Computer Systems
Series

The LOCUS Distributed System Architecture

Gerald J. Popek and
Bruce J. Walker

The MIT Press

The LOCUS Distributed System Architecture

edited by
Gerald Popek and Bruce J. Walker

The MIT Press
Cambridge, Massachusetts
London, England

This book was printed and bound in the United States of America.

Publisher's note: This format is intended to reduce the cost of publishing certain works in book form and to shorten the gap between editorial preparation and final publication. Detailed editing and composition have been avoided by photographing the text of this book directly from the author's typescript or word-processor output.

Library of Congress Cataloging-in-Publication Data
Main entry under title:

The LOCUS distributed system architecture.

(MIT Press series in computer systems)
Bibliography: p.
Includes index.
1. LOCUS (Computer operating system) 2. Computer architecture. 3. Electronic data processing—Distributed processing. I. Popek, Gerald II. Walker, Bruce J.
III. Series.
QA76.76.063L63 1985 005.4'3 85-18157
ISBN 0-262-16102-8

Contents

Figures

Preface

Distributed, interconnected and cooperating computing systems are an increasing reality in many environments, and both technological and social forces promise to accelerate this trend. VLSI and other simplifications are making desktop computers powerful engines and collections of these can have an aggregate compute power equivalent to a very large machine, but at far less cost. Therefore, in those cases where any single computation does not require the very high speed of the large machine, the small ones can be far more economical. It is also much easier in most organizations to justify low cost purchases; thus the result is often a rapid growth of investment in small computers, even without a rational evaluation of the economic advantage. The many advantages of desktop workstations, especially when coupled with larger mainframe machines for work which overflows the desk, are well known.

Unfortunately, it generally becomes necessary to make the resulting collection of computers work together smoothly, a daunting task not present in the single large machine case. This is the distributed computing problem. The LOCUS research effort, begun at UCLA in the late 1970s, attempted to understand how application software and computer networks should interact. The goal was to allow full network access to users who were "mere mortals", without special training. At the time, developing network software was so difficult that most people avoided it. The fundamental concept of *network transparency* grew out of these early efforts, and it was decided to try to build an actual transparent distributed operating system, both to demonstrate the feasibility of this concept, and also to provide a means to further understand its ramifications. By early 1981, a prototype system on a small collection of PDP-11s connected by an early ring network was operational. However, it wasn't until a year and a half later that enough of the associated technical problems and issues were understood and solved well enough that the system, then operational on VAXes, could be made generally available to users other than its original developers.

While many of the early users were enthusiastic in principle, they were far less happy with the non-product quality of the system. It was clear that the Locus approach had great utility but it was beyond the scope of a university research group to provide a finished production product.

Consequently, efforts were made to find a commercial vehicle to produce a product quality derivative of the concepts demonstrated by the research. Early in 1983, that work began at Locus Computing Corporation, an effort which would eventually encompass a number of different machine types, new concepts, largely reimplemented software, and a commitment of manpower and intensity which would dwarf that done earlier at UCLA.

There are several lessons worth learning from this experience. First, distributed functionality should be largely hidden from application programs and users. Providing a convenient distributed computing environment, as represented by a very high degree of transparency, is difficult. But, building software on top of such a base is essentially as easy as building for a single machine. Consequently, it is clear that these facilities ought to be provided in such a manner that maximum use of them may be made. An application should not be required to deal with knowing whether a resource it wishes to interact with is local or not, or knowing specialized means of interacting with remote resources, any more than high level language programmers should be required to write I/O channel programs, or overlays to provide the equivalent of virtual memory. The unnecessary cost and pain is inexcusable. The operating system is thus an appropriate vehicle for support of transparency.

Second, it is exceedingly difficult to reduce the elapsed time between the inception of a research idea and its commercial availability below the typical decade so often quoted. LOCUS has done better, but due only to a combination of vision within a major corporation, and an unusual level of dedication from its original conceivers and developers, many of whom participated in its entire history of maturation.

