James Jay Kistler

Disconnected Operation in a Distributed File System

Springer

Series Editors

Gerhard Goos
Universität Karlsruhe
Vincenz-Priessnitz-Straße 3, D-76128 Karlsruhe, Germany

Juris Hartmanis
Department of Computer Science, Cornell University
4130 Upson Hall, Ithaca, NY 14853, USA

Jan van Leeuwen
Department of Computer Science,Utrecht University
Padualaan 14, 3584 CH Utrecht, The Netherlands

Author

James Jay Kistler
Systems Research Center, Digital Equipment Corporation
130 Lytton Avenue, Palo Alto, CA 94301, USA

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Kistler, James Jay:
Disconnected operation in a distributed file sytem / James Jay Kistler. - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 1996
(Lecture notes in computer science ; 1002)
ISBN 3-540-60627-0

NE: GT

CR Subject Classification (1991): D.4, C.2.4, E.5, F.2.2

ISBN 3-540-60627-0 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10486054 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

For Chris

Foreword

Tension between *autonomy* and *interdependence* lies at the heart of every distributed system. The ability to use remote resources enhances the storage capacity and computational power of a client. But there is a price to be paid: reliance on remote resources renders the client vulnerable to failures of the network or servers. In the worst case, a client can be totally crippled by the unavailability of a remote resource. This problem is already serious today, and will only worsen with further growth in the size and complexity of distributed systems.

How can one alleviate this problem? The traditional approach has been to use *replication* at servers. Unfortunately, there are limits to the value of this approach. It comes at significant hardware cost. Worse, it is useless if a network failure isolates a client from all server replicas. The latter scenario is especially common in *mobile computing,* where intermittent connectivity is an unfortunate fact of life.

In this doctoral dissertation, Jay Kistler describes a radically different solution. His approach, called *disconnected operation,* calls upon a client to mask failures from users and applications by emulating the functionality of a server. For distributed file systems, Kistler observes that this emulation can efficiently and cheaply be performed by exploiting the file cache already maintained by the client for performance reasons. This leads to a tantalizingly simple design: it just requires pre-loading the cache with critical data, continuing normal operation until disconnection, logging all changes made while disconnected, and replaying them upon reconnection.

Of course, reality is never that simple. There are many conceptual and implementation problems that arise when one tries to implement this functionality. For example:

- How does one arrange to have the right files in the cache at disconnection?
- How does the client conserve scarce cache space while disconnected?
- How can the process of reintegrating changes be made efficient and transparent?
- What are the security implications of disconnected operation?
- How can one reconcile the conflicting demands of availability and performance on caching?

- How can disconnected operation be seamlessly integrated with server replication?
- How likely are update conflicts, and how does one detect and cope with them?

Kistler answers these and many related questions in this work. The implementation he describes is of such high quality that it has been in serious use for over four years. The evaluation of the system is thorough and it sheds much light on the above questions. In addition, the work describes a new model of computation called the "inferred transaction model" that cleanly captures the semantics of disconnected operation. This model serves as an excellent conceptual foundation for reasoning about the consistency properties of disconnected operation.

This research has had substantial impact on industry. Many software companies have efforts under way to exploit these results commercially. There is now broad consensus that disconnected operation is a key enabling technology for mobile computing.

In closing, it is a pleasure to read a dissertation that is such a model of clarity and lucid exposition. The subtleties of the subproblems are brought out with great skill, and the specific solutions adopted are convincingly substantiated. The document is indeed worthy of the research it describes. I would expect no less of my friend, colleague, and former graduate student, Jay Kistler.

M. Satyanarayanan
Professor of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania
August 1995

Preface

Disconnected operation refers to the ability of a distributed system client to operate despite server inaccessibility by emulating services locally. The capability to operate disconnected is already valuable in many systems, and its importance is growing with two major trends: the increasing scale of distributed systems, and the proliferation of powerful mobile computers. The former makes clients vulnerable to more frequent and less controllable system failures, and the latter introduces an important class of clients which are disconnected frequently and for long durations – often as a matter of choice.

This dissertation shows that it is practical to support disconnected operation for a fundamental system service: general purpose file management. It describes the architecture, implementation, and evaluation of disconnected file service in the Coda file system. The architecture is centered on the idea that the disconnected service agent should be one and the same with the client cache manager. The Coda cache manager prepares for disconnection by pre-fetching and *hoarding* copies of critical files; while disconnected it logs all update activity and otherwise *emulates* server behavior; upon reconnection it *reintegrates* by sending its log to the server for replay. This design achieves the goal of high data availability – users can access many of their files while disconnected – but it does not sacrifice the other positive properties of contemporary distributed file systems: scalability, performance, security, and transparency.

Disconnected operation in Coda was designed and implemented during the period of 1989 to 1993. At the time this dissertation was completed, the system had been actively used by more than 20 people over the course of two years. Both stationary and mobile workstations had been employed as clients, and disconnections had ranged up to about ten days in length. Usage experience was extremely positive. The hoarding strategy sufficed to avoid most disconnected cache misses, and partitioned data sharing was rare enough to cause very few reintegration failures. Measurements and simulation results indicated that disconnected operation in Coda should be equally transparent and successful at much larger scale.

