Making (Almost) Everything More Efficient

Jehan-François Pâris
jfparis@uh.edu

How?

- ***Performance Evaluation:***
  - Coming with ***more efficient solutions*** to ***known problems***
  - Kind of ***operational research***
    - Applied to CS

The techniques

- ***Probabilistic models:***
  - Simple Markov chains
  - …
- ***Simulation:***
  - Mostly discrete
    - Faster
    - Better suited to most CS problems

Applications (I)

- Improving program performance in virtual memory systems
  - *A long long time ago*
- Finding better ways to manage replicated data
  - *A long time ago*
- Finding more efficient ways to distribute video on demand
  - *Great fun*

Applications (II)

- Finding more effective file prefetching policies
- Using P2P networks to distribute videos in real-time
- Finding better fault-tolerant storage organizations
- Using P2P networks to manage highly replicated data

An Example: Data Hardening

The problem

- Archival data
  - Now stored on hard disks
- Losing data is not an option
  - 99.999% reliability over five years
  - Must protect them against disk failures
  - Many solutions
    - RAID level 6, …

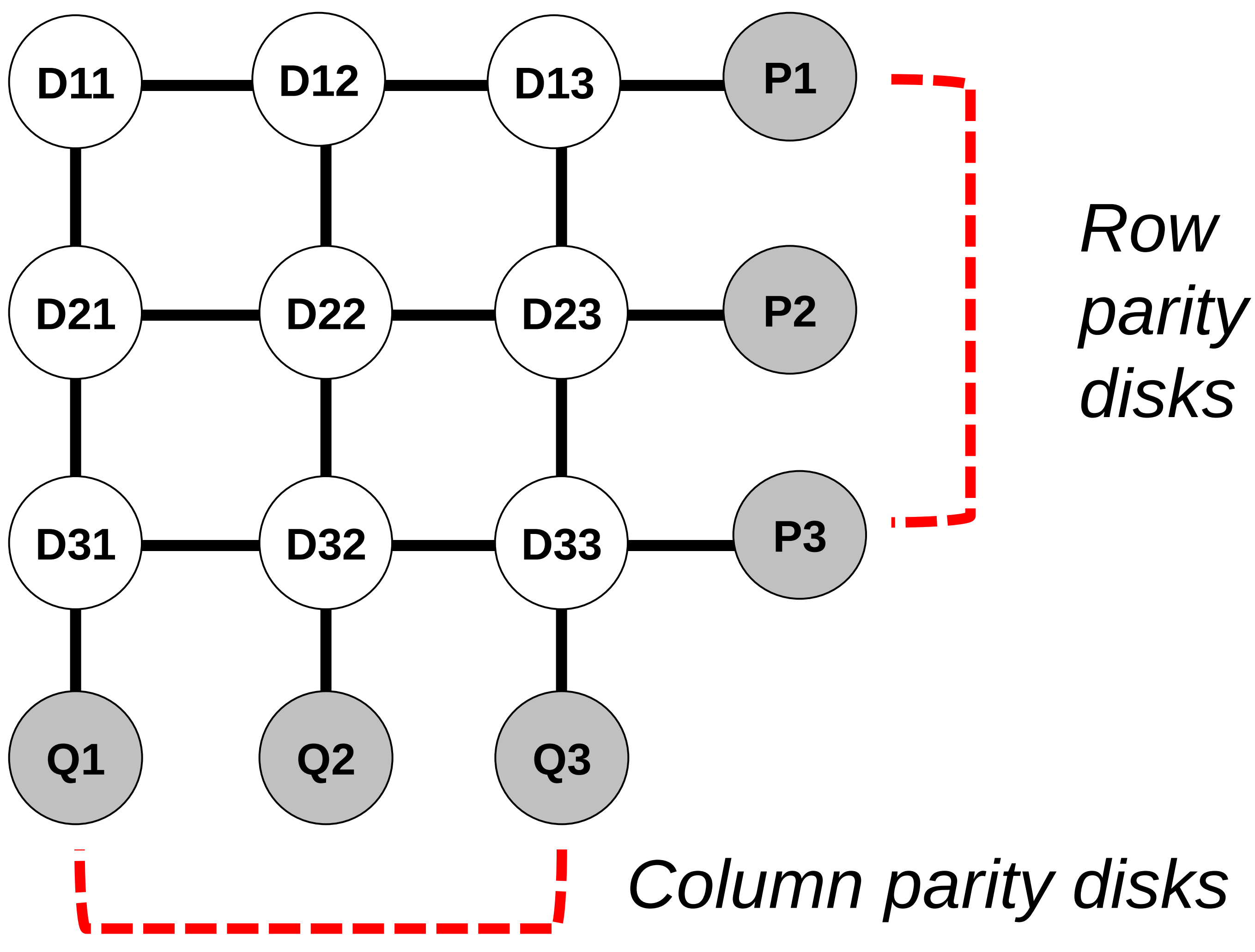
Picking the right solution

- Normal failure rate for disk drives is around 0.04 failures/year
- Can go up to .25 failures/year for some disks from reputable brands
- *How to pick the* ***right level*** *of protection?*

