Sistemas Distribuídos
Aula 22

Aula passada

- Replicação
- Conflitos
- Modelos de consistência
- Modelos de consistência no cliente

Aula de hoje

- *Reliability* e *Availability*
- Modelo de falhas
- Falhas na prática
- Redundância
- *Triple Module Redundancy*
- Tipos de falha

Falhas em Sistemas

- Por que sistemas reais falham?
- O que é uma falha?

- Queremos construir sistemas que sejam *tolerantes a falhas (fault tolerant)*
  - tolerância a falhas é um conceito vago
- Definir propriedades que um sistema deve oferecer
  - *availability, reliability, safety, maintainability*
- Tolerante a falhas é ter bons índices nestas métricas

Reliability (*confiabilidade*)

- **Reliability**: tempo operacional continuamente até que falha ocorra
  - ex. tempo até lâmpada queimar
- Tempo geralmente é aleatório
  - distribuição desconhecida ou complicada
  - MTTF: Mean Time To Failure (média)
- Outro conceito relacionado: tempo em falha contínua até ser reparado
  - ex. tempo até a lâmapada ser trocada depois de falhar
  - MTTR: Mean Time To Repair
- Mean Time Between Failures (MTBF) = MTTF + MTTR

Availability (*disponibilidade*)

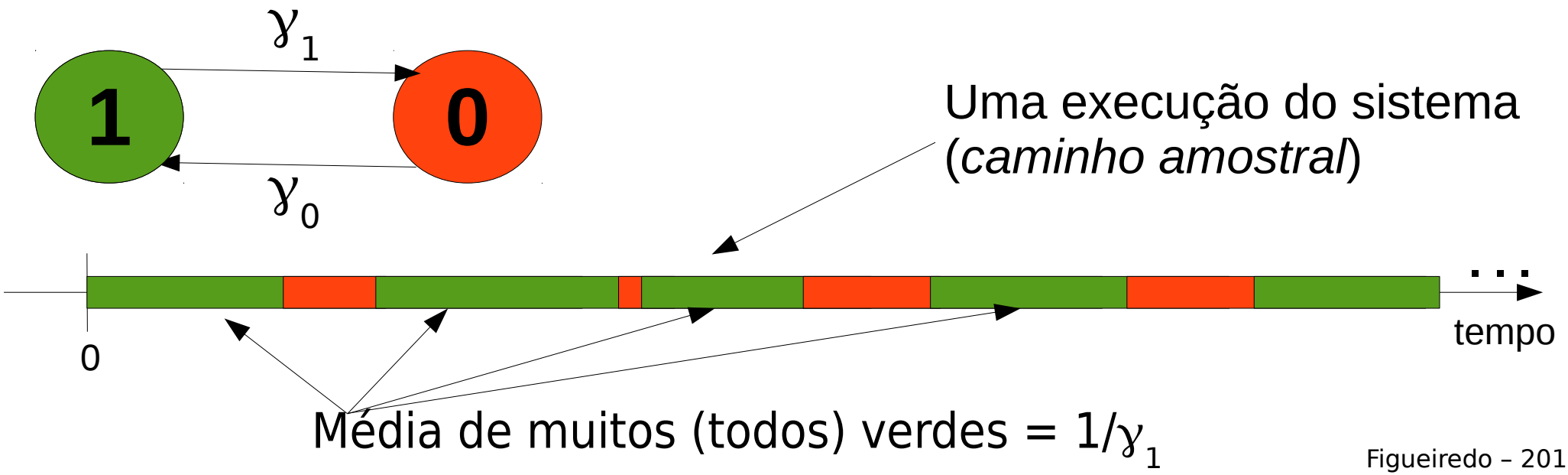
- **Availability**: fração de tempo que sistema está operacional
  - ex. fração de tempo que lâmpada está acesa
- Definida em função de *reliability* e *recovery*
  - na média, A = MTTF/(MTTF+MTTR)
- Exemplo
  - lâmpada dura em média 4.5 anos e demora em média 1.5 dias para ser trocada
  - MTTF = 4.5 anos = 4.5*365 = 1642.5 dias
  - MTTR = 1.5 dia
  - A = 1642.5/(1642.5+1.5) = 0.999087

