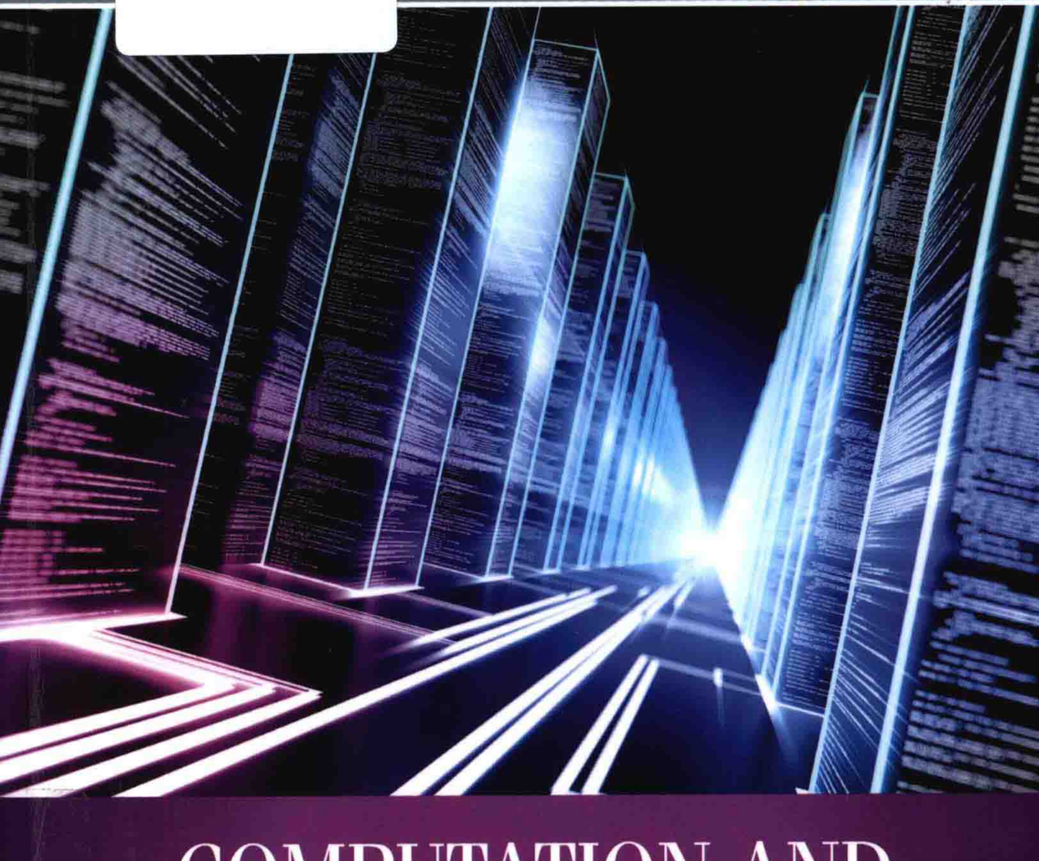
ELSEVIER INSIGHTS



COMPUTATION AND STORAGE IN THE CLOUD

UNDERSTANDING THE TRADE-OFFS

DONG YUAN, YUN YANG AND JINJUN CHEN

Computation and Storage in the Cloud

Understanding the Trade-Offs

Dong Yuan and Yun Yang

Centre for Computing and Engineering Software Systems,
Faculty of Information and Communication Technologies,
Swinburne University of Technology,
Hawthorn, Melbourne, Australia

Jinjun Chen

Centre for Innovation in IT Services and Applications,
Faculty of Engineering and Information Technology,
University of Technology,
Sydney, Australia

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON • NEW YORK • OXFORD
PARIS • SAN DIEGO • SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Elsevier
225 Wyman Street, Waltham, MA 02451, USA
32 Jamestown Road, London NW1 7BY

First edition 2013

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein.

In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

ISBN: 978-0-12-407767-6

This book has been manufactured using Print On Demand technology. Each copy is produced to order and is limited to black ink. The online version of this book will show color figures where appropriate.

