Performance Analysis of P2P Storage Systems

Data Lifetime Under Constrained Bandwidth and Multiple Failures

Remigiusz Modrzejewski

September 18, 2011
MASCOTTE Project

I3S(CNRS/UNS)–INRIA, PACA

Research report

This presentation is based on research report:

Frédéric Giroire, Sandeep Kumar Gupta, Remigiusz Modrzejewski, Julian Monteiro, Stéphane Perennes
Analysis of the Repair Time in Distributed Storage Systems
INRIA Rapport de recherche RR-7538, 2011

Outline

Motivation

- Indefinite backup
  - negligible read rate
  - high reliability: 10^{-5} loss probability/100GB $\rightsquigarrow$ 10^{-12} loss probability/5MB
- Cheap and scalable
  - highly distributed
  - unreliable hardware
  - uses consumer connections
- Better model
  - To be sure how design parameters shape reliability
  - Remove unasserted assumptions
  - Look into often omitted detail
  - Relate everything to costs and probability of failure
  - Thoroughly validated

Motivation

- Indefinite backup
  - negligible read rate
  - high reliability: 10^{-5} loss probability/100GB $\rightsquigarrow$ 10^{-12} loss probability/5MB
- Cheap and scalable
  - highly distributed
  - unreliable hardware
  - uses consumer connections
- Better model
  - To be sure how design parameters shape reliability
  - Remove unasserted assumptions
  - Look into often omitted detail
  - Relate everything to costs and probability of failure
  - Thoroughly validated

Motivation

- Indefinite backup
  - negligible read rate
  - high reliability: 10^{-5} loss probability/100GB $\rightsquigarrow$ 10^{-12} loss probability/5MB
- Cheap and scalable
  - highly distributed
  - unreliable hardware
  - uses consumer connections
- Better model
  - To be sure how design parameters shape reliability
  - Remove unasserted assumptions
  - Look into often omitted detail
  - Relate everything to costs and probability of failure
  - Thoroughly validated

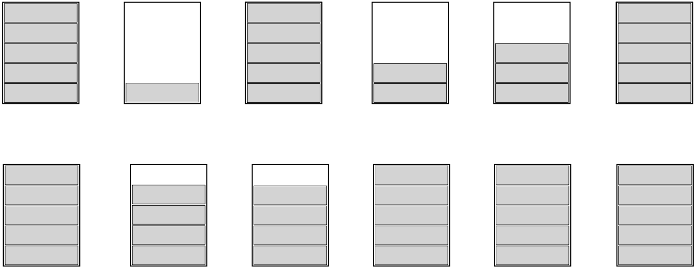
Case study

- Users have 1Mbps connections, but allocate 128kbps to repairs
- Users allocate 300GB disk space, insert 100GB data
- Expected lifetime = 1 year, neighbourhood size = 100 peers
- Repair time of 1 disk = 17 hours ($= 100 \cdot 8 \cdot 10^6$kb$/(100 \cdot 128$kbps$)$)
- Probability of data loss per year (PDLPY) of 10^{-8}
- By our model, repair time = 9 days, PDLPY = 0.2

Case study

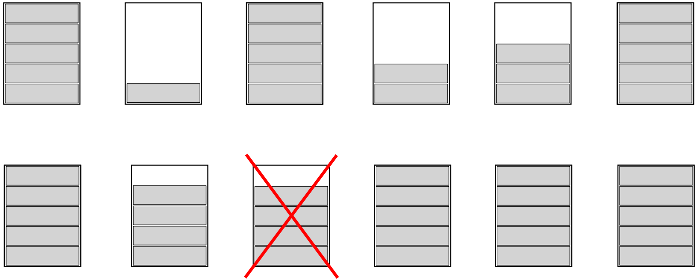
- Users have 1Mbps connections, but allocate 128kbps to repairs
- Users allocate 300GB disk space, insert 100GB data
- Expected lifetime = 1 year, neighbourhood size = 100 peers
- Repair time of 1 disk = 17 hours ($= 100 \cdot 8 \cdot 10^6$kb/($100 \cdot 128$kbps))
- Probability of data loss per year (PDLPY) of 10^{-8}
- By our model, repair time = 9 days, PDLPY = 0.2