Availability != Reliability

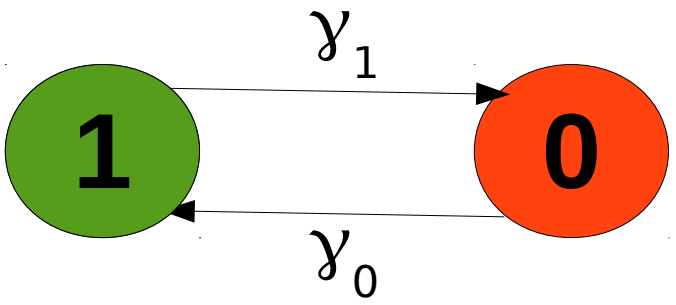
- Conceitos são fundamentalmente diferentes
  - reliability: duração de tempo (até falha ou reparo)
  - availability: fração de tempo
- Reliability alto e availability baixo?
  - Sim: MTTF = 5 anos, MTTR = 3 anos
- Availability alto e reliability baixo?
  - Sim: MTTF = 10 s, MTTR = 1 ms
- Sistema deve oferecer alto realiability, alto availability

Modelo de Falhas

- Abstração matemática para falhas
- Assume que sistema possui dois estados
  - operacional (1), falho (0)
- Alterna entre os estados
  - tempo em cada estado aletatório, com médias $1/\gamma_1$ e $1/\gamma_0$ (γ_1: *taxa* de falha, γ_0: *taxa* de reparo)



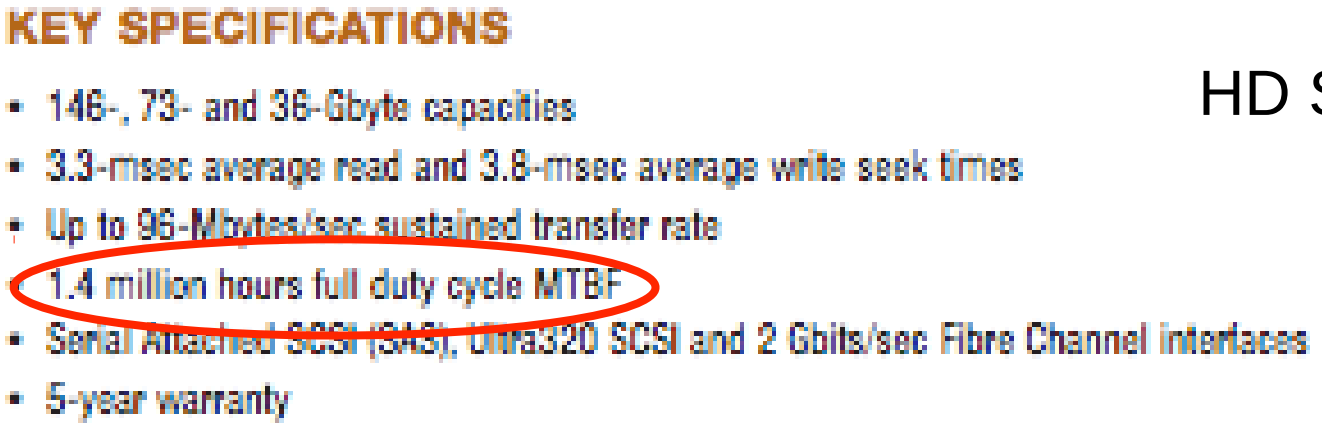
Modelo de Falhas



- Assumir que tempo até falhar e reparar possui distribuição exponencial
  - P(Falha < t) = 1 – $e^{-\gamma_1 t}$
- Neste caso, temos uma cadeia de Markov!
- P[1] = fração tempo no estado 1 (depois de muito tempo)
  = $\gamma_0/(\gamma_1+\gamma_0)$ = availability (pela definição)
- Depende apenas da razão (availability != reliability)
- Modelo permite responder muitas outras perguntas
  - ex. quantas falhas em T unidades de tempo?

