoverable temporal violations, a novel general automatic and cost-effective exception handling framework is proposed in this paper where fine-grained temporal violations are defined based on the empirical function for the capability lower bounds of the exception handling strategies. To serve as a representative case study, a concrete example exception handling framework which consists of three levels of fine-grained temporal violations and their corresponding exception handling strategies is presented. The effectiveness of the example framework is evaluated by large scale simulation experiments conducted in the SwinDeW-G scientific grid workflow system. The experimental results demonstrate that the example framework can significantly reduce the overall average violation rates of local temporal constraints and global temporal constraints to 0.127% and 0.167% respectively.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

Scientific workflow systems are a type of workflow management systems aiming at supporting complex scientific processes in many e-science applications such as climate modelling, disaster recovery simulation, astrophysics and high energy physics (Deelman et al., 2008; Taylor et al., 2007). Scientific workflow systems can also be seen as a type of high-level middleware services for high performance computing infrastructures such as cluster, grid, peer-to-peer (p2p) or cloud computing (Buyya et al., 2009; Foster and Kesselman, 2004; Kim et al., 2007; Yang et al., 2007). In recent years, due to the growing demand for high performance computing infrastructures and large scale distributed and collaborative e-science applications, scientific workflow systems have been attracting increasing interests from distributed and parallel system researchers in the area of High Performance Computing (HPGC, 2009; PDESC, 2009) and software engineering researchers

in the area of Software Engineering for Computational Science and Engineering (Chen and Yang, in press; SECES, 2008). One of the common research issues is how to deliver satisfactory workflow QoS (quality of service), i.e. how to satisfy workflow QoS constraints such as the constraints on time, cost, fidelity, reliability and security (Son and Kim, 2001; Yu and Buyya, 2005). Among them, time is one of the basic measurements for system and software performance and hence attracts many researchers in the workflow area (van der Aalst et al., 2000; Chen and Yang, 2008; Duan et al., 2009; Eder et al., 1999; Li et al., 2004; Yu and Buyya, 2005; Zhuge et al., 2001).

In reality, a scientific workflow and its workflow segments are normally subject to specific temporal constraints such as global temporal constraints (deadlines) for workflow instances, and local temporal constraints (milestones) for workflow segments, in order to achieve predefined scientific goals on schedule (Li et al., 2004; Zeng et al., 2008). Otherwise, the timeliness of its execution results will be significantly deteriorated. For example, a daily weather forecast scientific workflow has to be finished before the broadcasting of the weather forecast programme everyday at, for instance, 6:00 pm. Meanwhile, given the large number of data and computation intensive activities for scientific investigation purposes, scientific workflows are usually deployed on distributed

* Corresponding author. Tel.: +61 03 92148699.

E-mail addresses: xliu@swin.edu.au (X. Liu), nzwgd@hfut.edu.cn (Z. Ni), wuzhangjun@hfut.edu.cn (Z. Wu), dyuan@swin.edu.au (D. Yuan), jchen@swin.edu.au (J. Chen), yyang@swin.edu.au (Y. Yang).

high performance infrastructures such as grid and cloud. Therefore, to deliver satisfactory temporal QoS, the violations of both local temporal constraints (or local violations for short) and global temporal constraints (or global violations for short), need to be proactively detected and handled (Zhuge et al., 2001). Recent studies on temporal verification in scientific workflows mainly focus on runtime checkpoint selection (Chen and Yang, in press) and multiple-state based temporal verification (Chen and Yang, 2007) which can deal with the monitoring of temporal consistency states and the detection of potential temporal violations. However, a significant follow-up issue is how to handle those temporal violations. Till date, work on such an issue is still in its infancy. However, it must be properly addressed so as to guarantee high success rates for on-time completion of scientific workflows. Specifically, two fundamental requirements for handling temporal violations, *automation* and *cost-effectiveness*, need to be considered.