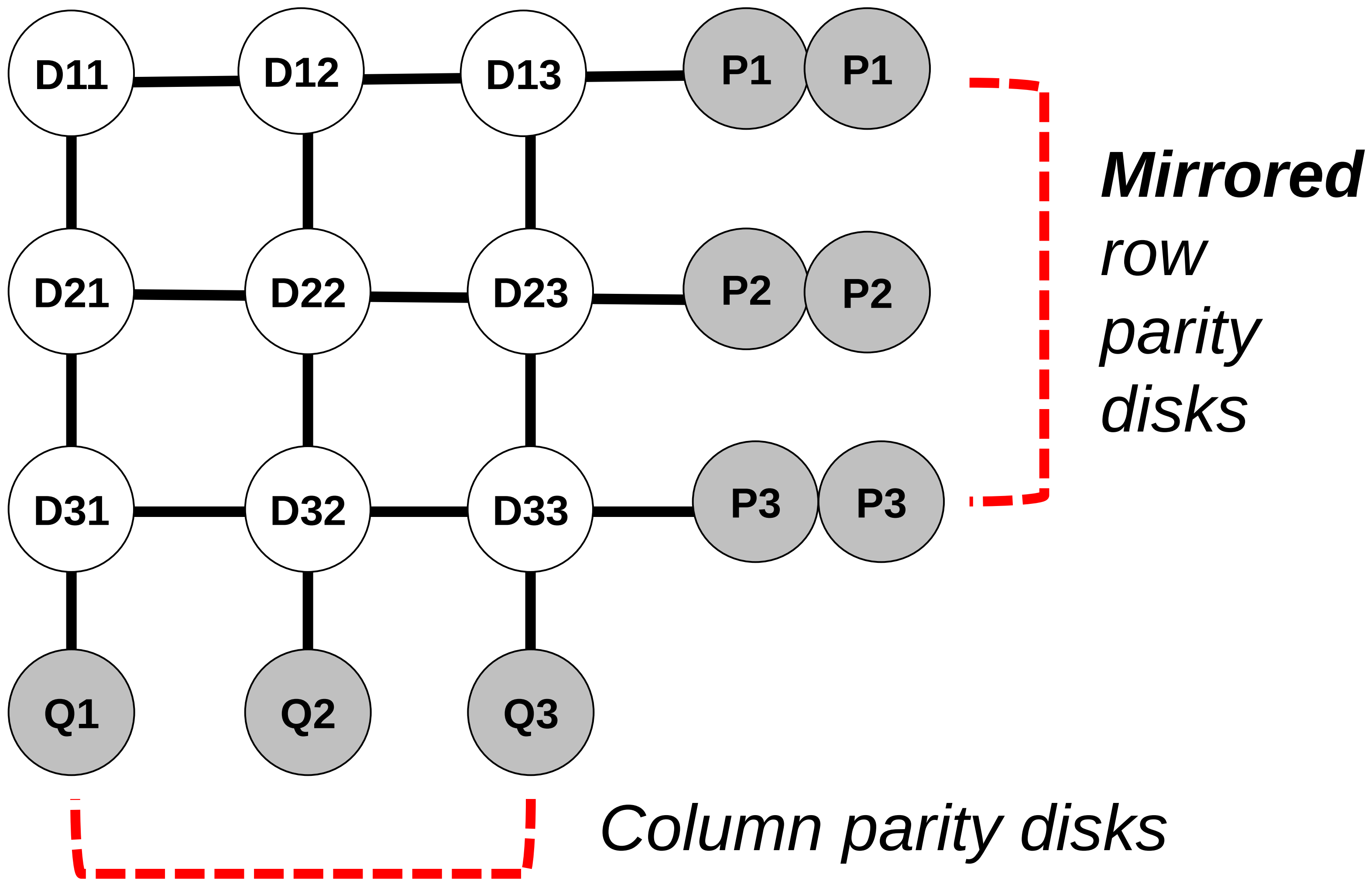
Our proposal

- Design disk arrays for the most frequent case:
  - 0.04 failures/year
- Add extra protection when needed:
  - Higher failure rates
- Like putting struts under a sagging floor
- ***It works!***

A two-dimensional RAID array



Same with added protection



If you want to know more

- Look at my recently published papers
  - On my web page
- Talk to me
  - Your topic or mine?

Thank you!