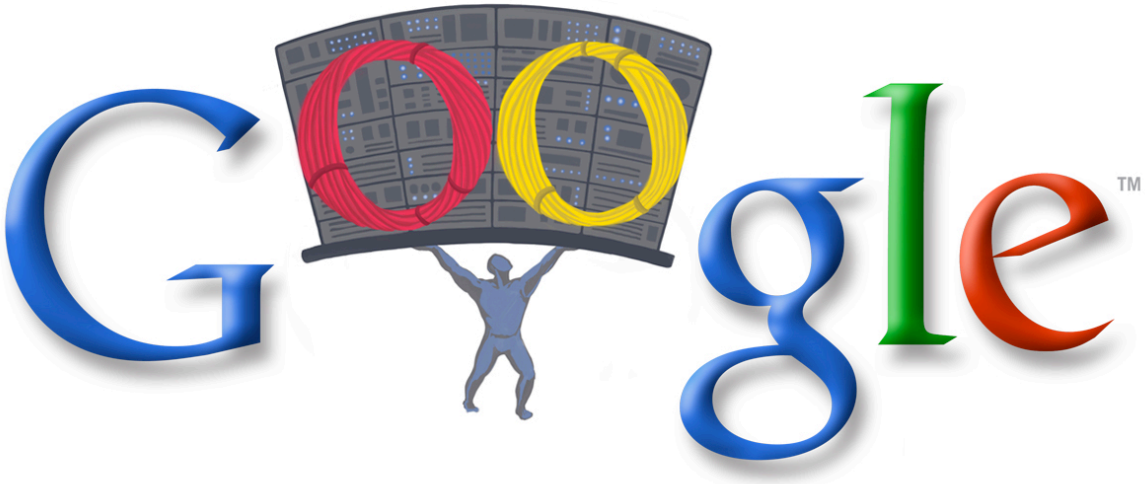


Local File Systems in the Cloud

Michael Rubin

Clouds & File Systems

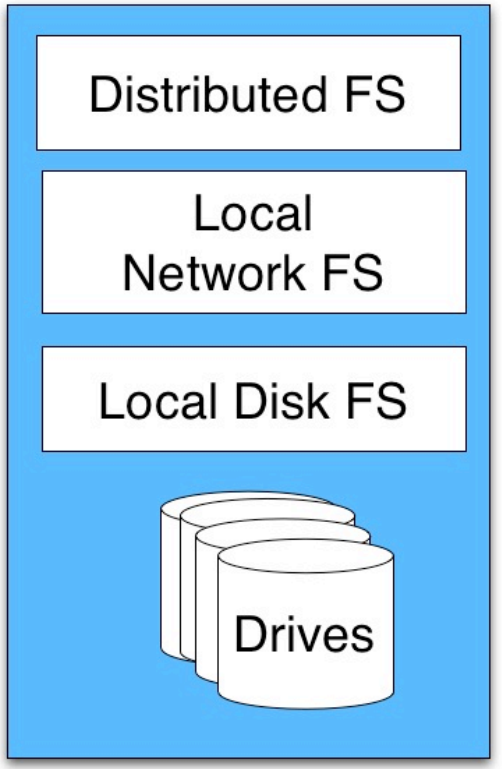
Google

Clouds
- Many machines managed by others
- Trusted with important information

Cloud storage:
- Managed by SW stack
- Local file system is Linux (Free, Open)

Clouds studied
Local file systems studied

Today it's local file systems in the cloud



Clouds are Different

Environments

Google

Environment	Systems	Workload	Storage Traffic
Desktop	1	Varied	Low & Bursty
Enterprise	100s	Focused	High & Varied
Cloud	10,000s	Varied	High & Constant

Storage Differences

Google

Environment	Storage HW	Support Model	Reliability	RAM for Storage
Desktop	Single Drive Single SSD	One to many	Luck	Lots
Enterprise	High End	Team to 100s	High End HW	Lots
Cloud	Cheap Commodity	Team to 10Ks	Replication	Limited (Shared Clouds)

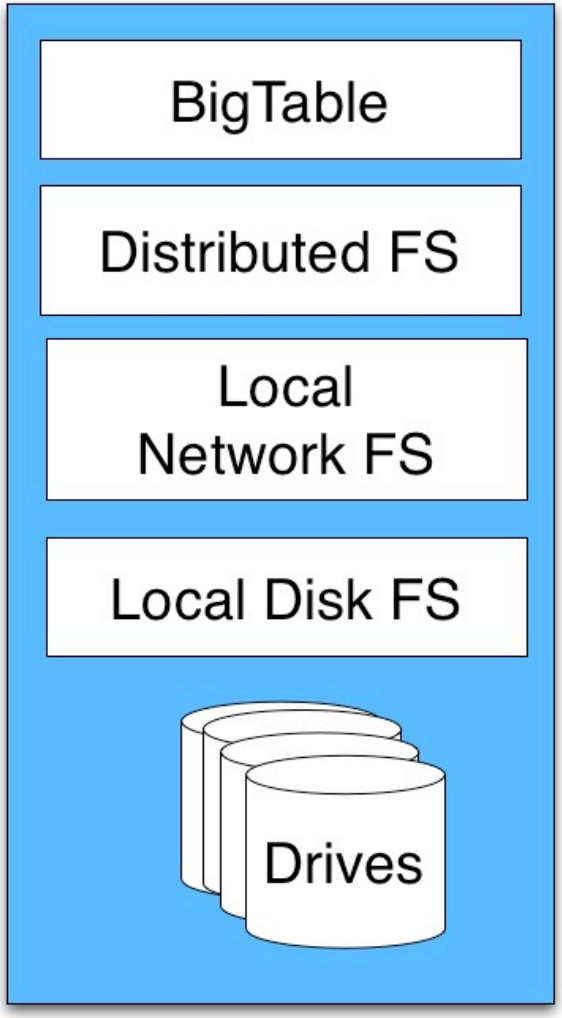
Cloud Resource Costs

Google

- SW customized for performance & management
- Multiply cost across entire cloud
- Require that local file systems operate within limited resources

Local File Systems Matter

Google Storage Stack



1,000s

10,000s

Google

ext2 used since 1996

Why still ext2?

