UNIVERSITY OF CALIFORNIA
SANTA CRUZ

STORAGE TANK: A STORAGE AREA NETWORK FILE SYSTEM

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

David A. Pease

December 2008

The Dissertation of David A. Pease
is approved:

Professor Darrell D. E. Long, Chair

Professor Richard P. Hughey

Professor Ethan L. Miller

Lisa C. Sloan
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Storage Tank: A Storage Area Network File System

by

David A. Pease

The aim of Storage Tank, a research project conducted at IBM's Almaden Research Center, was to design and build a first-of-its-kind, SAN-based file system. Some of the important attributes of this file system include direct SAN-based access to data blocks from clients; a true heterogeneous architecture with support for various client platforms; a scalable and highly-reliable cluster of metadata servers, scalability to very large environments; and single system consistency guarantees and locking semantics. A particular focus in designing this system was support of policy-based data lifecycle management, which allows automated control of every aspect of data management throughout a file's useful lifetime.

This dissertation describes the genesis and background of the Storage Tank project, and discusses related work in the area of distributed file systems. It then describes the architecture and operation of the system in some depth. Following those sections is a detailed discussion of the Policy-Based Data Lifecycle Management features of the system. Several companion projects are presented, followed by laboratory and pilot installation experiences with the system. Finally, proposed future work and a discussion of the impact of the system are presented.

Acknowledgments

I must start by acknowledging the long and dedicated effort from many people involved in Storage Tank and its related projects over the years. I begin by thanking the members of the original Storage Tank team at Almaden.

Bob Rees was the person who first suggested that we work together on a follow-on project to ADSM, and much of Storage Tank's high-level architecture was worked out between the two of us before we ever proposed the project to Research management. Bob functioned as the group's Chief Architect, and he wrote much of the server code as well as the first version of the AIX client. Wayne Hineman, our other long-time ADSM team member, was the third person who worked on the project from beginning to end. Wayne, along with Randal Burns (whose own Ph.D. dissertation at UC Santa Cruz was based on Storage Tank work), and Rajagopal Ananthanarayanan (Ananth) worked primarily on server-related components. Bryan Henderson (author of the Linux client), Demyn Plantenberg (author of the Windows client), Atul Goel, Miri Sivan-Zimet, and Ralph Becker-Szendy concentrated on more client and storage related aspects of the system. Cindy Sullivan wrote various internal documents and external papers for us, while Darrell Long consulted on aspects of the system's function and architecture. My manager, Jai Menon, provided technical input, and both he and our lab director, Robert Morris, gave us support along with the freedom to pursue our project goals.

After the initial version of Storage Tank was complete, another group at Almaden did great work testing, debugging, and tuning the code. This team, under

the leadership of Honesty Young and John Palmer, included Sandeep Uttamchandani, Chuck Tribolet, and Bruce Cassidy.

In 2000, Linda Duyanovich, who worked for the Storage Systems Group, formed a team whose charter was to work on bringing Storage Tank to market as an IBM product. In this role, Linda was my partner and product division counterpart. Her group included David Nowlen, John Lindley, Tu-an Cheng, Clem Dickey, Jialin Ju (author of the Solaris client), and eventually about 20 others. Linda's manager later in the project was Bruce Hillsberg, who provided valuable guidance and support to both Linda's group and my own.

Leo Luan led the Distributed Storage Tank research, with Manuel Pereira, Ted Anderson, and Jeff Riegel working on the project. Demyn Plantenberg and Miri Sivan-Zimet worked on the Hydra project, and built upon work done by Marc Eshel and Manoj Naik. Andrew Tridgell led the research on Distributed Samba.

The work on Object-Based Storage Devices involved many people. In Haifa, my initial contact was with Julian Satran. Eventually the project there expanded to include Dalit Naor, Michael Factor, Ohad Rodeh, Ealan Henis, and Ami Tavory. Miri Sivan-Zimet worked with the Haifa team on the first integration of their technology into Storage Tank. The Storage Tank-related part of the Kubos project was done by Richard Golding, Ralph Becker-Szendy, Ted Wong, Omer Zaki, and Jody Glider.

The STEPS team at Watson Research was led by Murthy Devarakonda and included Mandis Beigi, Rohit Jain, Marc Kaplan, Jim Rubas, Upendra Sharma, and Akshat Verma.

Besides members of my own group, several other people were crucial to the success of our pilot program at CERN. These include Robert Haas of the Zurich Research Lab, who spent a great deal of time in Zurich "babysitting" the system, Paul Bradshaw, who took over the day-to-day management of the Almaden-CERN relationship from me, Prasenjit Sarkar, who worked on the iSCSI microcode, Chuck Tribolet, who set up a CERN-like test environment at Almaden and who did a great deal of the problem determination on the system, and Juan Gomez who became an expert on Storage Tank installation and helped CERN an many others set up working systems.

