Proteus: A Flexible Simulation Tool for Estimating Data Loss Risks in Disk Arrays

Hsu-Wan Kao, U. of Houston
Jehan-François Pâris, U. of Houston
Thomas Schwarz, U. Católica del Uruguay
Darrell D. E. Long, U. of California, Santa Cruz

Motivation

The problem

- Estimating risks of data loss for fault-tolerant disk arrays is not an easy task
- Two options
  - Analytical methods
  - Simulation approach

Analytical Methods

- Based on Markov models
- Requirements:
  - Failures to be Poisson processes
    - Reasonable but not necessarily true
  - Repairs to be Poisson processes
    - ***Not true***

Simulation Approach

- Can simulate almost anything
- Only produces numerical results
  - Must repeat simulation for each parameter value (MMTF, MTTR, ...)
- Must write a different program for each disk array configuration
  - Takes time and effort

The Proteus Simulation Program

Proteus

- ***Flexible***
  - Program can model very different disk array configurations
- ***Fast***
  - Very simple model that runs fast
- ***Portable***
  - Written in Python 3.x

The secret

- Proteus characterizes any disk array configuration using only five parameters
  - Express which fraction of
    - Single,
    - Double,
    - Triple,
    - …

  disk failures will result in a data loss

The five parameters

- The number ***n*** of disks the array comprises
- The number $\boldsymbol{n_f}$ of failures it will ***always***s tolerate
- The fraction $\boldsymbol{f_1}$ of failures of $\boldsymbol{n_f}$ **+ 1** disks it will tolerate
- The fraction $\boldsymbol{f_2}$ of failures of $\boldsymbol{n_f}$ **+ 2** disks it will tolerate
- The fraction $\boldsymbol{f_3}$ of failures of $\boldsymbol{n_f}$**+ 3** disks it will tolerate

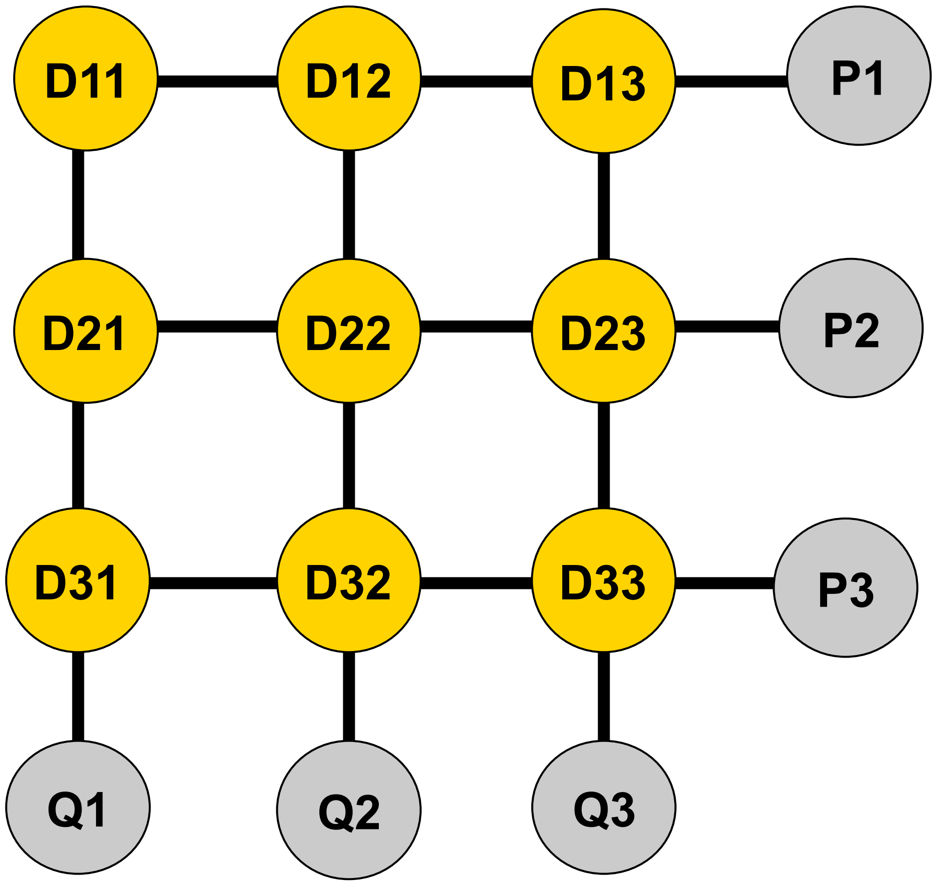
A very simple example

- RAID level 5 arrays tolerate
  - All single disk failures
  - No double failures
- Array parameters are
  - $\boldsymbol{n = 5, n_f = 1, f_1 = f_2 = f_3 = 0}$