- No measured benefit or detriment
- Do not risk user data
- Changes a fundamental assumption
- You'd have to be crazy

Just Crazy Enough

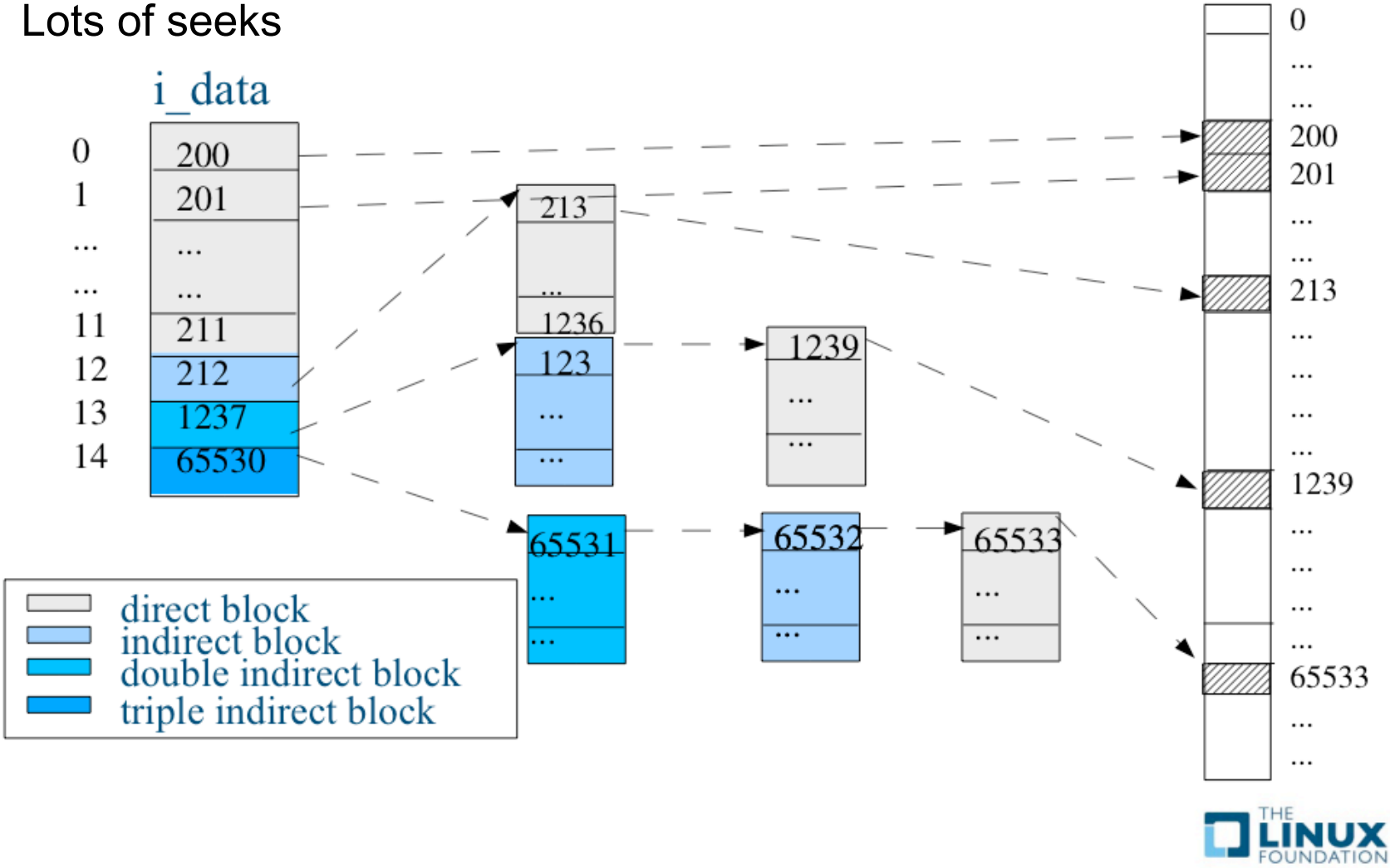
In 2008 Chad Talbott and Michael Rubin started measuring behavior of ext2 in the cloud