System mechanics



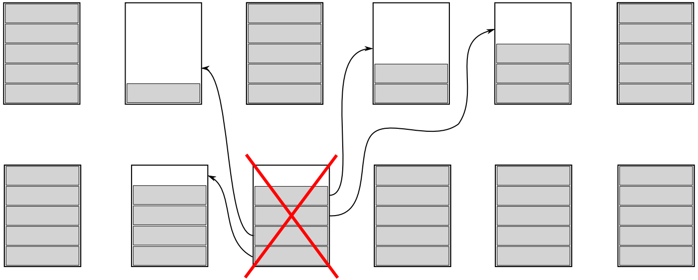
Data redundancy maintained in continuous repair process

System mechanics



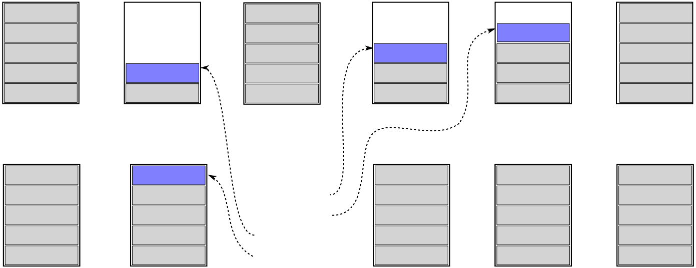
Data redundancy maintained in continuous repair process

System mechanics



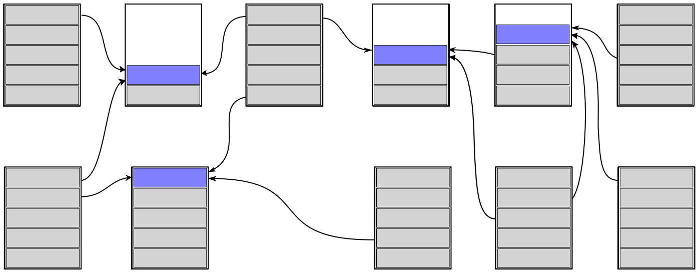
Data redundancy maintained in continuous repair process

System mechanics



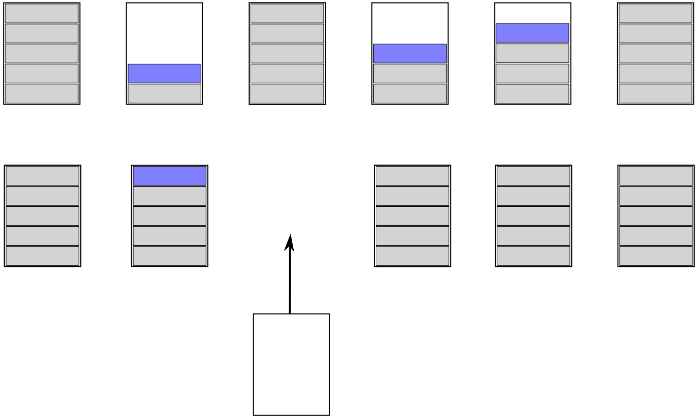
Data redundancy maintained in continuous repair process

System mechanics



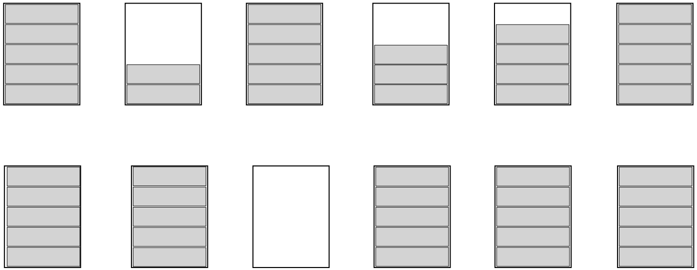
Data redundancy maintained in continuous repair process

System mechanics



Data redundancy maintained in continuous repair process

System mechanics



Data redundancy maintained in continuous repair process

Related work

Some of the similar works:

- **Analysis of Failure Correlation Impact on Peer-to-Peer Storage Systems** by Dalle et al. looks into whole disk failures, but assumes exponential reconstruction time; 2009
- **Simulation analysis of download and recovery processes in P2P storage systems** by Dandoush et al. find download/recovery time hypo-exponential, but looks only at single fragment level; 2009
- **Availability in Globally Distributed Storage Systems** by Ford et al. bases on a large body of data tracing Google storage systems; 2010

