Digital Crime Scene Investigation for Zettabyte File System

Andrew Li (31558046)
andrew.li@students.mq.edu.au

Supervisor: Prof. Josef Pieprzky

Expected Outcome

- File system basics
- Digital forensics on file system
- ZFS basics
- Digital forensics on ZFS with our new ZDB

Topic Outline

- Introduction
- File system in Unix
- ZFS overview
- New ZDB feature

INTRODUCTION

File System Basics

- Computer stores data on disk
- File system is the layer above the disk
- File system is like a filing cabinet

Digital Forensics

- Digital crime scene investigation is sometimes referred to as Digital Forensics
- Forensic science on computer
- Finding legal evidence
- Identifying how a computer arrives as its current state

Difficulties in file system forensics

- File system cannot be modified
- Every detail counts
- Access time, change time, owner, file content
- No mistakes allowed

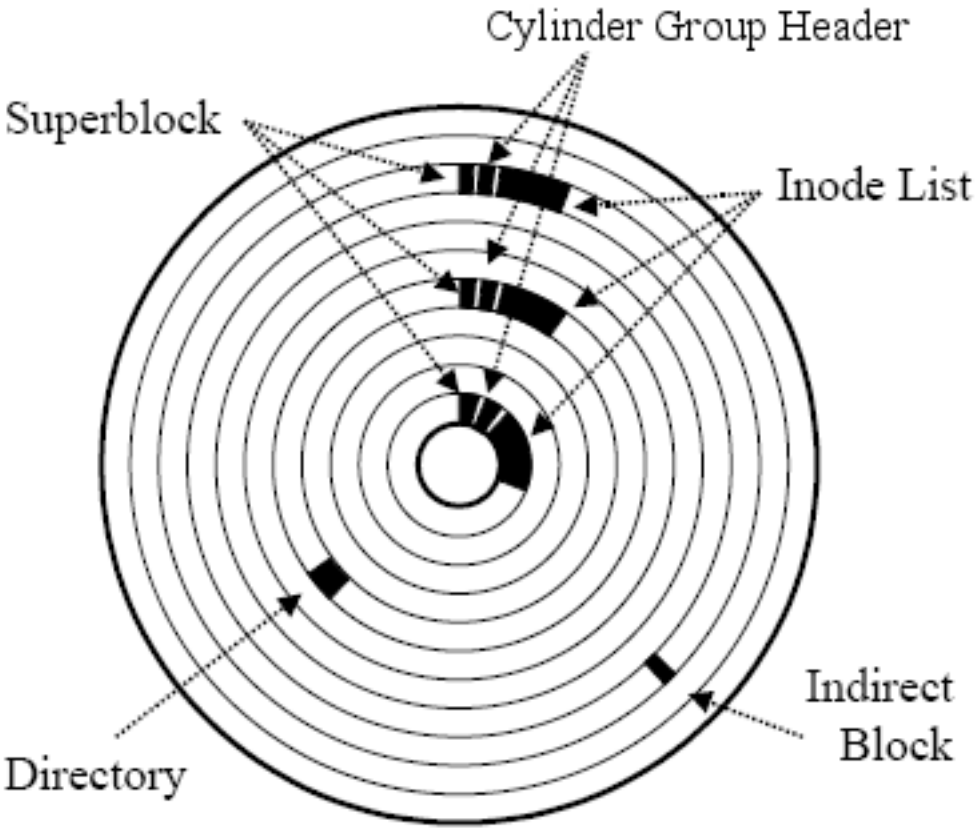
ZFS

- Zettabyte File System (ZFS)
- New file system from Sun Microsystems
- ZDB is a file system debugger for ZFS
- 1 Zettabyte = 1billion terabytes