My primary collaborator on tape Storage Pools in Storage Tank was Jürgen Deicke in Mainz. Members of the eRMM team there included Wolfgang Müller-Friedt, Christian Bolik, and Bernd Blaudow.

Finally (on the Storage Tank front), several people in Poughkeepsie were involved in integrating fibre-channel storage and Storage Tank into System z. They included Steven Greenspan, Al Merritt, Harry Yudenfriend, and Steve Henkels.

I owe particular thanks to the members of my Ph.D. committee at UC Santa Cruz. Darrell Long has been my advisor through many years of part-time Masters and Ph.D. study and has always been a source of encouragement and guidance. Richard Hughey was my department chair, Master's Thesis advisor, and a person I often relied on for advice and direction. Both Darrell and Richard were instrumental in my admission into the UCSC graduate program, for which I will always be grateful. My thanks to Ethan Miller, who gave me both encouragement and a great deal of valuable advice. Thomas Schwarz, who through an unfortunate intersection of timing and location does

not appear on my title page, participated in my every other aspect of my dissertation process. I appreciate his feedback as well as the effort he went through to be involved while remote.

At UCSC, I also wish to particularly thank Pat Mantey for his support, and Carol Mullane for keeping me on track and for helping me out more often than she should have had to.

I would be seriously remiss if I did not recognize the great latitude that several IBM managers have allowed me in pursuing my graduate career while being employed full-time at IBM. These managers include Jai Menon, Linda Duyanovich, and Bruce Hillsberg.

I save my last and great thanks for my wife Susan, who put up with a decade of part-time graduate school, numerous vacations interrupted by studying, and almost a year of my "non-work" time spent working on this dissertation. Thank you!

Chapter 1

Introduction

Storage Tank was a research project conceived and primarily conducted at IBM's Almaden Research Center. The project ran from 1997 through 2004, although in its later years its focus was more on producing an IBM file system product than on research. (The first product version of SAN File System shipped in 2003.) By the time that the project was finished, it had involved hundreds of people across multiple IBM divisions in several countries. (So the claim that the project was conducted primarily at Almaden, or even in Research, is only true for the original ideas and the core file system code, not for some of the later product-oriented work nor for the many ancillary projects that the project spawned.)

Storage Tank's goal was to build a file system that was different from what existed at the time in many important ways. Some of the ways in which Storage Tank is different include:

It is a SAN-based, heterogeneous, distributed file system:

- Data is shared and accessed directly by client systems across a Storage-Area Network
- Heterogeneous client support is integral to the design of the system Centralized control of the system is provided by a highly-available metadata server cluster
- A single name space is provided to all clients, with single-system file sharing semantics and consistency guarantees

  It includes support for Policy-based Data Lifecycle Management
- The name space is decoupled from the storage, and files can be stored where most appropriate (and moved, if necessary) regardless of their location in the name space
- Files can be placed on any of various classes of storage (storage pools) based on administrator-defined policies
- Activities like backup, migration (HSM), and even data expiration can all be controlled by local data lifecycle policies

It provides integrated capabilities for common data center operations without impacting system availability:

- Storage is virtualized into storage pools for flexible management and growth
- Data can be automatically moved off a volume (device) or group of volumes, even while in use

- Individual volumes and entire storage pools can be removed from the system without interrupting access to data
- New devices can be added to the system, either to grow existing storage pools or to create new pools, without impact to users or applications

However, the ideas and goals of the Storage Tank project did not arise spontaneously. Somewhat surprisingly given the amount of work that was going on around file systems in IBM Research at the time, neither did they evolve directly from any existing file system projects. (Section 2 discusses other IBM file system work of the time.) Instead, to a large extent they grew out of earlier work done on a sophisticated backup system called ADSM. Many of the ideas and technologies that went into Storage Tank were first explored by the team while working on ADSM; some of these include heterogeneous clients, SAN-based data transfer operations, and policy-based management. It is, therefore, difficult to properly introduce the Storage Tank project without a description of that earlier work.

1.1 In The Beginning

In the early 1990s a group of researchers and engineers at Almaden Research Center built a system called the Adstar Distributed Storage Manager (ADSM) [17]. ADSM (now called Tivoli Storage Manager, or TSM) is a client/server-based backup and archive system. It uses native software clients that run on heterogeneous systems (everything from PC-DOS to Cray and anything in between) to perform the local part

of the backup or archive process (identify files to send, transmit them to the server, etc.), and a server that can run on an almost equally broad range of systems to collect and store the backup and archive data.

From the outset ADSM had capabilities that are still uncommon in other backup/archive solutions. One is the ability to define policies that can be applied to files as they are backed up; policies can specify parameters such as how often a file is to be backed up, how many copies (generations) of a file are to be kept, and the type of storage (Storage Pool) on which the file is to be stored [47].