Peer imbalance

Reasons:

- New disks are **empty**, fill up gradually
  - disk load / age — truncated geometric distribution
- Workload depends on disk load
- Repairs typically need fragments from full disks

Effect:

- Repair time given by wait time for full disks
- Young disks unutilized ⇒ wasted bandwidth

Peer imbalance

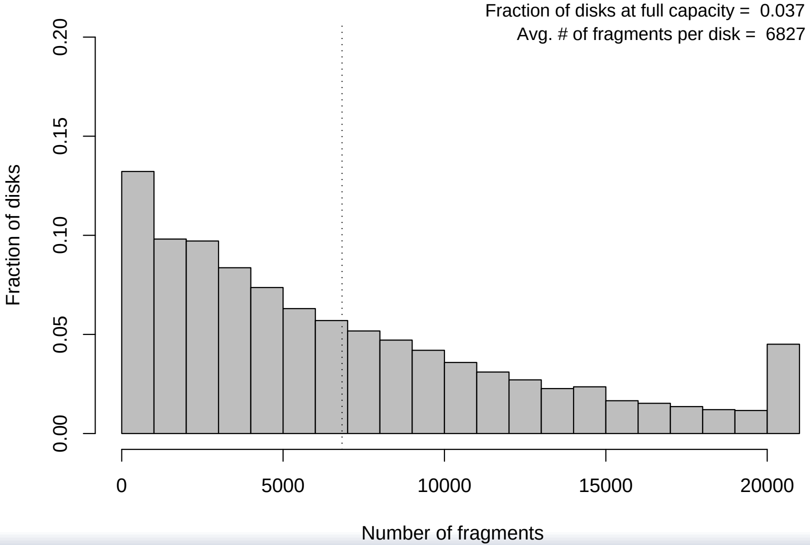
Reasons:

- New disks are **empty**, fill up gradually
  - disk load / age — truncated geometric distribution
- Workload depends on disk load
- Repairs typically need fragments from full disks

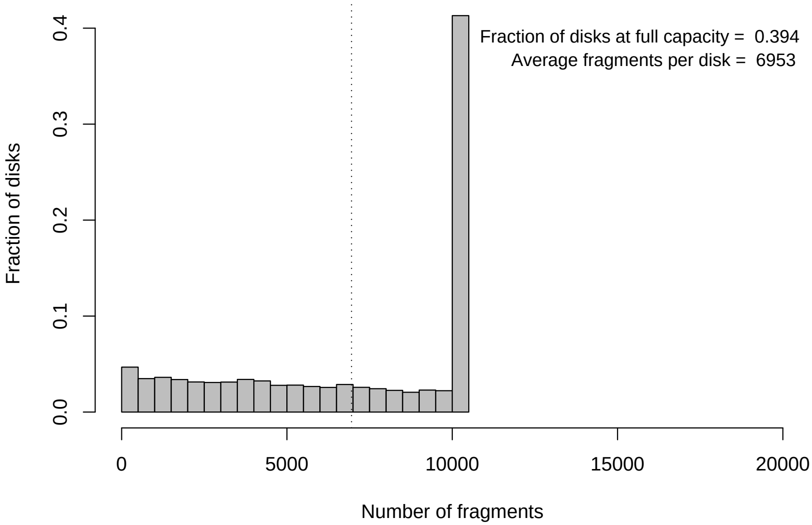
Effect:

- Repair time given by wait time for full disks
- Young disks unutilized ⇒ wasted bandwidth

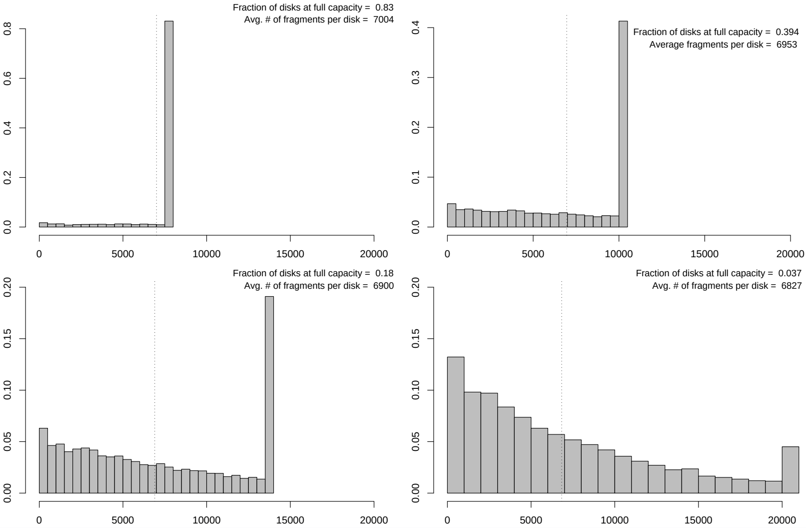
Disk load imbalance at global 33% ($x = 3$)