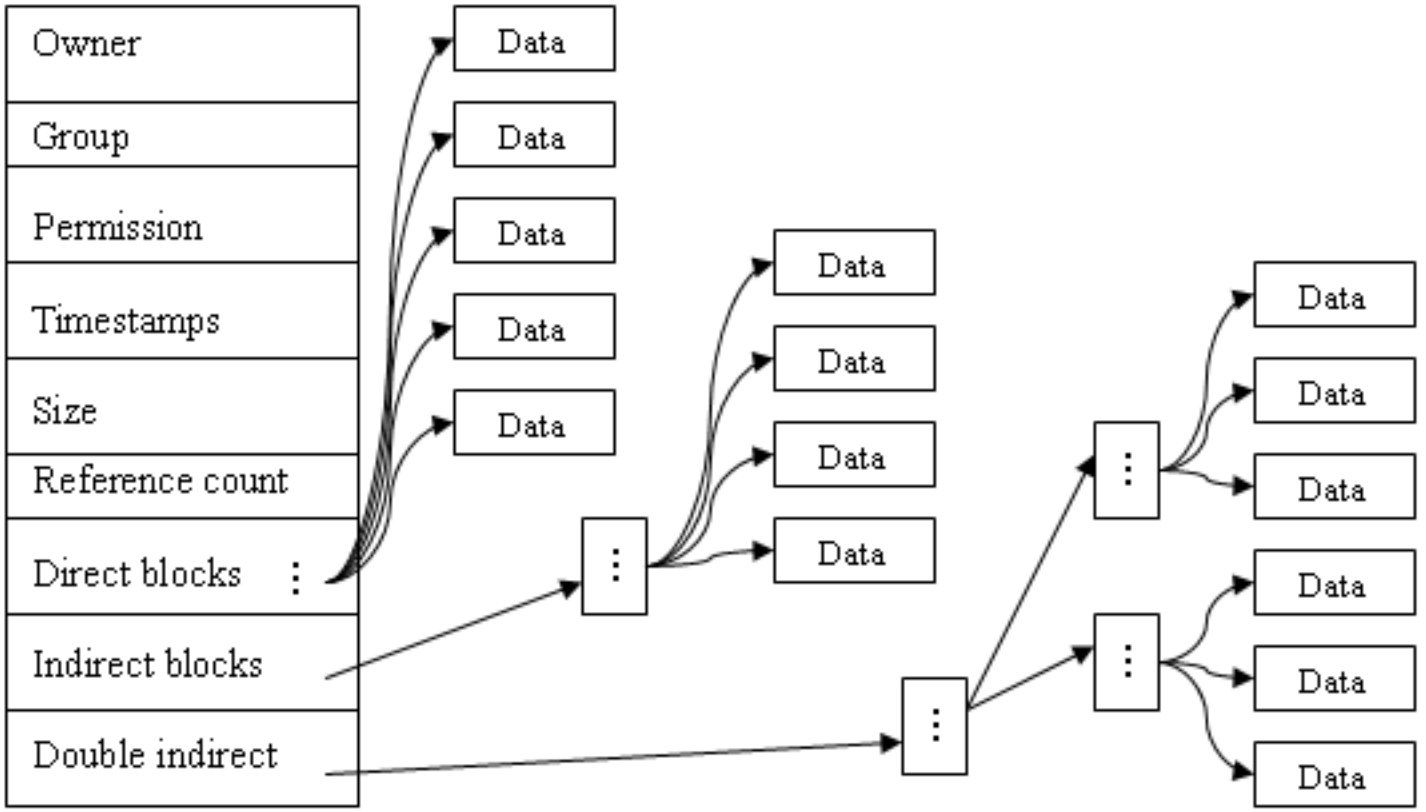
FILE SYSTEMS IN UNIX

File System Basics

- Partition
- Cylinder Group
- Superblock
- Inodes



Inode


Owner
Group
Permission
Timestamps
Size
Reference count
Direct blocks
Indirect blocks
Double indirect
Data
Data
Data
Data
Data
Data
Data
Data
Data
Data
Data
Data
Data
Data

How do operating system store files?

- Unix Sixth Edition from 1977

```
5507: struct  file
5508: {
5509:       char    f_flag;
5510:       char    f_count;        /* reference count */
5511:       int     f_inode;        /* pointer to inode structure */
5512:       char    *f_offset[2];   /* read/write character pointer */
5513: } file[NFILE];
```

- Modern Unix have more complex `struct file` but the idea is the same

ZFS

Do we need this project?

- OpenSolaris Forensic Toolset proposal
  http://www.opensolaris.org/jive/thread.jspa?threadID=45379
- The scope is much larger than ours
- It includes:
  - Live monitoring
  - Live system dissection tool with Modular debugger and Dtrace
  - Malware detection tool
  - Forensics bootable DVD
  - Solaris fingerprint database

Deleting a file in ZFS

```
unlink() -> vn_remove() -> vn_removeat() ->
VOP_REMOVE() -> zfs_remove() ->
zfs_link_delete() -> zfs_znode_delete() ->
zfs_znode_free() -> zfs_log_remove()
```

- Can also verify by

```
dd if=/dev/dsk/c0t0d0s0 | strings |
  grep illegal_data
```