Computation and Storage in the Cloud

Acknowledgements

The authors are grateful for the discussions with Dr. Willem van Straten and Ms. Lina Levin from the Swinburne Centre for Astrophysics and Supercomputing regarding the pulsar searching scientific workflow. This work is supported by the Australian Research Council under Discovery Project DP110101340.

About the Authors

Dong Yuan received his PhD degree in Computer Science and Software Engineering from the Faculty of Information and Communication Technologies at Swinburne University of Technology, Melbourne, Australia in 2012. He received his Master and Bachelor degrees from the School of Computer Science and Technology, Shandong University, Jinan, China in 2008 and 2005, respectively, all in Computer Science. He is currently a postdoctoral research fellow in the Centre of Computing and Engineering Software System at Swinburne University of Technology. His research interests include data management in parallel and distributed systems, scheduling and resource management, and grid and cloud computing.

Yun Yang received a Master of Engineering degree from the University of Science and Technology of China, Hefei, China in 1987, and a PhD degree from the University of Queensland, Brisbane, Australia in 1992, all in Computer Science. He is currently a full professor in the Faculty of Information and Communication Technologies at Swinburne University of Technology, Melbourne, Australia. Prior to joining Swinburne as an associate professor in late 1999, he was a lecturer and senior lecturer at Deakin University during 1996–1999. Before that, he was a research scientist at DSTC – Cooperative Research Centre for Distributed Systems Technology – during 1993–1996. He also worked at Beihang University in China during 1987–1988. He has published about 200 papers on journals and refereed numerous conferences. His research interests include software engineering; P2P, grid and cloud computing; workflow systems; service-oriented computing; Internet computing applications; and CSCW.

Jinjun Chen received his PhD degree in Computer Science and Software Engineering from Swinburne University of Technology, Melbourne, Australia in 2007. He is currently an associate professor in the Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia. His research interests include scientific workflow management and applications; workflow management and applications in Web service or SOC environments; workflow management and applications in grid (service)/cloud computing environments; software verification and validation in workflow systems; QoS and resource scheduling in distributed computing systems such as cloud computing, service-oriented computing, semantics and knowledge management; and cloud computing.

Preface

Nowadays, scientific research increasingly relies on IT technologies, where large-scale and high-performance computing systems (e.g. clusters, grids and super-computers) are utilised by the communities of researchers to carry out their applications. Scientific applications are usually computation and data-intensive, where complex computation tasks take a long time for execution and the generated data sets are often terabytes or petabytes in size. Storing valuable generated application data sets can save their regeneration cost when they are reused, not to mention the waiting time caused by regeneration. However, the large size of the scientific data sets makes their storage a big challenge.

In recent years, cloud computing is emerging as the latest distributed computing paradigm which provides redundant, inexpensive and scalable resources on demand to system requirements. It offers researchers a new way to deploy computation and data-intensive applications (e.g. scientific applications) without any infrastructure investments. Large generated application data sets can be flexibly stored or deleted (and regenerated whenever needed) in the cloud, since, theoretically, unlimited storage and computation resources can be obtained from commercial cloud service providers.

With the pay-as-you-go model, the total application cost for generated data sets in the cloud depends chiefly on the method used for storing them. For example, storing all the generated application data sets in the cloud may result in a high storage cost since some data sets may be seldom used but large in size; but if we delete all the generated data sets and regenerate them every time they are needed, the computation cost may also be very high. Hence, there is a trade-off between computation and storage in the cloud. In order to reduce the overall application cost, a good strategy is to find a balance to selectively store some popular data sets and regenerate the rest when needed. This book focuses on cost-effective data sets storage of scientific applications in the cloud, which is currently a leading-edge and challenging topic. By investigating the niche issue of computation and storage trade-off, we (1) propose a new cost model for data sets storage in the cloud; (2) develop novel benchmarking approaches to find the minimum cost of storing the application data; and (3) design innovative runtime storage strategies to store the application data in the cloud.