Another simple example

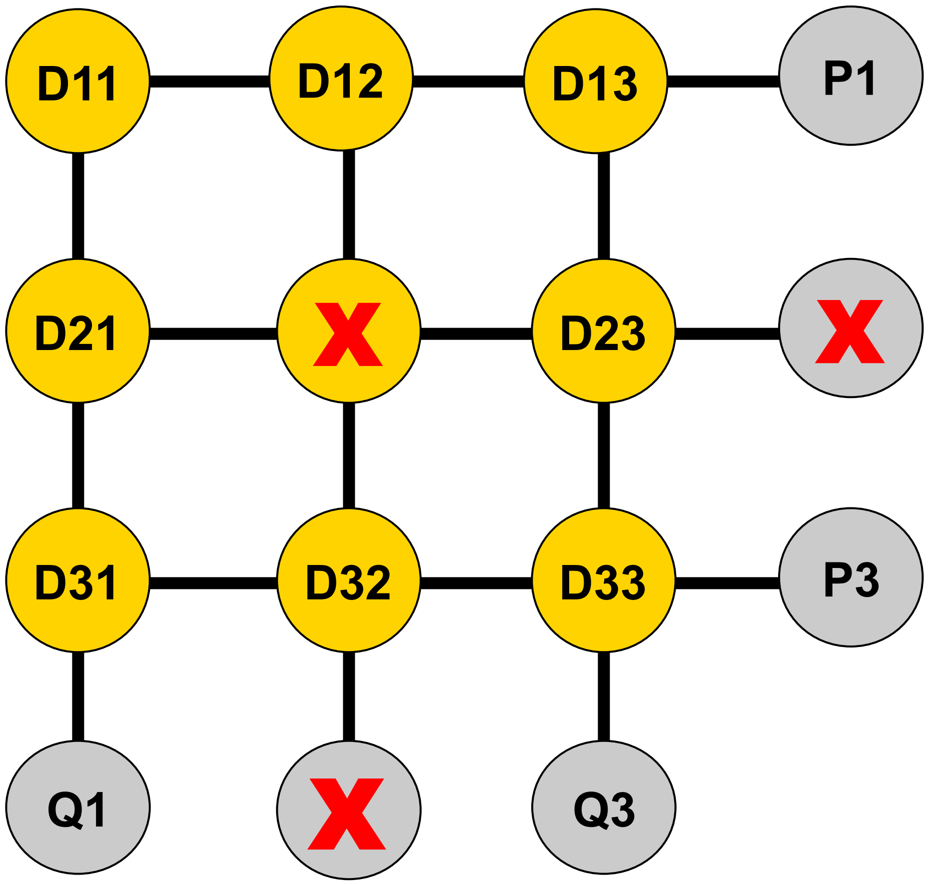
- RAID level 6 arrays tolerate
  - All single and double disk failures
  - No triple disk failures
- Array parameters are
  - $\boldsymbol{n = 5, n_f = 2, f_1 = f_2 = f_3 = 0}$

Something more complicated: A two-dimensional RAID array



Tolerates all double disk failures

A triple failure causing a data loss



One data disk and its two parity disks

Fatal triple failures

- Array has n^2 data disks and $2n$ parity disks
- Only n^2 of all $\binom{n^2+2n}{4}$ possible quadruple failures result in a data loss
- We have $n_f = 2$ and $f_1 = 1 - \frac{n^2}{\binom{n^2+2n}{4}}$

Quadruple and quintuple failures

- Can redo same analysis to compute fraction of quadruple failure that do not result in a data loss
  - More complicated
  - Outcome is value of f_2
- Assumed $f_3 = 0$
  - Would not make a difference

Limitation

- Assumes we can neglect contributions of array states with more than $\boldsymbol{n_f + 3}$ simultaneous disk failures to array reliability
- Good assumption for small to medium-size disk arrays
- ***Not always true*** for ***very large disk arrays***

Experimental Results

Scope

- Simulated
  - RAID level 5 array with 5 disks
  - RAID level 6 array with 10 disks
  - Two dimensional RAID array with 64 data disks and 16 parity disks
- Disk mean time to fail was set to 100,000 hours
- Disk mean time to repair varied between ½ day and ten days

Outcomes

- Measured probability each disk array would suffer no data loss over five years
- Observed
  - Perfect agreement with analytical results obtained using Markov chains
  - No difference between results obtained assuming
    - Exponential repair times
    - Deterministic repair times

Conclusions

Conclusions

- Proteus allows us to estimate reliability of many disk array organizations
  - Without having to write a new simulation program
  - Without having to assume that failures and repairs are Poisson processes
    - Can use Weibull distribution for failures
    - Can use deterministic repair times

Availability

- Proteus is free and can be downloaded from
  - www.cs.uh.edu/~paris/Proteus
  - http://www.ssrc.ucsc.edu/proteus.html

Thank you!

- Any questions?