Disk load imbalance at global 66% ($x = 3/2$)



Disk load imbalance — comparison



Factor of efficiency

Let:

- x be the disk **overcapacity** — average capacity / average usage
- ρ be the **factor of efficiency** — total throughput / total bandwidth

We found out that:

$$\rho \approx \frac{1}{x}$$

Verification

x	1.1	1.5	2.0	3.0
φ_{sim}	0.83	0.39	0.18	0.04
φ_{model}	0.83	0.42	0.20	0.06
φx	0.91	0.63	0.4	0.18
P_{full}	$1-10^{-14}$	$1-10^{-5}$	0.999	0.92
ρ_{sim}	0.83	0.63	0.48	0.40
ρ_{model}	0.91	0.67	0.5	0.33
T_{sim}	1.07	2.69	8.55	21.76
T_{model}	1.00	2.61	17.81	54.61

Where: φ – fraction of full disks in network; P_{full} – probability of a block to have ≥ 1 fragment on a full disk; T – reconstruction time; $n = 14$ – # fragments for each block

Verification

x	1.1	1.5	2.0	3.0
φ_{sim}	0.83	0.39	0.18	0.04
φ_{model}	0.83	0.42	0.20	0.06
φx	0.91	0.63	0.4	0.18
P_{full}	$1 - 10^{-14}$	$1 - 10^{-5}$	0.999	0.92
ρ_{sim}	0.83	0.63	0.48	0.40
ρ_{model}	0.91	0.67	0.5	0.33
T_{sim}	1.07	2.69	8.55	21.76
T_{model}	1.00	2.61	17.81	54.61

Even when full disks are rare, most blocks have a fragment on them

Verification

x	1.1	1.5	2.0	3.0
φ_{sim}	0.83	0.39	0.18	0.04
φ_{model}	0.83	0.42	0.20	0.06
φx	0.91	0.63	0.4	0.18
P_{full}	$1 - 10^{-14}$	$1 - 10^{-5}$	0.999	0.92
ρ_{sim}	0.83	0.63	0.48	0.40
ρ_{model}	0.91	0.67	0.5	0.33
T_{sim}	1.07	2.69	8.55	21.76
T_{model}	1.00	2.61	17.81	54.61

Model closely matches simulations

Verification

x	1.1	1.5	2.0	3.0
φ_{sim}	0.83	0.39	0.18	0.04
φ_{model}	0.83	0.42	0.20	0.06
φx	0.91	0.63	0.4	0.18
P_{full}	$1 - 10^{-14}$	$1 - 10^{-5}$	0.999	0.92
ρ_{sim}	0.83	0.63	0.48	0.40
ρ_{model}	0.91	0.67	0.5	0.33
T_{sim}	1.07	2.69	8.55	21.76
T_{model}	1.00	2.61	17.81	54.61

x has high impact on reconstruction time

Verification

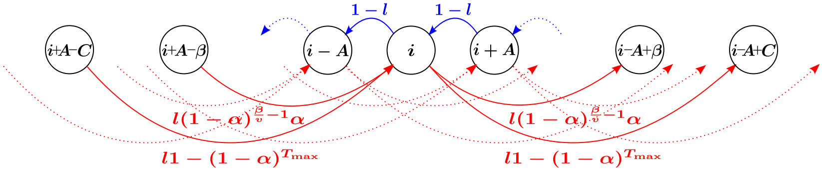
x	1.1	1.5	2.0	3.0
φ_{sim}	0.83	0.39	0.18	0.04
φ_{model}	0.83	0.42	0.20	0.06
φx	0.91	0.63	0.4	0.18
P_{full}	$1-10^{-14}$	$1-10^{-5}$	0.999	0.92
ρ_{sim}	0.83	0.63	0.48	0.40
ρ_{model}	0.91	0.67	0.5	0.33
T_{sim}	1.07	2.69	8.55	21.76
T_{model}	1.00	2.61	17.81	54.61

Model would need extension for big x,
but this represents inefficient resource usage

Why a queueing model?