Since 1993 the system has continued to be used as a research vehicle at Carnegie Mellon University. The number of local users has grown significantly and the code has been made available for distribution outside of CMU.

Coda researchers have extended the system with important new functionality, including weakly-connected operation, advanced hoarding support, and transactional file system semantics. Reports of this progress are beginning to appear and to generate discussion in the academic literature.

Disconnected file service is also beginning to make its mark in the commercial world, driven by the tremendous success of mobile computers in the marketplace. Products offering limited forms of disconnected file support have been available from small companies for several years, and industry heavyweights such as IBM, DEC, and Microsoft are beginning to weigh-in with efforts of their own. These initial products are all hampered to some degree by the legacy of PC operating systems, but the obstacles are rapidly being overcome. I am certain that we will see more – and better – products of this type in the near future. Indeed, I am as confident today about the future of disconnected operation as I was two years ago when I wrote the final sentence of this dissertation: *The advantages of disconnected file service are so compelling that its support will – in my opinion – be a standard feature of all widely-used operating environments of the future.*

Acknowledgments

Performing the thesis research and writing this dissertation turned out to be a larger undertaking than I ever imagined. I could not have completed it without the care and support of many wonderful people, and I'm delighted to be able to acknowledge them here.

First, I would like to thank my advisor, Satya. I couldn't have had a better mentor. He always made time to see me, no matter how busy his schedule. He was a constant source of good ideas and an infallible detector of bad ones. He challenged me when I needed to be challenged and boosted my confidence when it needed to be boosted. He taught me the importance of critical thinking and of validating one's ideas through experimentation. More than anything, though, he has been a true and steady friend.

The other members of my thesis committee were helpful throughout my career at CMU. Early on, Rick Rashid and Eric Cooper co-advised me and made me feel comfortable as I was finding my way. Later, they gave sound advice on my topic and on research in general. Mike Schroeder helped to expose and formulate the problem that my work addresses, and his careful reading and critiquing of the dissertation made it a much better document. I thank all three of them for their efforts.

I warmly thank my colleagues in the Coda group: Maria Ebling, Puneet Kumar, Qi Lu, Hank Mashburn, Lily Mummert, Brian Noble, Josh Raiff, Ellen Siegel, and David Steere. They are all very talented individuals and it was a pleasure to work with them. Their help in the design, implementation, and usage phases of my work was invaluable. Many of them also read the dissertation and gave useful feedback on it, and I thank them for that extra effort. Special thanks go to Puneet, who never failed to help with a problem or to put a happy face on a discouraging development. I have never known anyone as cheerful or as generous as he.

I thank all of the people who used my software, who persevered through the rough stages of development and who gave me their feedback on what was good and what was bad. My users included all of the members of the Coda group, plus these other folks: Anurag Acharya, Adam Beguelin, Avrim Blum, David Eckhardt, Garth Gibson, Tammy Green, Tom Mitchell, Hugo Patterson, Henry Rowley, Karen Shay, Peter Stout, Manuela Veloso, and Matt Zekauskus.

A number of people on the CMU facilities staff helped me in setting up the Coda testbed environment, in collecting measurements, and in troubleshooting various problems. Special thanks go to Mike Accetta, Paul Parker, Mark Puskar, and Dimitris Varotsis. Bob Baron, Mike Jones, and Mary Thompson of the Mach group also assisted in these respects, and I thank them as well.

I also wish to thank the CMU-SCS faculty and administration for providing such a stimulating – and yet humane – environment in which to conduct research. I know of no other place that meets both of those objectives so well. Special thanks go to Sharon Burks as the cutter of all red-tape and to Catherine Copetas as the junk-food angel of mercy.

A large and wonderful circle of friends made my family's stay in Pittsburgh very special. In addition to the people mentioned already, I'd like to thank Robert and Edie, Harry, Linda and Phil, Claire and Craig, Rhoda and Julian, Geli and Bernd, Dave and Gretchen, Dave and Ann, Deborah, Wendy and Craig, Brian and Debbie, and Wayne and the rest of the "brew crew" for their friendship and for all of the good times.

My parents, Jim and Rita, instilled in me the desire to learn and the value of hard work. Without that grounding and their love I never could have come this far. My pal, Rigsby, took me on countless rejuvenating trips to the park and returned my affection manyfold. I thank the three of them for always being there.

My first daughter, Hannah, was born part way through my thesis research, and she gave me added incentive to finish. More importantly, she made me realize that, although I couldn't write the perfect dissertation, I couldn't be such a bad guy because I'd helped to bring her into the world. My second daughter, Nora, was born after I'd finished, and she continues to help put my work in perspective. My love and thanks go out to both of them.

Finally, my wife, Chris, deserves greater thanks than I can possibly give. For more than a dozen years she has stayed with me, patiently allowing me to pursue my dreams. She didn't complain when I asked her to move from Berkeley to England to continue my studies. Unbelievably, she didn't object when I asked her to leave London for Pittsburgh so that I could enter graduate school. In the course of almost seven years at CMU she never once complained that I was taking too long. During the really grim periods of dissertation

writing, when I doubted everything I'd done and believed I could do no more, she comforted me and gave me the strength to keep going. Without her there for me I surely would have quit. I will always be grateful to her.

James Jay Kistler
Palo Alto, California
August 1995

Table of Contents

List of Figures

List of Tables