MTTF na Prática

- Fabricante deve especificar MTTF do componente
  - ex. CPU, memória, disco, etc
  - necessário para calcular MTTF do *sistema*
- Para HDs, MTTF entre 300K e 1.5M horas



KEY SPECIFICATIONS

- 146-, 73- and 36-Gbyte capacities
- 3.3-msec average read and 3.8-msec average write seek times
- Up to 96-Mbytes/sec sustained transfer rate
- 1.4 million hours full duty cycle MTBF
- Serial Attached SCSI (SAS), Ultra320 SCSI and 2 Gbits/sec Fibre Channel interfaces
- 5-year warranty

HD Seagate Cheetah

- 1M horas = 114 anos: como é possível?
- Valor estimado com milhares casos em condições ideais, extrapolado com modelo matemático
  - realidade é bem diferente!

Estudo de Falhas em Sistemas

- Caracterização de falhas em grandes sistemas reais
  - +1000 servidores, discos, etc
- Fração relativa de troca de componentes

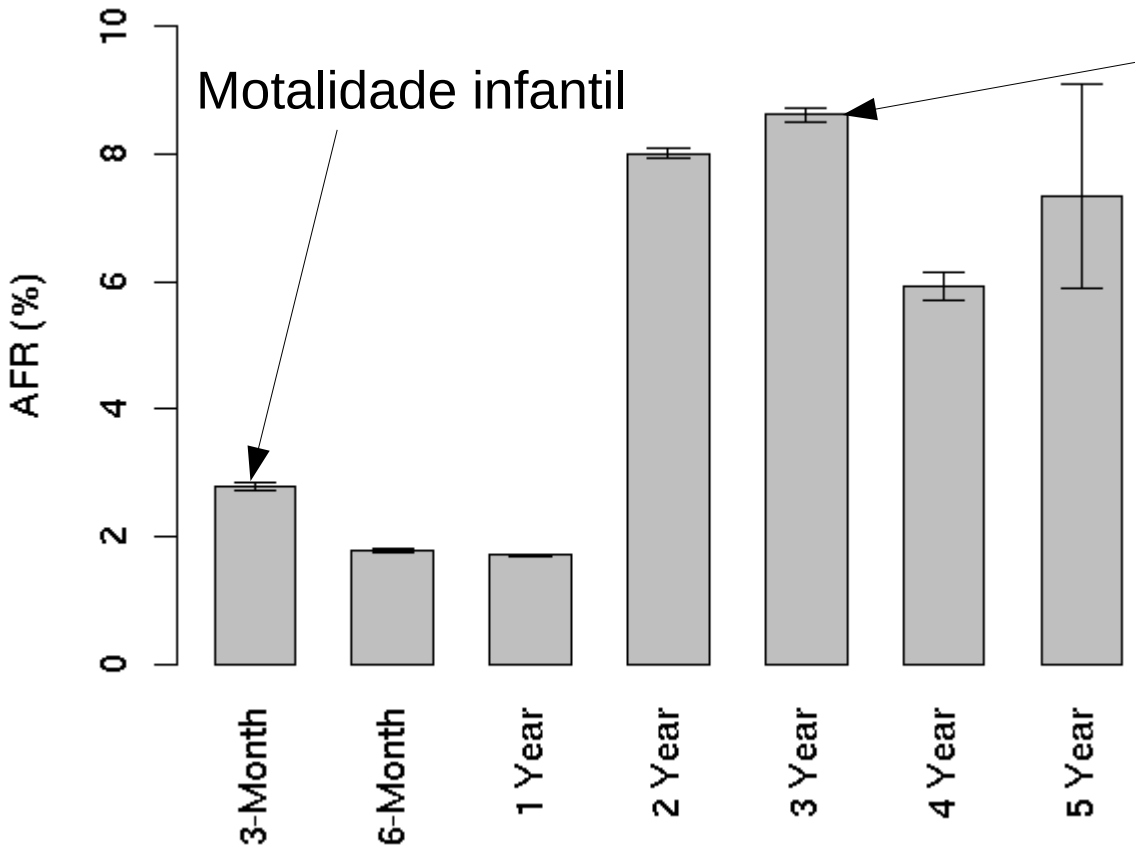
HPC1		COM1		COM2	
Component	**%**	**Component**	**%**	**Component**	**%**
Hard drive	30.6	Power supply	34.8	Hard drive	49.1
Memory	28.5	Memory	20.1	Motherboard	23.4
Misc/Unk	14.4	Hard drive	18.1	Power supply	10.1
CPU	12.4	Case	11.4	RAID card	4.1
motherboard	4.9	Fan	8	Memory	3.4
Controller	2.9	CPU	2	SCSI cable	2.2
QSW	1.7	SCSI Board	0.6	Fan	2.2
Power supply	1.6	NIC Card	1.2	CPU	2.2
MLB	1	LV Pwr Board	0.6	CD-ROM	0.6
SCSI BP	0.3	CPU heatsink	0.6	Raid Control.	0.6