- Target system has **many** peers
- We want to know what happens in years of work
- Simulations would consume prohibitive amounts of time
  - some operations done on each block in each time step
  - 100'000 peers
  - 100GB per peer / 5MB blocks = 20'000 blocks per peer
  - 10 years with 1 hour resolution
  - over 5 years of simulation assuming 10^6 operations / second

Markovian queuing model



- Global queue of all blocks needing repair
- $M^{\beta}/D/1$, β is the batch size function
- States — number of fragments in queue
- Transitions — reconstructions or failings
- 2 batch sizes — full disk; expected value of non-full disk

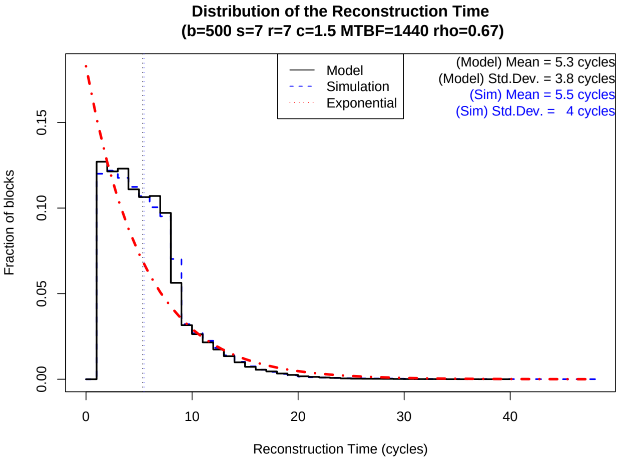
What does it give?

- Waiting time given directly by stationary state of the queue
- Same goes for bandwidth usage
- Expected data loss computed using stationary state
- Stationary state computed semi-analytically or numerically
- Implementation in R converges in <2s

Simulations/Experiments

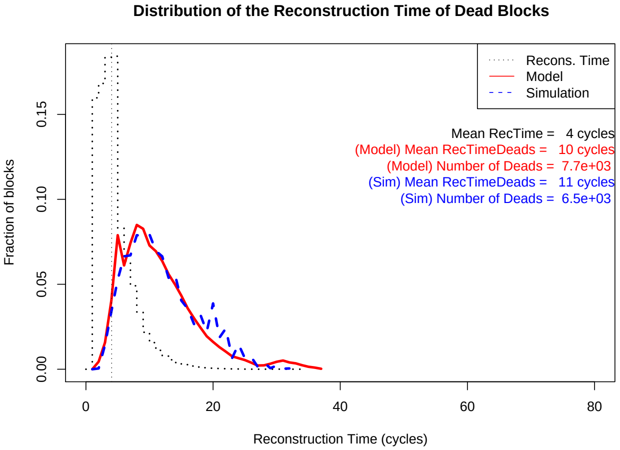
- Markovian queuing model implemented in R
- Custom simulator implemented in Java
- Experiments based on UbiStorage system deployed on Grid5000