- (1) *Automation*. Due to the complex nature of scientific applications and their distributed running environments such as grid and cloud, a large number of temporal violations may often be expected in scientific workflows. Besides, scientific workflow systems are designed to be highly automatic to conduct large scale scientific processes, human interventions which are normally of low efficiency should be avoided as much as possible, especially during workflow runtime (Deelman et al., 2008). Therefore, similar to dynamic checkpoint selection and temporal verification strategies (Chen and Yang, in press), handling strategies are required to automatically tackle a large number of temporal violations and relieve users from the heavy workload of handling those exceptions.
- (2) *Cost-effectiveness*. The purpose of handling temporal violations is to reduce, or ideally remove, the delays of workflow execution by exception handling strategies with the sacrifice of additional cost which consists of both monetary cost and time overheads. Conventional exception handling strategies for temporal violations, such as resource recruitment and workflow restructure, are usually very expensive (Buhr and Mok, 2000; Hagen and Alonso, 2000; Prodan and Fahringer, 2008; Russell et al., 2006a). The cost for recruiting new resources (e.g. the cost for service discovery and deployment, the cost for data storage and transfer) is normally very large during workflow runtime in distributed computing environments (Prodan and Fahringer, 2008). As for workflow restructure, it is usually realised by the amendment of local workflow segments or temporal QoS contracts, i.e. modifying scientific workflow specifications by human decision makers (Liu et al., 2008b). However, due to budget (i.e. monetary cost) limits and temporal constraints, these heavy-weight strategies (with large monetary cost and/or time overheads) are usually too costly to be practical. To avoid these heavy-weight strategies, recoverable violations (in comparison to severe temporal violations which can be regarded as non-recoverable in practice) need to be identified first and then handled by light-weight strategies (with small monetary cost and/or time overheads) in a cost-effective fashion.

Given the requirement of *Automation*, exception handling strategies need to be designed to handle temporal violations in an automatic fashion without human interventions. Meanwhile, since most strategies have their limits in the capability of recovering temporal violations, different handling strategies are normally only effective for a range of temporal violations with limited amount of time deficits (the time delays given specific temporal constraints). Given the requirement of *Cost-effectiveness*, for all the candidate strategies which are capable of handling the current temporal violation, ideally, only the one with the lowest cost should be applied. Therefore, the definition of fine-grained temporal violations and

the design of exception handling strategies should be investigated as two interdependent tasks within the same exception handling framework. However, since recent studies in temporal verification mainly focus on the detection of temporal violations, fine-grained temporal violations are usually defined for the general application purpose ignoring the performance of exception handling strategies in the specific workflow systems. For example, the work in Chen and Yang (2007) proposes a multiple-states based temporal consistency model. Besides SC (strong consistency) which requires no action, three types of fine-grained temporal inconsistency states including WC (weak consistency), WI (weak inconsistency) and SI (strong inconsistency) are defined based on the minimum, mean and maximum workflow execution time. However, without the investigation on the performance of different exception handling strategies, it is difficult to determine which strategy should be applied to handle the detected temporal violations. Therefore, it is more reasonable that fine-grained temporal violations should be defined according to the selection of different exception handling strategies with different capabilities, rather than most of the previous studies where fine-grained temporal violations are defined in the first place then looking for available exception handling strategies. To the best of our knowledge, this is the first work to systematically investigate a general exception handling framework for automatic and cost-effective handling of temporal violations in scientific workflow systems.

In this paper, along with the probability based temporal consistency model which defines the range of recoverable temporal violations, a novel general automatic and cost-effective exception handling framework is proposed. Specifically, fine-grained temporal violations are first defined based on the empirical function for the capability lower bounds of the exception handling strategies. Afterwards, to serve as a case study, a concrete example framework is presented which consists of three levels of fine-grained temporal violations, viz., level I, level II and level III temporal violations defined within the recoverable probability range, and three light-weight automatic exception handling strategies, viz., TDA (Time Deficit Allocation), ACOWR (Ant Colony Optimisation based two-stage Workflow local Rescheduling) and TDA + ACOWR (the combined strategy of TDA and ACOWR). Large scale simulation experiments are conducted in the SwinDeW-G scientific grid workflow system (Yang et al., 2007) to evaluate the effectiveness of the example framework.

The remainder of the paper is organised as follows. Section 2 presents a motivating example and the problem analysis. Section 3 proposes a general exception handling framework for temporal violations. Section 4 presents a case study with a concrete exception handling framework with three levels of temporal violations and their corresponding handling strategies. Section 5 demonstrates comprehensive simulation results. Section 6 reviews the related work. Finally, Section 7 concludes the paper and points out the future work.

2. Motivating example and problem analysis

2.1. Motivating example

In this section, we present an example scientific workflow in Astrophysics. Parkes Radio Telescope (<http://www.parkes.atnf.csiro.au/>, located 380 km west of Sydney, Australia), one of the most famous radio telescopes, is serving institutions around the world. Swinburne Astrophysics group has been conducting a pulsar searching survey based on the observation data from Parkes Radio Telescope (<http://astronomy.swin.edu.au/pulsar/>). The pulsar searching process is a typical scientific workflow which involves a large number

Download English Version:

<https://daneshyari.com/en/article/461372>

Download Persian Version:

<https://daneshyari.com/article/461372>

[Daneshyari.com](https://daneshyari.com)