- Disco e memória estão entre principais causas de reparo

Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?
Bianca Schroeder, Garth A. Gibson
5th USENIX Conference on File and Storage Technologies, 2007

Estudo da Google sobre Discos

- Discos de um serviço (interno) da Google
- Estudo com +100 mil discos em 5 anos
  - taxa anual de falhas por idade



Características do fabricante, e não envelhecimento

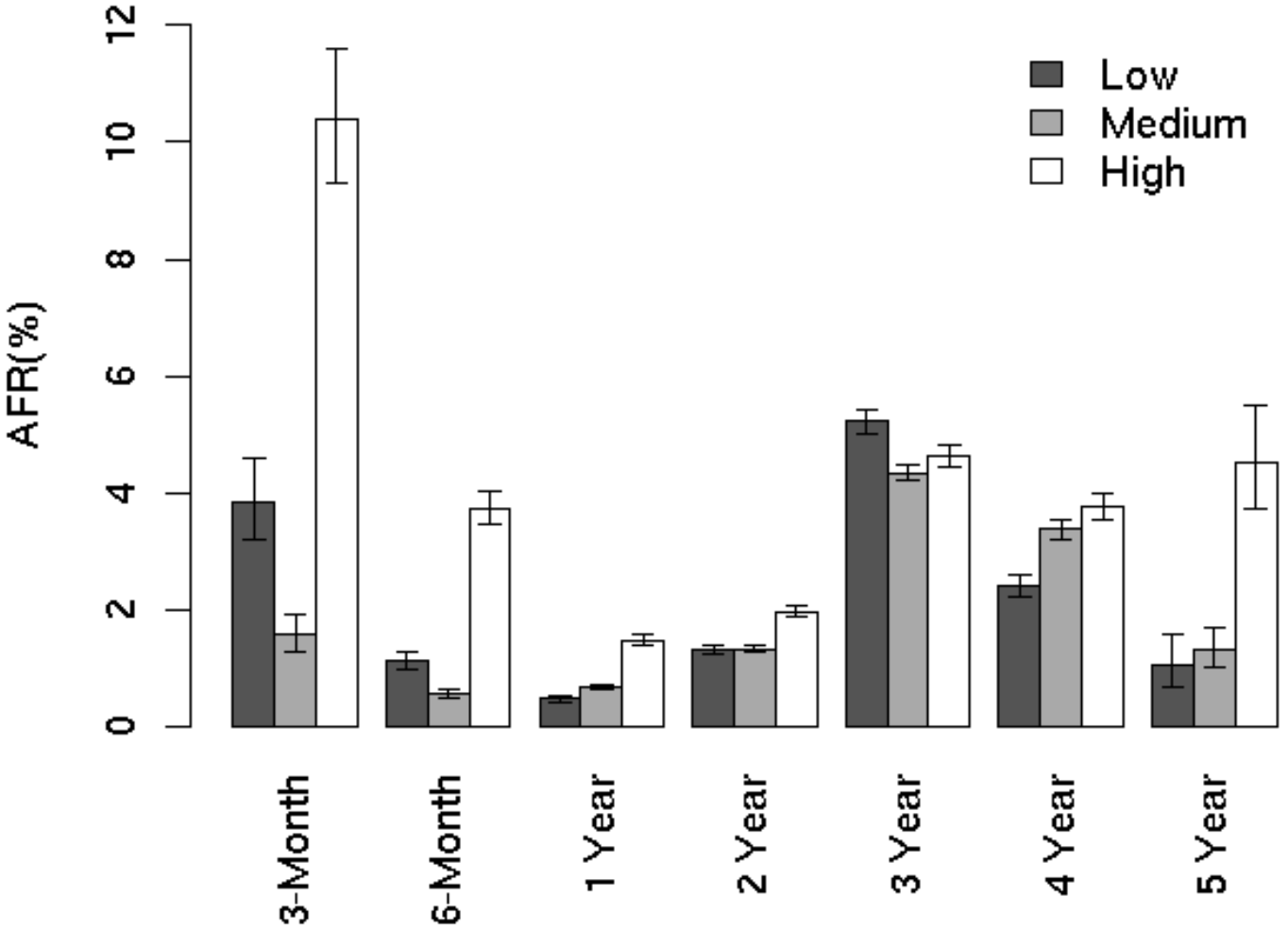
- Disco com 1 ano falham menos do que disco com 3 meses
- Disco com 3 anos falham mais que 4 anos