In addition, server-level policies can define how data is migrated from one Storage Pool to another over time. Storage Pools can be made up of almost any type of storage media (disk, tape, optical), but each pool must be homogeneous.

The ADSM server uses a proprietary database with full transactional semantics and an ARIES-based [62] write-ahead log and recovery scheme. Among other things, the database keeps a complete set of metadata about all of the files in the backup and archive repositories. Because this metadata is often of interest to installations using ADSM/TSM, the database supports an SQL92-compliant query interface [1], including access through ODBC/JDBC, for querying and mining the metadata.

In the mid 1990s, IBM developed an early storage-area network (SAN) technology for open systems called Serial Storage Architecture (SSA) [46]. SSA allowed multiple disk drives to be directly attached to multiple hosts, a novel capability in open systems at that time. (A discussion of the rush to develop a competing SAN standard based on Fibre Channel, and the ensuing SSA versus FC SAN wars is beyond the scope

of this paper, though it makes interesting reading [27].)

The ADSM team at Almaden recognized the potential for solving some of the growing backup network and server bottleneck problems by using a SAN (though that term was not yet in common use). They developed two extensions to the ADSM client/server architecture called "LAN-free Backup" and "Server-Free Backup." One of these schemes allows a SAN-attached ADSM/TSM client to write backup data directly to server disk Storage Pools (which, of course, must be on the SAN); the other uses the SCSI-3 extended copy command and relies on a 3rd-party data mover to copy data from client disks to server Storage Pool disks without any client or server overhead. These features were among the earliest commercial exploitations of SAN technology.

1.2 Towards a New File System

The original ADSM group at Almaden Research meanwhile observed that the amount of data that was being stored on open systems computers was growing much more quickly than they had expected when the ADSM project began. This was due, at least in part, to the growing size of relatively inexpensive disk drives, and the exploding number of applications that were being implemented on open systems. Relatively new technologies such as the World-Wide Web and rich media were also contributing to the ever-growing amount of data being stored on these systems.

The group saw two major trends emerging as a result of this explosion of data. One was the increasing need to share the data and to share it across heterogeneous

systems. The other was the growing difficulty and expense of trying to manage the data. Most data still existed on islands of isolated systems, and so-called "sneakernet" was a prevalent form of data sharing. Companies could not reliably identify the location of data that was important to their business, or control the often inconsistent copies of that data stored on various systems. (Sadly, this situation has not really improved much in the intervening decade.)

The team felt that capabilities they had developed for ADSM, such as heterogeneous client support, efficient SAN-based data access, and policy-based management of data, could be adapted to address some of the challenges created by the rapidly growing amount of active file system data. The SAN could provide a vehicle for direct data access and data sharing between heterogeneous clients while allowing a metadata server to control data placement, access, and lifecycle management.

An early idea was to modify the ADSM server to provide these services to clients, perhaps through a special protocol (rather than through standard file system calls). As the nascent project progressed, however, two things became apparent. The first was that retrofitting the existing ADSM server, especially making servers operate in a scalable, fault-tolerant cluster, was likely to be more work than starting over. (Such fault-tolerance was judged to be critical for live data.) The second was that providing data access through the native file system interfaces on the various client systems would make the system much more useful, and provide an immediate base of applications that could take advantage of it.

Thus, the idea of building a SAN-based, heterogeneous, policy-managed file

system was born. Even at this point the group shied away from committing to such a large undertaking, and looked around for an existing file system that they could use as a starting point. An obvious option was to use the relatively new General Parallel File System (GPFS) [80] that was being developed in the same lab. However, at the time GPFS had some attributes that made it a poor fit for the project, including a cluster (rather than client/server) architecture, a reliance on the virtual shared disk (VSD) architecture of the SP2, and a very AIX-centric design. (Many of these constraints have been eased in the ensuing years.)

Other possibilities were the AFS [45] or DFS [55] systems that IBM had recently acquired with its purchase of Transarc. These systems already had many of the attributes the group desired, including a single namespace, heterogeneous clients, and scalable servers. At the time, however, AFS development had stopped and attention was focused on DFS. DFS might have been a good starting place, but its reliance on the DCE infrastructure and its perceived poor performance eventually caused it to be bypassed.

In the end, the group decided that a totally new code base was the only alternative that would allow them to cleanly implement all of the features they proposed for the new system. Their list of features included the following:

- A distributed file system with a client/metadata server architecture
- Highly-available, clustered metadata servers with shared metadata access
- A single file system namespace

- Heterogeneous client platforms with transparent user and application access (no required changes to existing applications and a native file system personality)
- Direct data access from clients, with high-speed, secure access over the SAN
- Single-system semantics for file locking and sharing
- Optimization for high performance over a wide variety of file types and sizes
- Scalability to petabytes of data, billions of files, and thousands of clients
- Integrated policy-based data lifecycle management
- Centralized access control
- Support for common operational activities like adding and removing disk volumes

This was an ambitious list, to be sure (and one that predictably took the group much longer to implement than was originally expected). In late 1996 and early 1997, the group added new members, set down their goals, and turned their attention to building this file system.