We start with introducing a motivating example from astrophysics and analyse the problems of computation and storage trade-off in the cloud. Based on the requirements identified, we propose a novel concept of Data Dependency Graph (DDG) and propose an effective data sets storage cost model in the cloud. DDG is based on data provenance, which records the generation relationship of all the data sets. With DDG, we know how to effectively regenerate data sets in the cloud and can further calculate their generation costs. The total application cost for the generated data sets includes both their generation cost and their storage cost.

Based on the cost model, we develop novel algorithms which can calculate the minimum cost for storing data sets in the cloud, i.e. the best trade-off between computation and storage. This minimum cost is a benchmark for evaluating the cost-effectiveness of different storage strategies in the cloud. For different situations, we develop different benchmarking approaches with polynomial time complexity for a seemingly NP-hard problem, where (1) the static on-demand approach is for situations in which only occasional benchmarking is requested; and (2) the dynamic on-the-fly approach is suitable for situations in which more frequent benchmarking is requested at runtime.

We develop novel cost-effective storage strategies for users to facilitate at runtime of the cloud. These are different from the minimum cost benchmarking approach, and sometimes users may have certain preferences regarding storage of some particular data sets due to reasons other than cost – e.g. guaranteeing immediate access to certain data sets. Hence, users' preferences should also be considered in a storage strategy. Based on these considerations, we develop two cost-effective storage strategies for different situations: (1) the cost-rate-based strategy is highly efficient with fairly reasonable cost-effectiveness; and (2) the local-optimisation-based strategy is highly cost-effective with very reasonable time complexity.

To the best of our knowledge, this book is the first comprehensive and systematic work investigating the issue of computation and storage trade-off in the cloud in order to reduce the overall application cost. By proposing innovative concepts, theorems and algorithms, the major contribution of this book is that it helps bring the cost down dramatically for both cloud users and service providers to run computation and data-intensive scientific applications in the cloud.

Contents

1 Introduction

This book investigates the trade-off between computation and storage in the cloud. This is a brand new and significant issue for deploying applications with the pay-as-you-go model in the cloud, especially computation and data-intensive scientific applications. The novel research reported in this book is for both cloud service providers and users to reduce the cost of storing large generated application data sets in the cloud. A suite consisting of a novel cost model, benchmarking approaches and storage strategies is designed and developed with the support of new concepts, solid theorems and innovative algorithms. Experimental evaluation and case study demonstrate that our work helps bring the cost down dramatically for running the computation and data-intensive scientific applications in the cloud.

This chapter introduces the background and key issues of this research. It is organised as follows. Section 1.1 gives a brief introduction to running scientific applications in the cloud. Section 1.2 outlines the key issues of this research. Finally, Section 1.3 presents an overview for the remainder of this book.

1.1 Scientific Applications in the Cloud

Running scientific applications usually requires not only high-performance computing (HPC) resources but also massive storage [34]. In many scientific research fields, like astronomy [33], high-energy physics [61] and bioinformatics [65], scientists need to analyse a large amount of data either from existing data resources or collected from physical devices. During these processes, large amounts of new data might also be generated as intermediate or final products [34]. Scientific applications are usually data intensive [36,61], where the generated data sets are often terabytes or even petabytes in size. As reported by Szalay et al. in [74], science is in an exponential world and the amount of scientific data will double every year over the next decade and on into the future. Producing scientific data sets involves a large number of computation-intensive tasks, e.g., with scientific workflows [35], and hence takes a long time for execution. These generated data sets contain important intermediate or final results of the computation, and need to be stored as valuable resources. This is because (i) data can be reused – scientists may need to re-analyse the results or apply new analyses on the existing data sets [16] – and (ii) data can be shared – for collaboration, the computation results may be shared, hence the data sets are used by scientists from different institutions [19]. Storing valuable generated application data sets can save their regeneration cost when they are reused, not to mention the waiting time caused by regeneration. However, the large size of the scientific data sets presents a serious challenge in terms of storage. Hence, popular scientific applications are often deployed in grid or HPC systems [61] because they have HPC resources and/or massive storage. However, building and maintaining a grid or HPC system is extremely expensive and neither can easily be made available for scientists all over the world to utilise.