Ex-aluno de mestrado do PESC, hoje na Google!

Failure Trends in a Large Disk Drive Population
Eduardo Pinheiro, Wolf-Dietrich Weber, and Luiz André Barroso
5th USENIX Conference on File and Storage Technologies, 2007

Estudo da Google sobre Discos

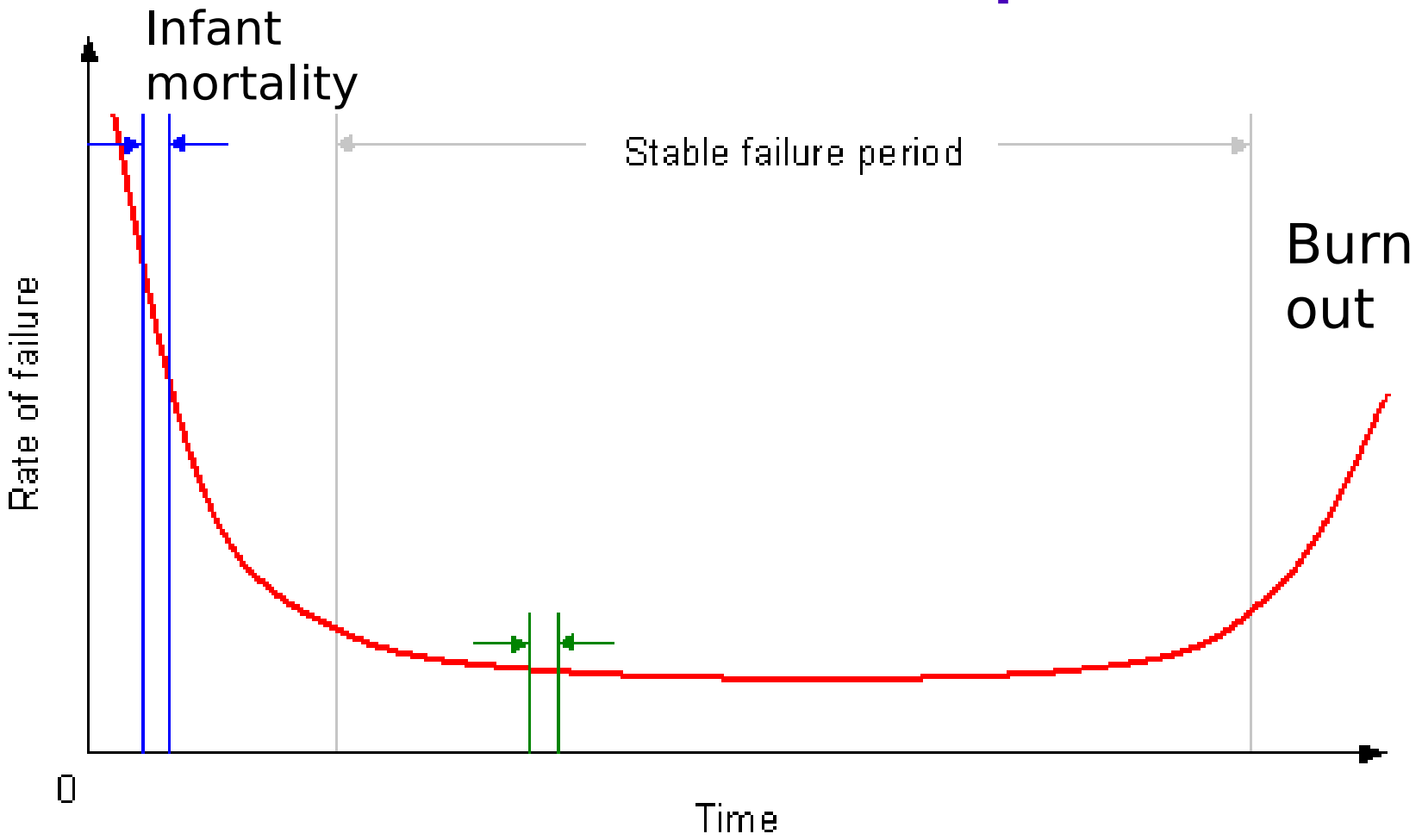
- taxa anual de falhas por idade e carga
  - carga: bytes total escritos/lidos em uma semana



