What not to do with a Billion files in DMF

DMFUG Dec 2013

Introduction

A new beginning...

- **15PB DMF instance, ~800 million files**
  - Data dating back to the early 2000's, fronted by little over 300 spindles and 45000 tapes.
- **New t950 tape library installation, with 8x LTO6 drives,**
  - Bought to replace the existing LTO1/3 infrastructure.
- **Eternal queueing mechanism**
- **Current functional purpose, deep archive**

"Truth is, I wouldn't know a gigabyte from a snakebite." ~Dolly Parton

The problems

Performance

- New tape setup alone only provided 30% increase in performance. Roughly 18T/day
- Backups extended longer than a week
- Audits took a similar length of time

Stability

- Machine was constantly hanging or crashing, multiple times a week.
- Perception of issues was bad enough that no sysadmin dared touch it.

"I'm a Doctor, not a computer"

What we tried

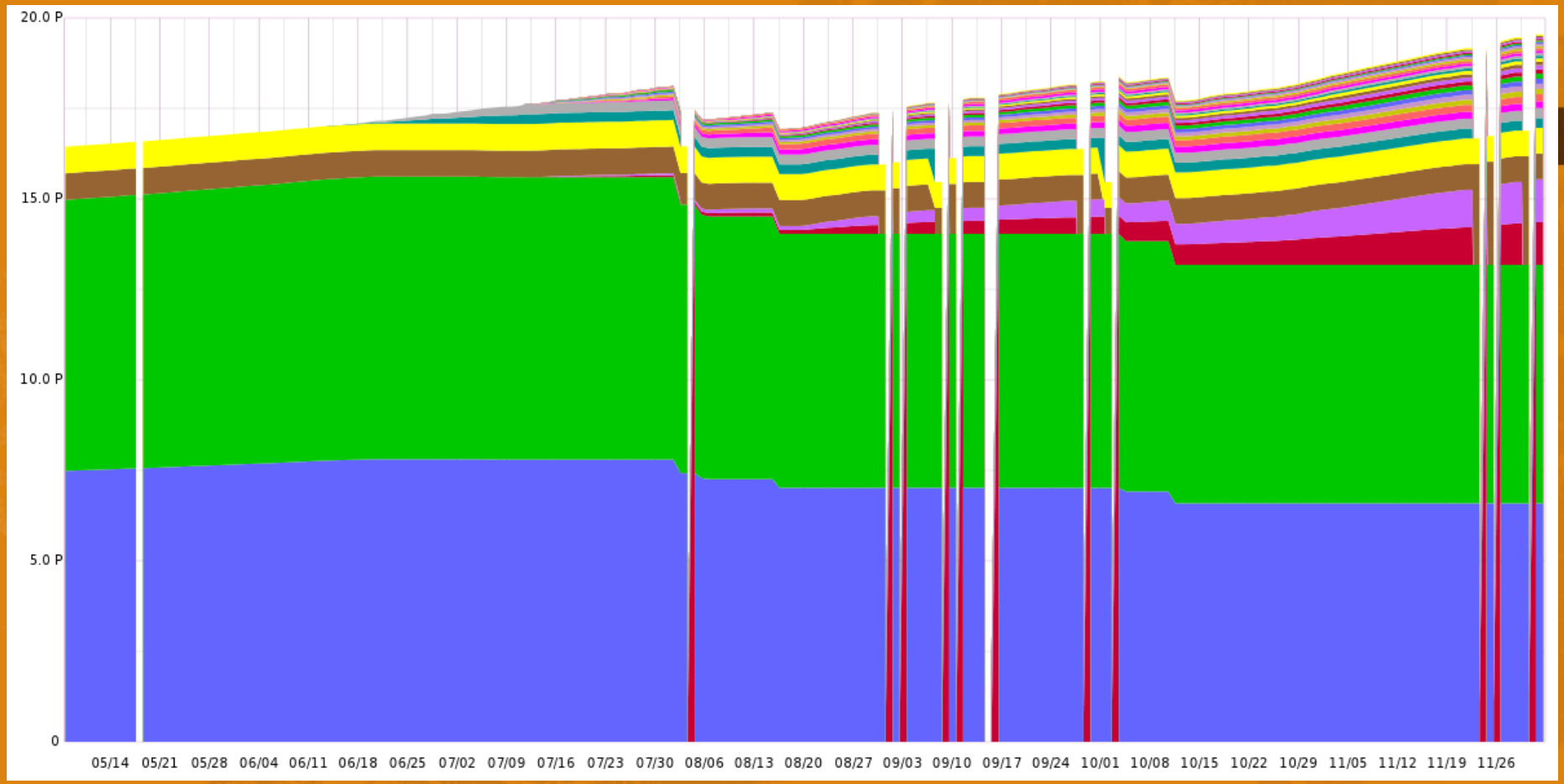
- Made it so that nothing under 4k was migrated, this kept our database smaller but made backup performance terrible.
- We deleted 100s of millions of files that we're no longer required.
- We used symbolic links to spread project data over all available filesystems
- We made our queuing mechanism aware of the destination filesystems so we could control threads
- Made 2 volume groups per filesystem to alleviate fifo bottlenecks.