Evaluating ext2: Indirect Blocks

Google

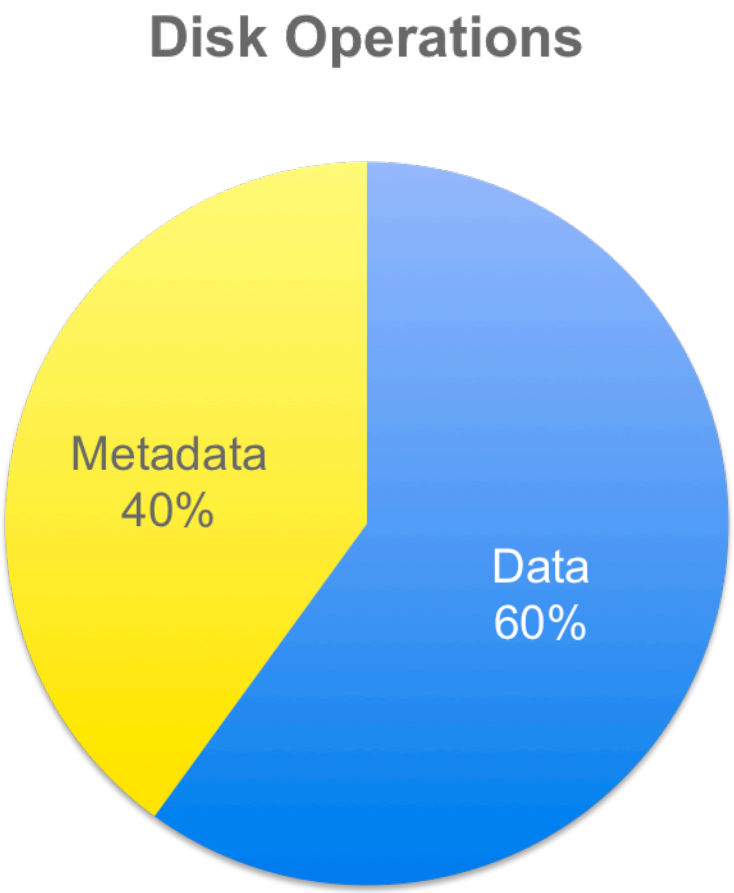
Lots of metadata as files get larger

Lots of seeks



THE LINUX FOUNDATION

Evaluating ext2: Metadata vs. Data



Evaluating ext2: Longer than 20ms

Google

Expecting about 20ms for drive operations

- Removing 8MB files could take 800 seconds under heavy load
- More than just metadata ratio problem

Fragmentation

- Disk < 90% full
- Allocation challenges
- 4KB reads take > 100ms due to 3-8 accesses
- Workaround to read all metadata in cache

Evaluating ext2: Cloud Threats

Google

Event Resilience (Power Fail, Panic)

- Over 90% errors fixed with fsck
- Unavailable for 85 min/TB

Traffic Resilience

- Even with replication data loss has adverse effects
- ext2 rarely lost data
- Linux support great

Selecting a File System

Google

Active Linux Support and Users Required

Eliminated: reiserfs, jfs, homemade fs

Candidate	Pros	Cons
btrfs	Great Features	Not ready yet
ext3	Journaling, Simple	ext2-performance
ext4	Performance, Simple, Journaling, Easy Upgrade	Not highest performer
xfs	Performance	Complex
zfs	Full featured, Reliable, Performance	License Complexity

We chose…

Google

2½ years

Google

- Commitment to not lose data nor disturb Google
- File systems upgrades are one way
- Power fail testing
- Reliability validation
- Upgrade configurations
- 10s then 100s then 1,000s then 10,000s

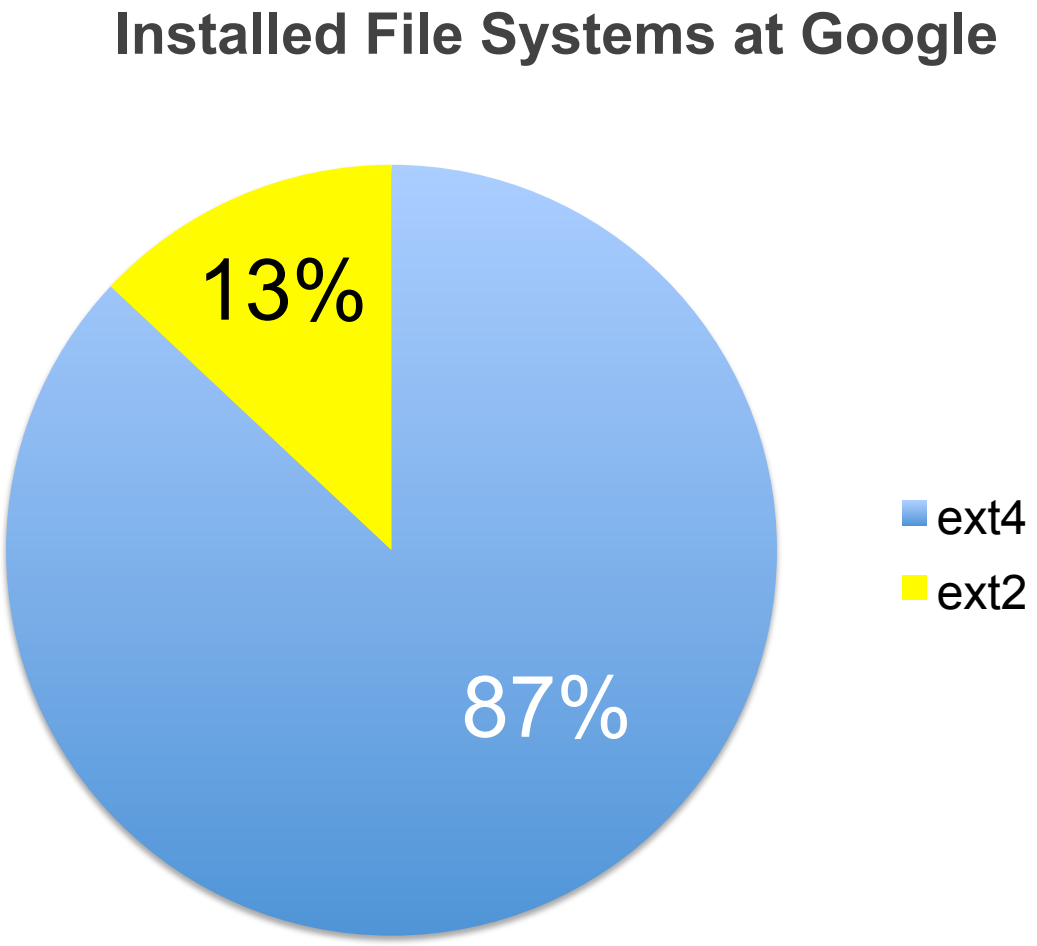
Next time will take closer to 12 months

2011 Upgrade Complete

No data loss to our knowledge

Why still 87%?