- Disco com mais carga falham mais, principalmente quando jovens!

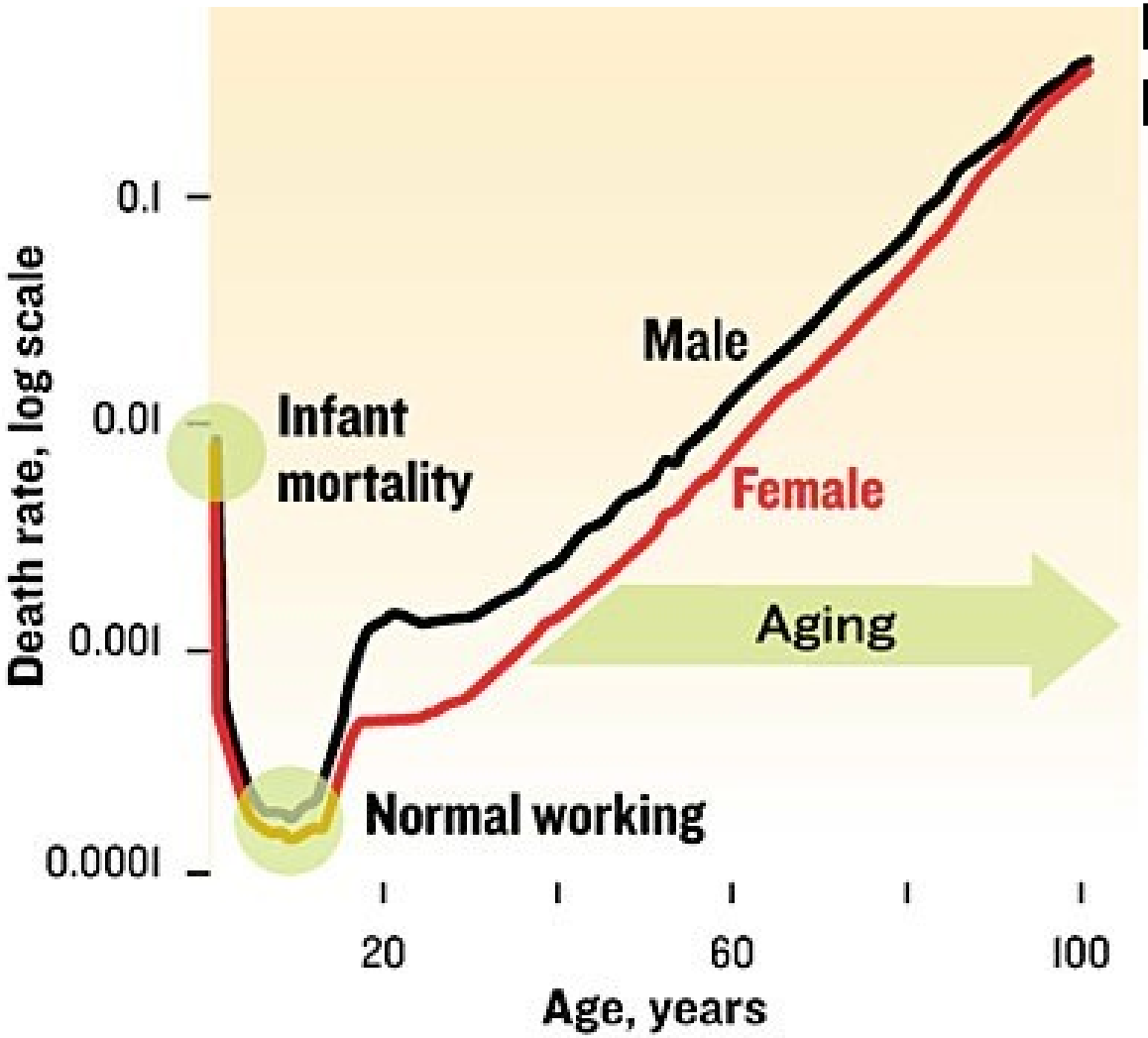
Failure Trends in a Large Disk Drive Population
Eduardo Pinheiro, Wolf-Dietrich Weber, and Luiz André Barroso
5th USENIX Conference on File and Storage Technologies, 2007