"I think there is a world market for maybe five computers." - IBM Chairman Thomas Watson, 1943

What we tried, that worked

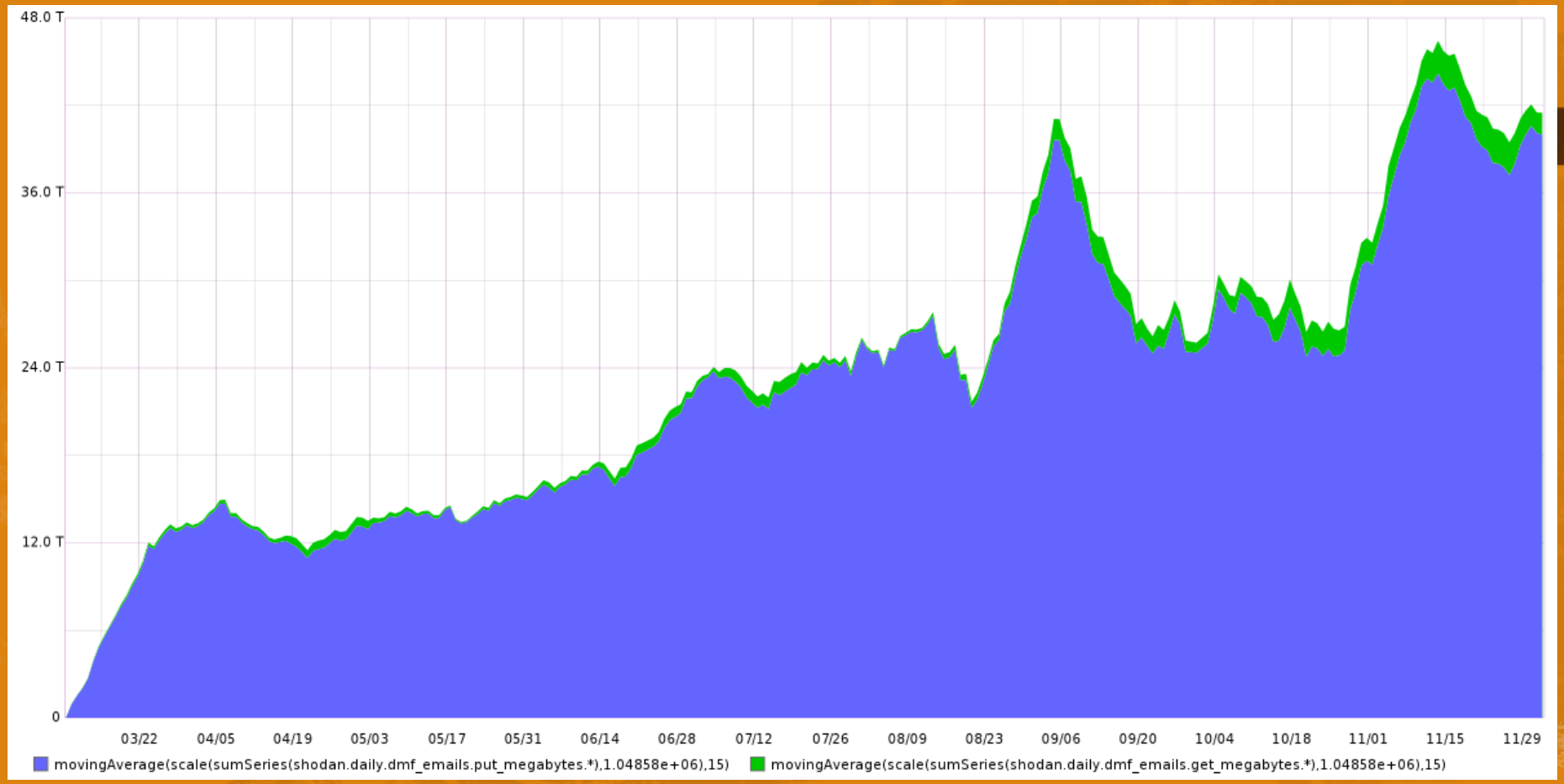
- SGI helped us identify that our main performance issues were IO bound.
  - We reconfigured our filesystems with the correct stripe alignment
  - We had available spindles so used raid 10
  - When migrating the filesystems we changed to 64bit so inodes were spread
- Controlling our queuing and volume groups gave us much better performance, as did symlinking to spread over filesystems
- SGI helped……
  - Us quickly identified a number of the crashes we were having were known issues and helped get them fixed quickly
  - Gave us the tools to gather the data to enable them to fix our issues
  - Identified the main bug we were hitting was related to IO pressure so limiting parallel housekeeping tasks helped a lot.

The results



No one will need more than 637 kb of memory for a personal computer. 640K ought to be enough for anybody. - Bill Gates, 1981

The results continued….



Moss: This Jen, is the internet.

What we missed

- Increaced performance means using it becomes more attractive. The side affect of this was that it's use case changed. It also meant that restores were more frequent and more impacting because of reliance on it's new found performance
- We exceeded 2147483647 entries in our database, this meant that dmsort failed because it "HAD" a 32bit signed integer limitation. This meant that audits and database backups failed.

Only two things are infinite, the universe and human stupidity, and I'm not sure about the former. - Albert Einstein