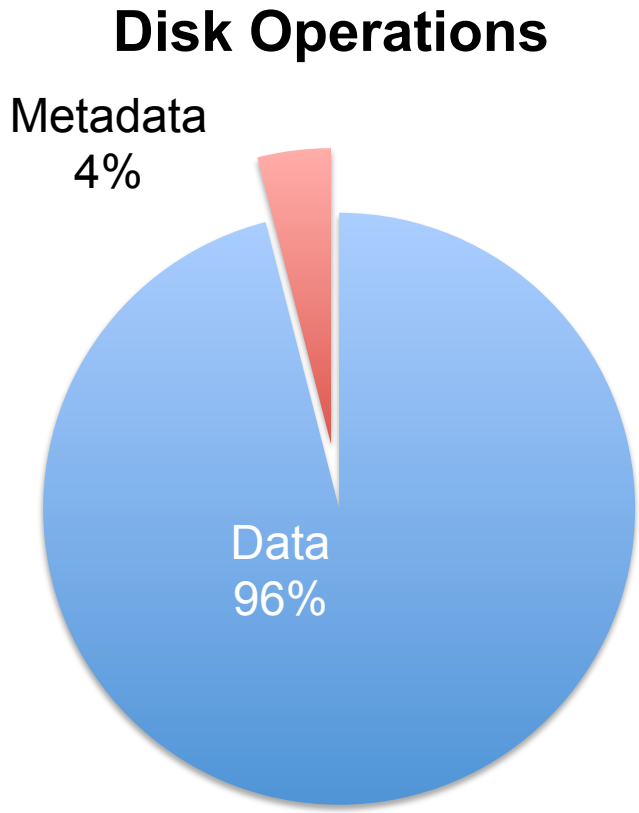
- Gmail rolls out slowly
- ext2 specific flags