In recent years, cloud computing is emerging as the latest distributed computing paradigm which provides redundant, inexpensive and scalable resources on demand to system requirements [42]. Since late 2007 when the concept of cloud computing was proposed [83], it has been utilised in many areas with a certain degree of success [17,21,45,62]. Meanwhile, cloud computing adopts a pay-as-you-go model where users are charged according to the usage of cloud services such as computation, storage and network[1] services in the same manner as for conventional utilities in everyday life (e.g., water, electricity, gas and telephone) [22]. Cloud computing systems offer a new way to deploy computation and data-intensive applications. As Infrastructure as a Service (IaaS) is a very popular way to deliver computing resources in the cloud [1], the heterogeneity of the computing systems [92] of one service provider can be well shielded by virtualisation technology. Hence, users can deploy their applications in unified resources without any infrastructure investment in the cloud, where excessive processing power and storage can be obtained from commercial cloud service providers. Furthermore, cloud computing systems offer a new paradigm in which scientists from all over the world can collaborate and conduct their research jointly. As cloud computing systems are usually based on the Internet, scientists can upload their data and launch their applications in the cloud from anywhere in the world. Furthermore, as all the data are managed in the cloud, it is easy to share data among scientists.

However, new challenges also arise when we deploy a scientific application in the cloud. With the pay-as-you-go model, the resources need to be paid for by users; hence the total application cost for generated data sets in the cloud highly depends on the strategy used to store them. For example, storing all the generated application data sets in the cloud may result in a high storage cost since some data sets may be seldom used but large in size, but if we delete all the generated data sets and regenerate them every time they are needed, the computation cost may also be very high. Hence there should be a trade-off between computation and storage for deploying applications; this is an important and challenging issue in the cloud. By investigating this issue, this research proposes a new cost model, novel benchmarking approaches and innovative storage strategies, which would help both cloud service providers and users to reduce application costs in the cloud.

[1] In this book, we investigate only the trade-off between computation and storage, where a network is not incorporated. Please refer to Section 3.2.2 for detailed explanations.

1.2 Key Issues of This Research

In the cloud, the application cost highly depends on the strategy of storing the large generated data sets due to the pay-as-you-go model. A good strategy is to find a balance to selectively store some popular data sets and regenerate the rest when needed, i.e. finding a trade-off between computation and storage. However, the generated application data sets in the cloud often have dependencies; that is, a computation task can operate on one or more data set(s) and generate new one(s). The decision about whether to store or delete an application data set impacts not only the cost of the data set itself but also that of other data sets in the cloud. To achieve the best trade-off and utilise it to reduce the application cost, we need to investigate the following issues:

1. *Cost model.* Users need a new cost model that can represent the amount that they actually spend on their applications in the cloud. Theoretically, users can get unlimited resources from the commercial cloud service providers for both computation and storage. Hence, for the large generated application data sets, users can flexibly choose how many to store and how many to regenerate. Different storage strategies lead to different consumptions of computation and storage resources and ultimately lead to different total application costs. The new cost model should be able to represent the cost of the applications in the cloud, which is the trade-off between computation and storage.
2. *Minimum cost benchmarking approaches.* Based on the new cost model, we need to find the best trade-off between computation and storage, which leads to the theoretical minimum application cost in the cloud. This minimum cost serves as an important benchmark for evaluating the cost-effectiveness of storage strategies in the cloud. For different applications and users, cloud service providers should be able to provide benchmarking services according to their requirements. Hence benchmarking algorithms need to be investigated, so that we develop different benchmarking approaches to meet the requirements of different situations in the cloud.
3. *Cost-effective dataset storage strategies.* By investigating the trade-off between computation and storage, we determine that cost-effective storage strategies are needed for users to use in their applications at run-time in the cloud. Different from benchmarking, in practice, the minimum cost storage strategy may not be the best strategy for the applications in the cloud. First, storage strategies must be efficient enough to be facilitated at run-time in the cloud. Furthermore, users may have certain preferences concerning the storage of some particular data sets (e.g. tolerance of the accessing delay). Hence we need to design cost-effective storage strategies according to different requirements.