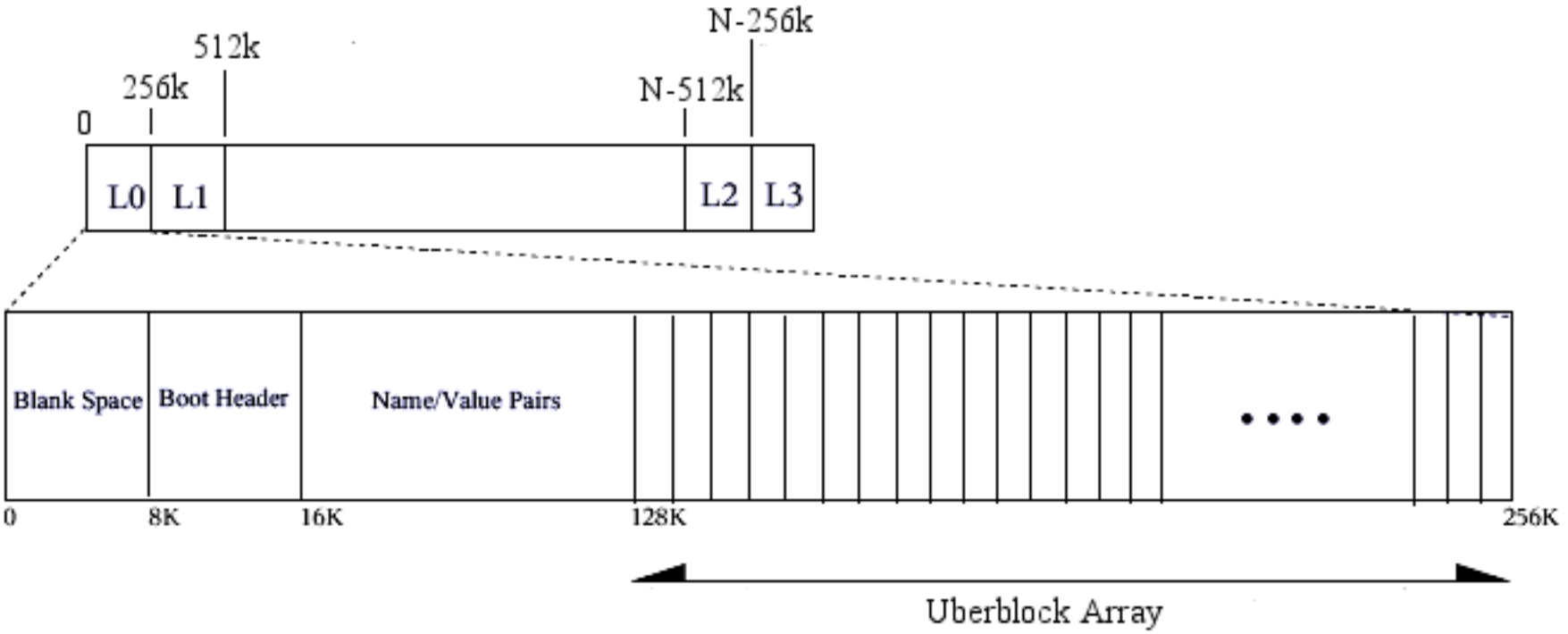
Design and Concept

- No more re-partitioning
- Guarantee file system consistency
- Self-healing

ZFS Basics

- Virtual Device (vdev)
- Vdev label
- Uberblock
- Block pointer containing a Data Virtual Address pointing to data blocks on disk
- Every object is described by a dnode

Virtual Device and Label



NEW ZDB FEATURE

Our new ZDB extension

- Retrieve the active ZFS uberblock and block pointer
- Retrieve the dnode for the metadata object set
- Retrieve the Object Directory dnode and its ZAP object
- Retrieve the DSL Directory object
- Retrieve the DSL Dataset object
- Using the DSL Dataset dnode, retrieve the ZFS file system object set
- Using the ZFS file system object, get the Master dnode and its ZAP object
- From the ZAP object of the Master dnode, get the root directory dnode of the ZFS file system
- From the block pointer of the root directory, find the object id of our target file
- Using the address stored in the object id dnode, retrieve the block of data directly from the disk and output the raw data.

Summary

- In summary, our new ZDB travels through layers of the ZFS file system to search for the target data stored on disk.
- ZDB grabs a chuck of data directly from the disk without invoking the file system of the operating system

Future work

- Add option to trace whole uberblock array
- Turn code into library function
- Clean up and submit