Reconstruction time: model vs simulation



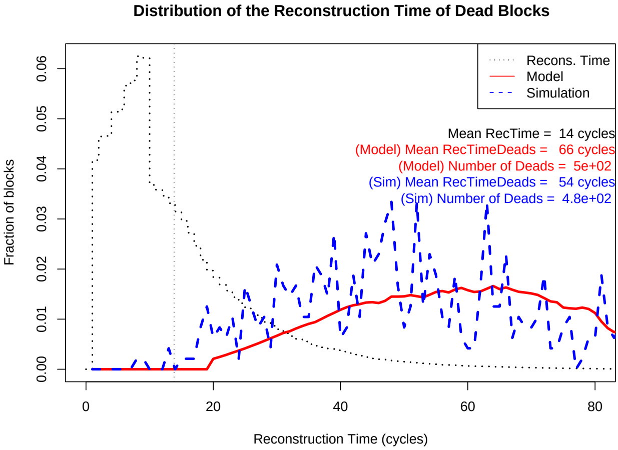
Not always exponential

Lost data: model vs simulation



With too much short repairs; almost perfect fit

Lost data: model vs simulation

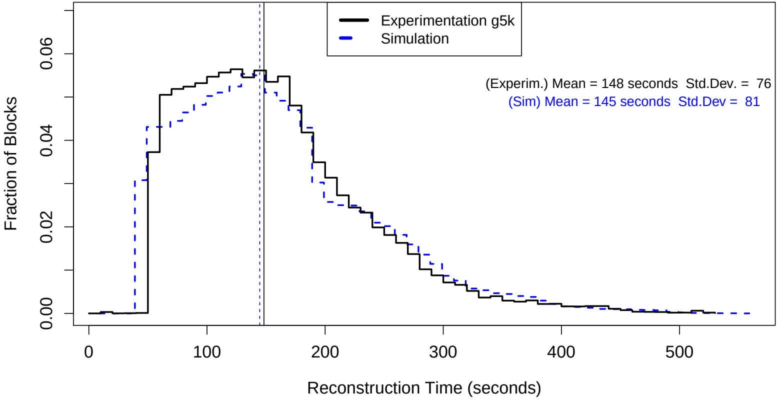


For the long tail; smooth fit

Experimentation setup

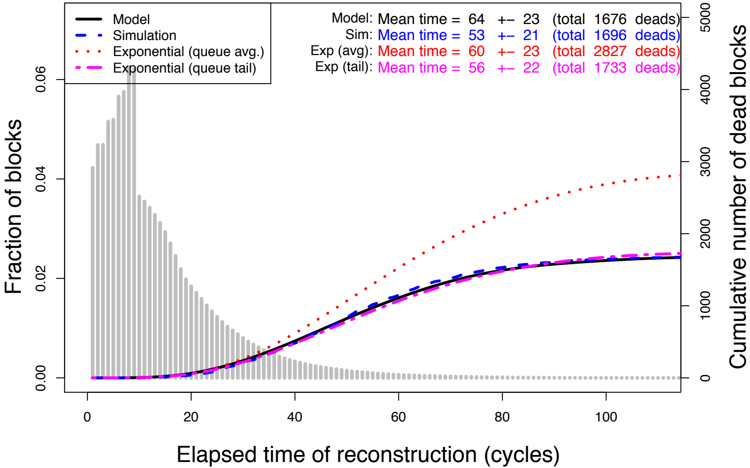
- An overlay of 50-200 peers built on Grid5000 nodes
- Failures according to traces or a random process
- Acceleration factor of 3–350
  - All times compressed
  - Less data
  - Limited bandwidth
- Same parameters fed into simulator and model

Experimentation results



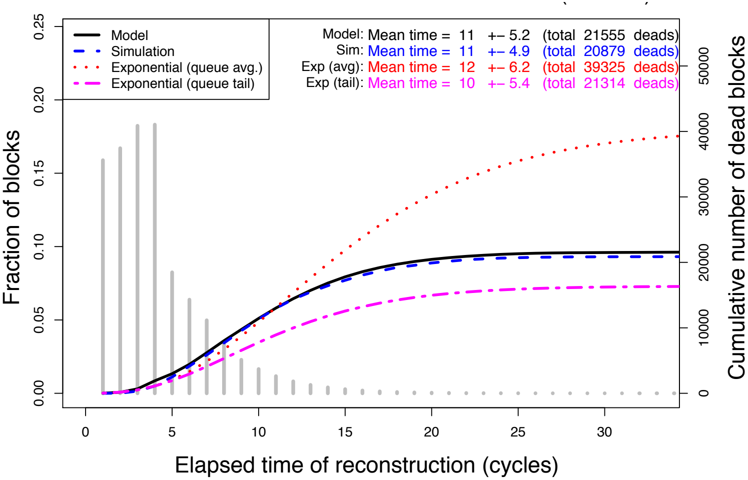
Pretty good fit too

Exponential, average and tail



Sometimes exponential is good enough if fitting tail

Exponential, average and tail



Sometimes it is not

Take-aways

- Intuition is not sufficient
- Simplistic methods fit only some scenarios
- Seemingly irrelevant details do matter
- Simple, accurate and validated model was proposed