1.3 Overview of This Book

In particular, this book includes new concepts, solid theorems and complex algorithms, which form a suite of systematic and comprehensive solutions to deal with the issue of computation and storage trade-off in the cloud and bring cost-effectiveness to the applications for both users and cloud service providers. The remainder of this book is organised as follows.

In Chapter 2, we introduce the work related to this research. We start by introducing data management in some traditional scientific application systems, especially in grid systems, and then we move to the cloud. By introducing some typical cloud systems for scientific application, we raise the issue of cost-effectiveness in the cloud. Next, we introduce some works that also touch upon the issue of computation and storage trade-off and analyse the differences to ours. Finally, we introduce some works on the subject of data provenance which are the important foundation for our own work.

In Chapter 3, we first introduce a motivating example: a real-world scientific application from astrophysics that is used for searching for pulsars in the universe. Based on this example, we identify and analyse our research problems.

In Chapter 4, we first give a classification of the application data in the cloud and propose an important concept of data dependency graph (DDG). DDG is built on data provenance which depicts the generation relationships of the data sets in the cloud. Based on DDG, we propose a new cost model for datasets storage in the cloud.

In Chapter 5, we develop novel minimum cost benchmarking approaches with algorithms for the best trade-off between computation and storage in the cloud. We propose two approaches, namely static on-demand benchmarking and dynamic on-the-fly benchmarking, to accommodate different application requirements in the cloud.

In Chapter 6, we develop innovative cost-effective storage strategies for user to facilitate at run-time in the cloud. According to different user requirements, we design different strategies accordingly, i.e. a highly efficient cost-rate-based strategy and a highly cost-effective local-optimisation-based strategy.

In Chapter 7, we demonstrate experiment results to evaluate our work as described in the entire book. First, we introduce our cloud computing simulation environment, i.e. SwinCloud. Then we conduct general random simulations to evaluate the performance of our benchmarking approaches and storage strategies. Finally, we demonstrate a case study of the pulsar searching application in which all the research outcomes presented in this book are utilised.

Finally, in Chapter 8, we summarise the new ideas presented in this book and the major contributions of this research.

In order to improve the readability of this book, we have included a notation index in Appendix A; all proofs of theories, lemmas and corollaries in Appendix B; and a related method in Appendix C.

2 Literature Review

This chapter reviews the existing literature related to this research. It is organised as follows. In Section 2.1, we summarise the data management work about scientific applications in the traditional distributed computing systems. In Section 2.2, we first review some existing work about deploying scientific applications in the cloud and raise the issue of cost-effectiveness; we then analyse some research that has touched upon the issue of the trade-off between computation and storage and point out the differences to our work. In Section 2.3, we introduce some work about data provenance which is the important foundation for our work.

2.1 Data Management of Scientific Applications in Traditional Distributed Systems

Alongside the development of information technology (IT), e-science has also become increasingly popular. Since scientific applications are often computation and data intensive, they are now usually deployed in distributed systems to obtain high-performance computing resources and massive storage. Roughly speaking, one can make a distinction between two subgroups in the traditional distributed systems [11]: clusters (including the HPC system) and grids.

Early studies about data management of scientific applications are in cluster computing systems [9]. Since cluster computing is a relative homogenous environment that has a tightly coupled structure, data management in clusters is usually straightforward. The application data are commonly stored according to the system's capacity and moved within the cluster via a fast Ethernet connection while the applications execute.

Grid computing systems [40] are more heterogeneous than clusters. Given the similarity of grid and cloud [42], we mainly investigate the existing related work about grid computing systems in this section. First, we present some general data management technologies in grid. Then, we investigate the data management in some grid workflow systems which are often utilised for running scientific applications. Finally, we briefly introduce the data management technologies in some other distributed systems.