As the project began, much discussion took place around what the primary development platform should be. One early team member was in favor of developing on Windows NT (and proposed calling the project WolfTracks, since it would use Microsoft's WolfPack clustering technology); however, the other team members did not feel that Windows was a good fit. Unix seemed a better choice, and from an IBM point of view the obvious candidate was AIX, a proven Unix platform. However, the

group would need many systems for development, and AIX systems of the time were very expensive. This author had been using early versions of Linux on his personal PCs since 1994, and was convinced that the servers could be developed on a Linux base (and ported back to AIX if necessary). Thus, Storage Tank became the first large project undertaken using the Linux operating system at IBM.

In the early days of the project, the decision to use what one Research executive referred to as "a toy operating system built by hackers on the net" had to be defended at the highest levels of the Research Division. Several years into the project IBM officially embraced Linux, and all such concerns were forgotten.

Another early question was what SAN technology the group should start with. At the time, IBM's SSA seemed the logical choice. In fact, diagrams of the Storage Tank architecture (like Figure 3.1 at the beginning of Section 3) are almost always drawn with the SAN represented by an oval (rather than the more common cloud network). This is because the initial architecture drawings were done with the assumption of an SSA network, which had a loop architecture and was usually drawn as a circle or oval. In time, FibreChannel superceded SSA, and was used instead.

A final seemingly important issue was what to call the project (once the idea of WolfTracks was abandoned). At the time the project began, the founders of the project both owned rural mountain properties with water storage tanks on them. The image of these tanks holding a large amount of water and distributing it as needed gave the project its name.

1.3 The Author's Contributions

The author of this dissertation was one of the two originators of the Storage Tank project, and was its manager from its inception through the time Storage Tank shipped as an IBM product. During this time, he made many, varied contributions to the project.

The two people who conceived of the project that became Storage Tank were responsible for setting the project's initial goals as well establishing much of its basic design. Because the goals and design were worked out between the two of them over a period of months, it is impossible to attribute particular ideas to specific people.

The ideas and approaches that the project's originators proposed included: building a shared file system with direct data access over a Storage Area Network; implementing a truly heterogeneous architecture that properly supported both the metadata needs and the semantics of differing client platforms; using a client-server architecture with lightweight, non-clustered clients and a highly-available, clustered metadata server; and providing policy-based management capabilities for data stored in the system.

The author directed or participated in the design of many aspects of the basic Storage Tank file system, although by agreement the technical leadership of that team was left to its Chief Architect, Bob Rees (who was the project's co-founder). In addition, the author was directly involved in the establishment, research, and technical leadership of many of the auxiliary parts of the project.

The author led the research in all aspects of Policy-Based Data Lifecycle Man-

agement. This included specifying most of the capabilities of the basic policy-based management of the system, as well as the language that it would use. It also included directing the research into a scalable architecture for implementing asynchronous data management policies that became the STEPS project, again directing both the goals that the project would pursue and how they would be be accomplished. Finally, the author recognized the need for application-aware policies, and developed an approach for providing them in the framework of the existing Storage Tank policy infrastructure.

Other projects on which the author led the research included Tape Storage Pools and the use of eRMM, and much of work in integrating Storage Tank into the IBM mainframe. He was also traveled to the Haifa Research Lab to present the first proposal for building an Object-Based Storage Device and remained very involved in that research for most of its duration.

In a less research-oriented capacity, it has already been pointed out that he managed the group for seven years. During this time time he was the primary person who "sold" the project, first to his management, then to Research management, and later to other parts of IBM. He was involved in starting many of IBM's Storage Tank-associated research projects, and was the spokesperson for Storage Tank both inside the company and with potential customers, various government agencies, industry analysts, and the press. He was involved in the first visit to CERN, and was instrumental in establishing that relationship as well as setting direction for the pilot program. He was also responsible for the choice of Linux as the project's primary development platform.

Chapter 2

Related Work

2.1 Introduction

Due to the length and scope of the Storage Tank project, the related work section is divided into several parts. The first part covers distributed file system work that predated the project, or that was current when the project was in its early stages and its goals and directions were being set. The second covers some of the work done after Storage Tank was well underway and publicized. The final part discusses related work that is not specifically distributed file system-oriented, such as work on policy-based management and information lifecycle management.

Related work that is considered part of the larger Storage Tank research project and is covered in other sections, such as IBM's work on Object-Based Storage Devices or secure file systems, is not repeated here.