Taxa de Falha de HDs (e outros componentes)



- Modelo geral para taxa anual de falha
- Curva do tipo *bathtube* (banheira)
  - taxa maior no início, depois cresce com o tempo

“Taxa de Falhas” em Pessoas



- Outra curva do tipo banheira
- Ser humano é formado por componentes
- Modelos de sobrevida similar?

L. Gavrilov & N. Gavrilova, “Why We Fall Apart,” IEEE Spectrum, 2004.
Data from http://www.mortality.org

Lidando com Falhas

- Como lidar com falhas?

Redundância!

- Natureza faz isto (dois olhos, dois rins, etc)
- **Ideia**: diferentes componentes fazendo mesma função
  - ex. dois HDs com os mesmos dados (replicação)
- Construir sistemas tolerante a falhas com partes que falham
  - essencial em sistemas distribuídos

Usando Redundância

- Como redundância pode ajudar?

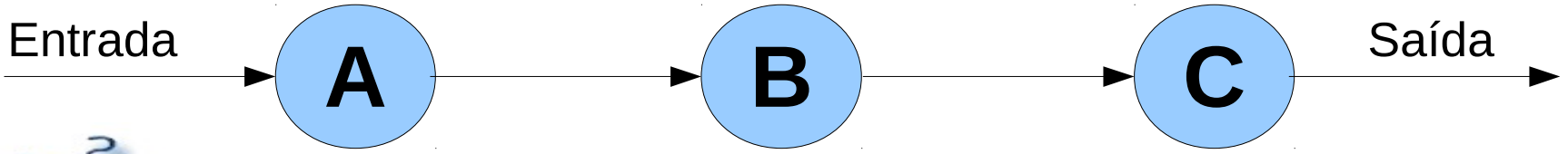
- Exemplo: 1 HD possui availability de 90%. Gostaria que availability fosse 99.9%. Quantos HDs eu devo usar?
- avalilability = fração de tempo operacional = probabilidade de encontrar operacional = p
- p_k = probabilidade de ao menos um em k estar operacional
- Assumir que falhas são independentes (uma falha não afeta a outra, forte premissa)
  - $p_k = 1 - (1-p)^k$ ← $(1-p)^k$ = prob. de todos falharem
- p_k aumenta com k, converge para 1