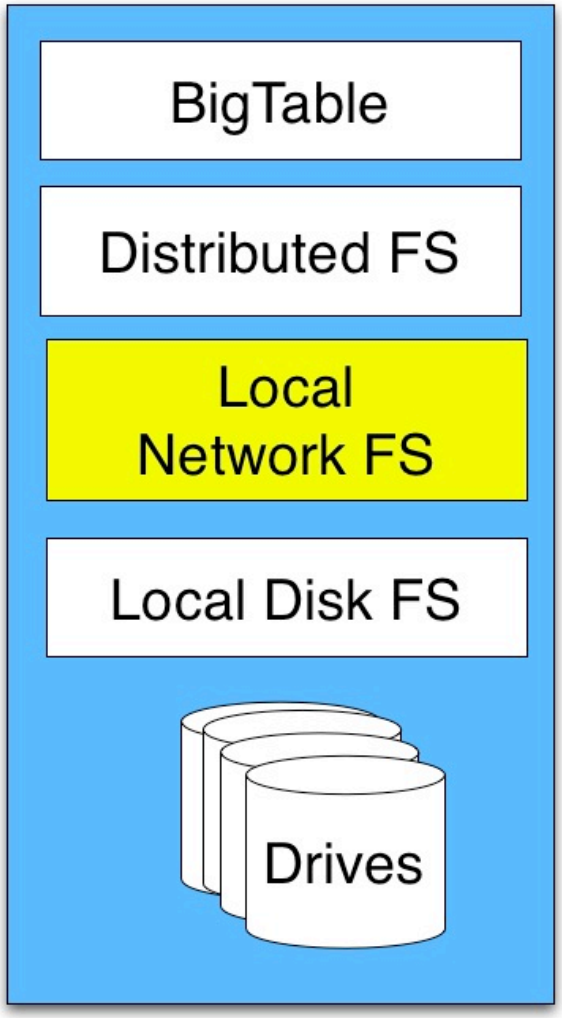
ext4: Metadata vs. Data

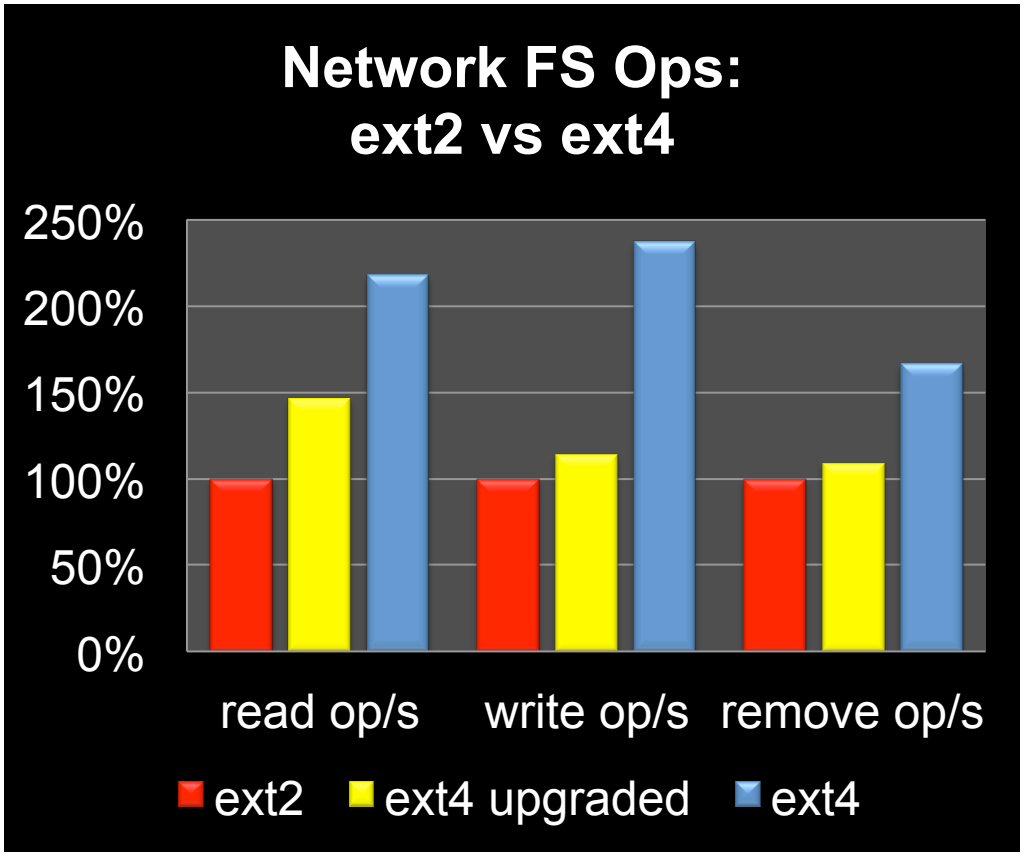
Same workload

Other workloads show 10%-20% of metadata

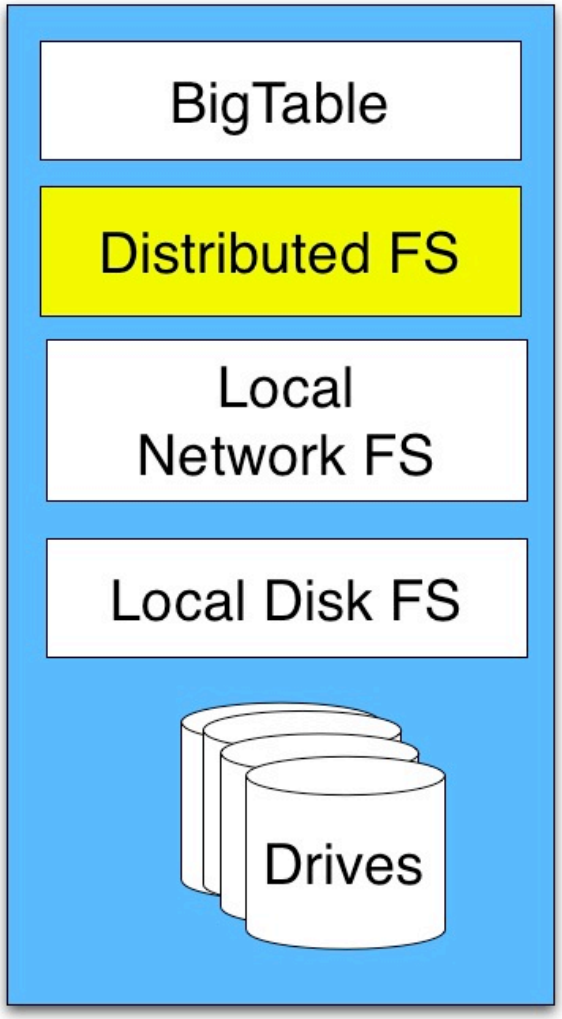


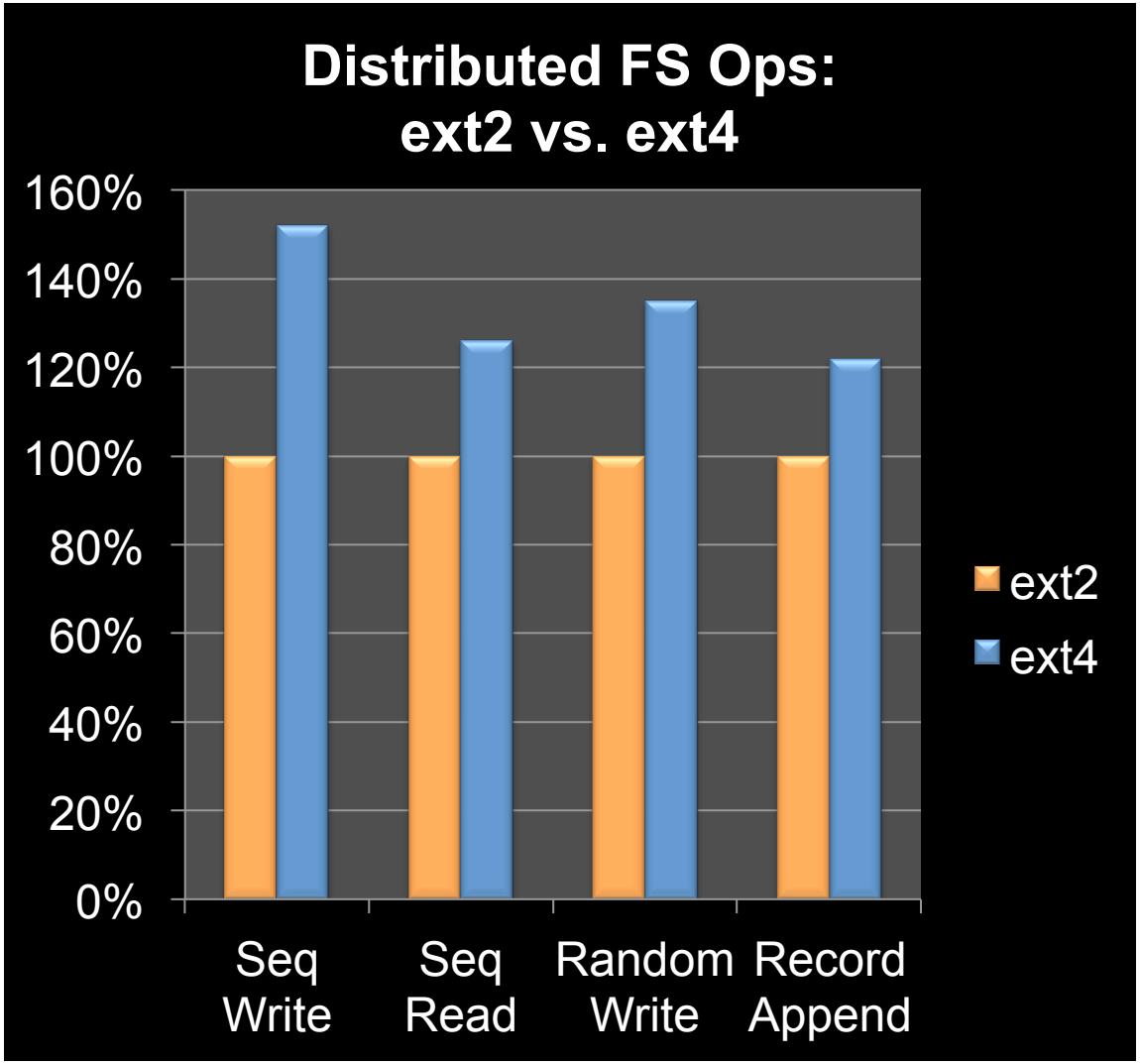
Impact on Network File Server



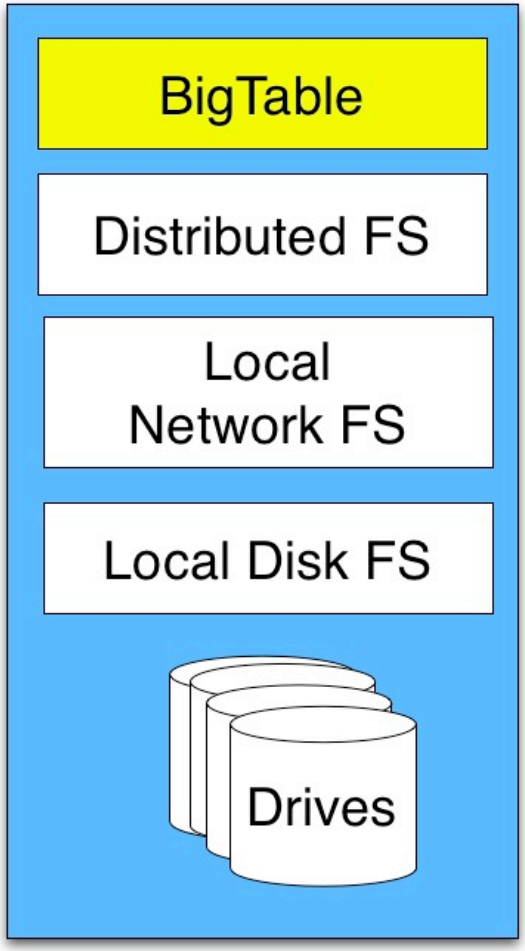


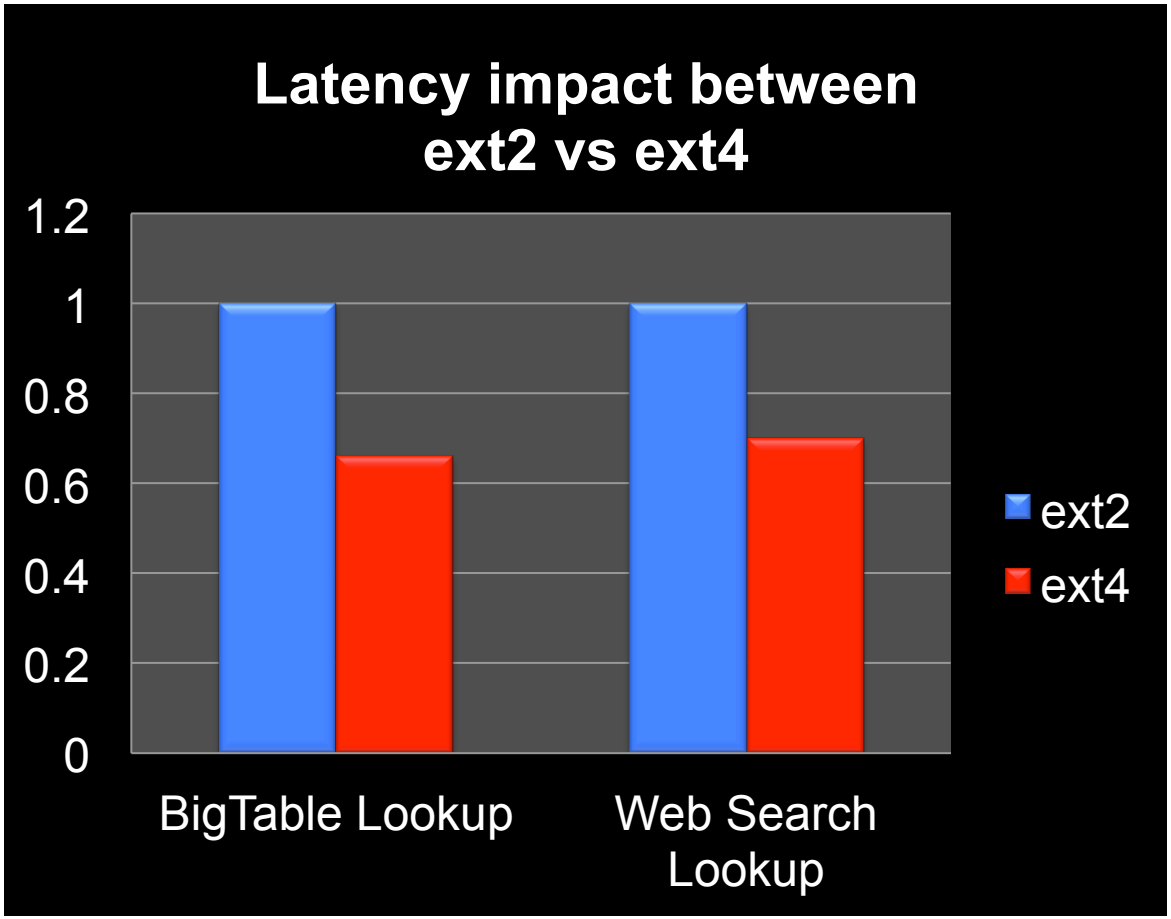
Impact on Distributed File System





BigTable & Web Search Impact





Don’t forget that with latency lower is better

Requirements for the Cloud

Visibility Requirement

Google

1 person responsible for 10K machines

What is going on?

- Is buffered data stored on disk?
- Where is the slow down? VFS? FS? Block layer?
- Which application is hogging the spindle?
- What is the metadata to data ratio?
- Which disk/app/user is slowing me down?

Visibility Requirement: Example

Google

Deployed ext4 after early tests to tens of thousands

- Happy for a few weeks
- Did not know we were corrupting data, other systems knew
- Linux was telling us in dmesg
- Found out after problem became unwieldy
- Consistent upgrades compounds problem
  - Which version corrupted data? Which made it worse?
  - How severe is the problem?
  - Do we need to reformat to fix?

Predictable Requirement

Google

Understand Performance vs. Cost

Example:

- Under 20 ms when serving data with 5GB of RAM for storage
- 500MB provisioned yields 500ms-2sec latencies
- Latency spike only happens sometimes
- Stragglers hold others back
- Metadata paging was the source of this one
- Solvable

