UNIVERSITY OF
NEWCASTLE

University of Newcastle upon Tyne

COMPUTING SCIENCE

Wrapper Based Adaptive Management of Concurrent Workflow Executions

J. Smith and P. Watson

TECHNICAL REPORT SERIES

No. CS-TR-1063 December, 2007

TECHNICAL REPORT SERIES

No. CS-TR-1063 December, 2007

Wrapper Based Adaptive Management of Concurrent Workflow Executions

Jim Smith and Paul Watson

Abstract

This work relates to data centre systems, where a provider hosts applications, used in the context of Service Level Agreements (SLA). However, the applications are structured as composite services, i.e. workflows. Thus, the work relates also to service composition. In a possible scenario an organization exports a collection of application services, based on a pool of computational resources and a pool of (possibly redundant, possibly hierarchically composed) component services.

Printed and published by the University of Newcastle upon Tyne,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

SMITH, J., WATSON, P..

Wrapper Based Adaptive Management of Concurrent Workflow Executions
[By] J. Smith, P. Watson.

Newcastle upon Tyne: University of Newcastle upon Tyne: Computing Science, 2007.

(University of Newcastle upon Tyne, Computing Science, Technical Report Series, No. CS-TR-1063)

Added entries

UNIVERSITY OF NEWCASTLE UPON TYNE
Computing Science. Technical Report Series. CS-TR-1063

Abstract

This work relates to data centre systems, where a provider hosts applications, used in the context of Service Level Agreements (SLA). However, the applications are structured as composite services, i.e. workflows. Thus, the work relates also to service composition. In a possible scenario an organization exports a collection of application services, based on a pool of computational resources and a pool of (possibly redundant, possibly hierarchically composed) component services.

About the author

Jim Smith worked as a computer programmer for some years with the UK electricity supply industry, then moved to the University of Newcastle upon Tyne where, following studies for MSC and PHD, he is working as an RA on the Polar project.

Paul Watson is Professor of Computer Science and Director of the North East Regional e-Science Centre. He graduated in 1983 with a BSc (I) in Computer Engineering from Manchester University, followed by a PhD in 1986. In the 80s, as a Lecturer at Manchester University, he was a designer of the Alvey Flagship and Esprit EDS systems. From 1990-5 he worked for ICL as a system designer of the Goldrush MegaServer parallel database server, which was released as a product in 1994. In August 1995 he moved to Newcastle University, where he has been an investigator on research projects worth over £13M. His research interests are in scalable information management. This includes parallel database servers, data-intensive e-science and grid computing. In total, he has over thirty refereed publications, and three patents. Professor Watson is a Chartered Engineer, a Fellow of the British Computer Society, and a member of the UK Computing Research Committee.

Suggested keywords

DATA CENTER,
SERVICE COMPOSITION,
ADAPTIVE EXECUTION,
WORKFLOW

Wrapper Based Adaptive Management of Concurrent Workflow Executions

Jim Smith #1, Paul Watson *2

#School of Computing Science, Newcastle University, Newcastle upon Tyne, UK
1Jim.Smith@ncl.ac.uk, 2Paul.Watson@ncl.ac.uk

I. Introduction

This work relates to data centre systems, where a provider hosts applications, used in the context of Service Level Agreements (SLA). However, the applications are structured as composite services, i.e. workflows. Thus, the work relates also to service composition. In a possible scenario an organization exports a collection of application services, based on a pool of computational resources and a pool of (possibly redundant, possibly hierarchically composed) component services.

II. Wrapper Approach

A prototype adaptive workflow management system has been implemented by wrapping an established 'basic' workflow engine in "black-box" fashion, as shown in Figure 1(a). At deployment, a workflow is modified by a **Translator** to add extra identification parameters to each component invocation. Each invocation is intercepted by a **Broker**, which uses these identification parameters and statistics forwarded by probes, **P1** and **P2**, to determine, in accordance with a chosen policy, how to process that request. **Broker** controls the mapping of invocations to alternate service implementations and the deployment and use of services within the pool of available machines. An invocation for a composite service is passed to the **Basic engine**. Service implementations may be selected on the basis of properties recorded in a registry, and/or on the basis of directly measured attributes. See [1] for details.

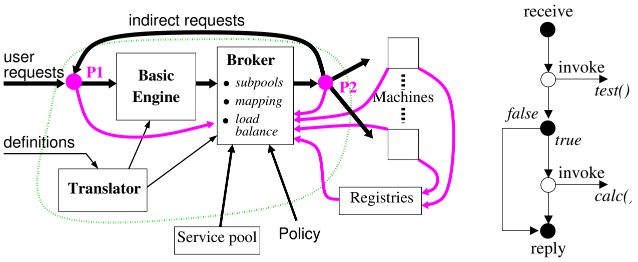

Fig. 1. (a) Wrapper approach and (b) example workflow **W**

III. Experiment

An example workflow **W**, shown in Figure1(b) has the form "**if** *test()* **then** *calc()*". The operations are dummy CPU heavy computations, and the ratio of *true*/*false* values returned as the boolean result of *test()* is configurable. Two implementations of *calc()* have different response times; the assumption being that another non-functional property (e.g. cost) varies accordingly. 200 requests are applied, at constant rate, to W. The rate of *true* returns from *test()* is increased from approximately 0.1 to 0.5 at request number 72, thereby inducing a step change in load. The size of the sub-pool allocated to the workload is kept at 4 machines. A set of alternate policies use measured workflow and component invocation response times to determine mappings; the results are shown in Figure 2. Two policies map all invocations of *calc()* to one or other implementation. Two switch dynamically between faster and slower implementations, seeking to achieve a specified average response time.

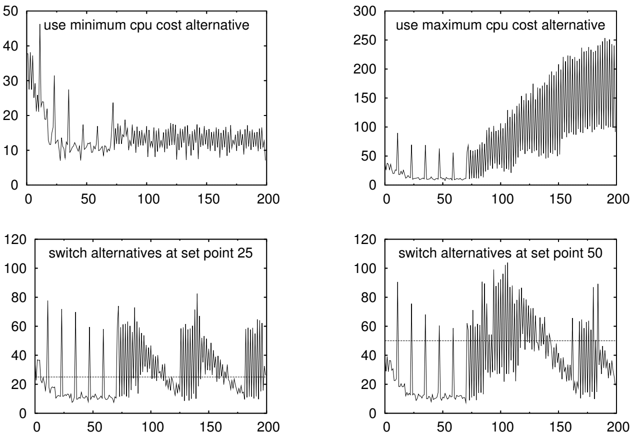

Fig. 2. Measured per-request response (seconds) for 200 request workloads.

IV. Discussion

Concurrent workflow management features both provider and user viewpoints. The prototype system described here wraps an establised basic engine to support adaptation from both viewpoints. The results presented show the possibility of employing service invocation mapping to control usage of machine resources. Experiments in [1] demonstrate use of sub-pool size to regulate response time and selection of component service implementation to meet a user's non-functional requirements. Ongoing work is investigating strategies for employing these controls in meeting service level agreements.

References

[1] J. Smith and P. Watson, "Experiments towards adaptation of concurrent workflows," in *ECOWS*. IEEE, Nov. 2007, to appear.