This book is intended to convey to the reader both the concepts which are key to the distributed system architecture, and "truth" in the form of an actual case study of what is required to support such functionality. The system as described in these pages has been fully operational for some time; ideas, however appealing on the surface, which were not successfully implemented in a product quality way, complete with error handling, integration with other features, and stability under a wide range of conditions are not included in the book.

In closing, it is difficult to convey how many things had to "go right", and how many people's enormously dedicated support was essential in order to develop these concepts and make a resulting system product publically available. The director and research managers at the U.S. Department of Defense research agency ARPA, staff at UCLA, and

the dedicated Locus Computing Corporation team are obvious. Numerous people within IBM and especially the IBM Palo Alto Science Center have invested considerable effort in "making it happen". Heartfelt thanks to all.

Gerald J. Popek
Santa Monica, California
June 1985

Acknowledgments

The LOCUS development was greatly aided by numbers of people, whose contributions have made an enormous difference. First, there are several designers and developers not otherwise mentioned in this book. Charles Kline participated in the original conception of LOCUS on a plane flight from Silicon Valley, and has been a significant intellectual force throughout the system's history. Evelyn Walton has contributed in countless ways, first at UCLA and then at Locus Computing Corporation. A number of students participated in spirited design sessions and built early code as part of the ARPA research effort.

The development team at Locus Computing Corporation has spent enormous energy, intellect, and dedication in both the hard and the never ending smaller tasks. They truly created Locus.

UCLA Computer Science provided a tolerant environment in which the original research work could proceed. Hundreds of students and staff exercised early versions of the system; later, thousands of students provided the half million connect hours that gave a certain measure of reality.

The first large scale LOCUS system at UCLA did not come about without effort. Terry Gray in particular, together with Doris McClure and others, had responsibility for creating and then operating a local area network composed of twenty Vaxes, in the days when Ethernets were objects of curiosity, LOCUS installation was still an art form, and system stability was a goal.

Several major corporations had a substantive positive effect: Xerox Palo Alto Research Center provided a fertile environment for early conceptual development; Digital Equipment Corporation provided a major subsidy for a large Vax network at UCLA. IBM provided the enormous support required to turn a prototype concept into a working system; including functionality extensions, documentation, quality assurance, service tools, and not least, increasing the understanding of what is required to achieve very high product standards.

The MIT Press
Massachusetts Institute of Technology
Cambridge, Massachusetts 02142

The LOCUS Distributed System Architecture

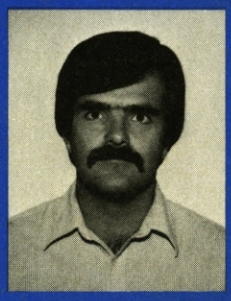
by Gerald J. Popek and Bruce J. Walker

Computer systems consisting of many machines will be the norm within a few years. However, making a collection of machines appear as a single, coherent system — in which the location of files, servers, programs, or users is invisible to users who do not wish to know — is a very difficult problem.

LOCUS, a distinguished version of the popular operating system Unix, provides an excellent solution. It makes a collection of computers, whether they are workstations or mainframes, as easy to use as a single computer by providing a set of supports for the underlying network that is virtually invisible to users and applications programs. This "network transparency" dramatically reduces the cost of developing and maintaining software, and considerably improves the user model of the system. It also permits a variety of system configurations, including diskless workstations, full duplex I/O to large mainframes, transparently shared peripherals, and incremental growth from one workstation to a large network including mainframes with no effect on applications software required to take advantage of the altered configurations.

In addition to transparent, distributed operation, LOCUS features also include high performance and reliability; compatibility with Unix and support for heterogeneous machines and systems as well as database management; and architectural extensions to support bitmap graphics, extensive interprocess communication, internetworking, process snapshots for very high reliability and availability, and a B-tree package for data management.

Gerald J. Popek



Bruce J. Walker

Gerald Popek is Associate Professor of Computer Science at UCLA and President of Locus Computing Corporation in Santa Monica. *The LOCUS Distributed System Architecture* is included in the Computer Systems series, edited by Herb Schwetman.

POPLH
0–262–16102–8