HW Abstraction Requirement

Google

This topic is a talk on its own

HW is changing rapidly and fundamentally in storage

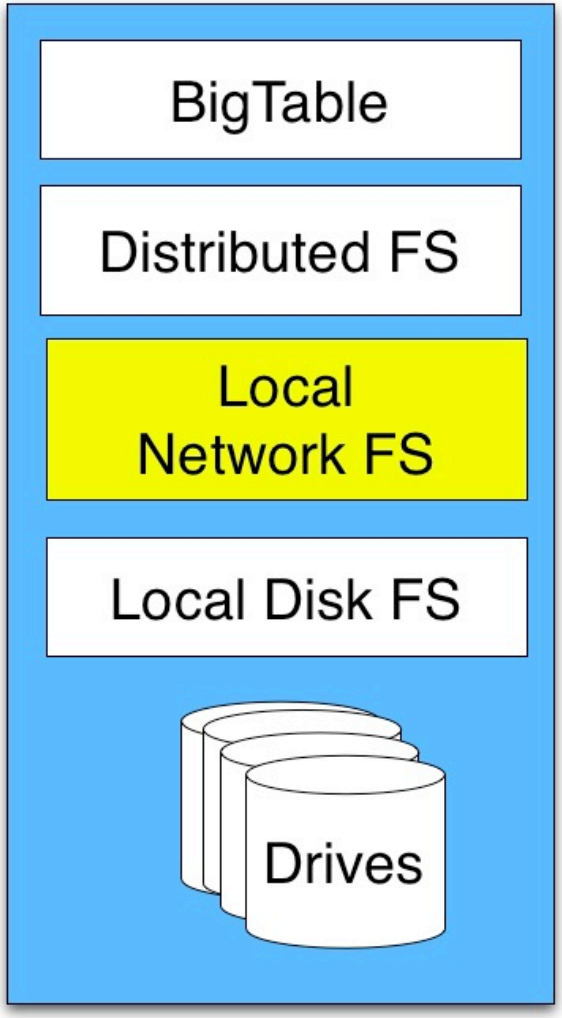
Magnetic Platters

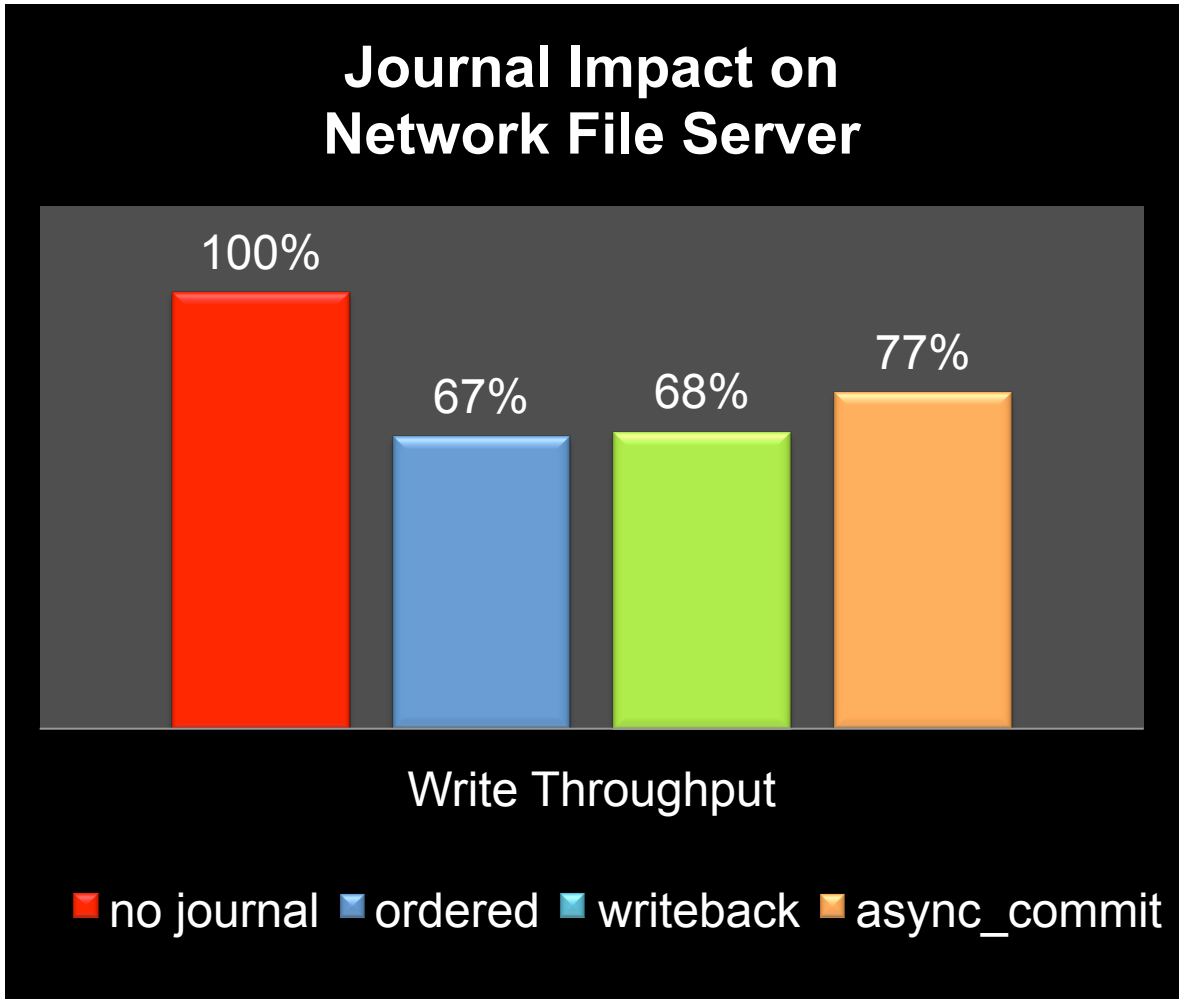
- As drives get bigger they are slower
- Clouds prefer lowest price for capacity, changes requirements

SSDs

- Many different types
- Many different failure modes
- Many different uses
- Local file systems must present a simple abstraction

Is Journaling Required?





Is Journaling Required?

Google

Big Debate:

For	Against
Restart < 10s	Performance
Coherent File System	Higher Drive Utilization
5 sec assurance	Infrequent Reboots
Much more predictable over all	Average fsck < 5 minutes/TB
	Replication

Decision

- Not deploying journals now
- Need to revisit the issue

Today & Tomorrow

Google ext4 Team

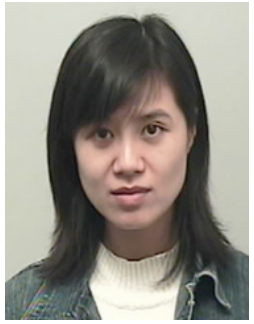
Aditya Kali

Akshay Lal

Chad Talbott

Curt Wohlgemuth

Frank Mayhar



Jiaying Zhang

Ted Tso

ext4 Team is Much Larger & Older

Google

2003 Lustre patches for mballoc, delayed allocation, extents

2006 Ted Tso agreed to drive ext4

Engineer	Company
Akira Fujita	NEC
Alex Tomas	Sun
Amir Goldstein	CTERA Networks
Andreas Dilger	WhamCloud
Aneesh Kumar	IBM
Eric Sandeen	Red Hat
Erik Whitney	HP
Lukas Czerner	Red Hat
Jan Kara	SuSE
MingMing Cao	IBM

Great time to be in Linux Storage

Google

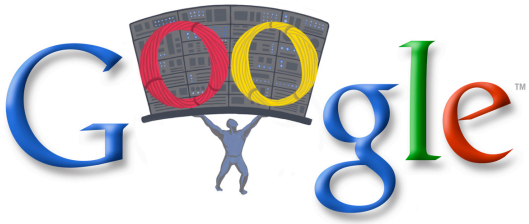
Linux is the best environment for cloud file systems

Opportunity to build solutions

- Visibility
- Predictability
- Commodity HW Diversity

You don't need a cloud to address these issues.

Anyone can hack on this.



Local file systems impact the cloud

ext4 rocks ext2

Linux is the best place for this work

Thank you!