Exemplo

- Exemplo: 1 HD possui availability de 90%. Gostaria que availability fosse 99.9%. Quantos HDs eu devo usar?
- $p_k = 1 - (1-p)^k$
- $0.999 = 1 - (1-0.9)^k \rightarrow k = 3$
- Outro lado da moeda: probabilidade de ao menos um estar em reparo?
  - assumindo independência nas falhas
- $q_k = 1 - p^k$
- Para $k=3 \rightarrow q_3 = 1 - (0.9)^3 = 27.1\%$
- Grande fração de tempo com algum disco em reparo
  - aumenta com k

Organizando Componentes

- Três módulos em sequência (A,B,C)
  - resultado enviado do anterior enviado ao próximo

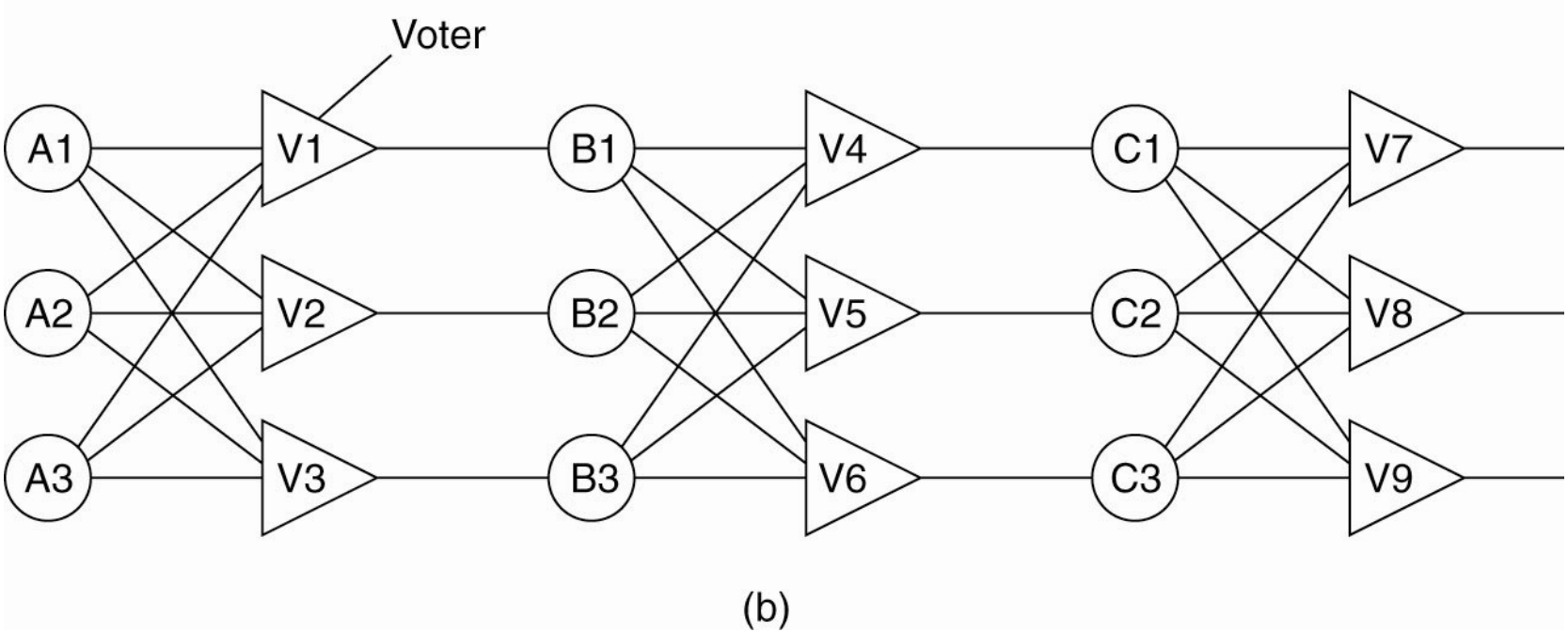


- Como organizar a redundância?

- **Ideia 0**: replicar componentes de forma independente
  - ex. três linhas em paralelo
- Operacional apenas quando não há falhas em alguma linha

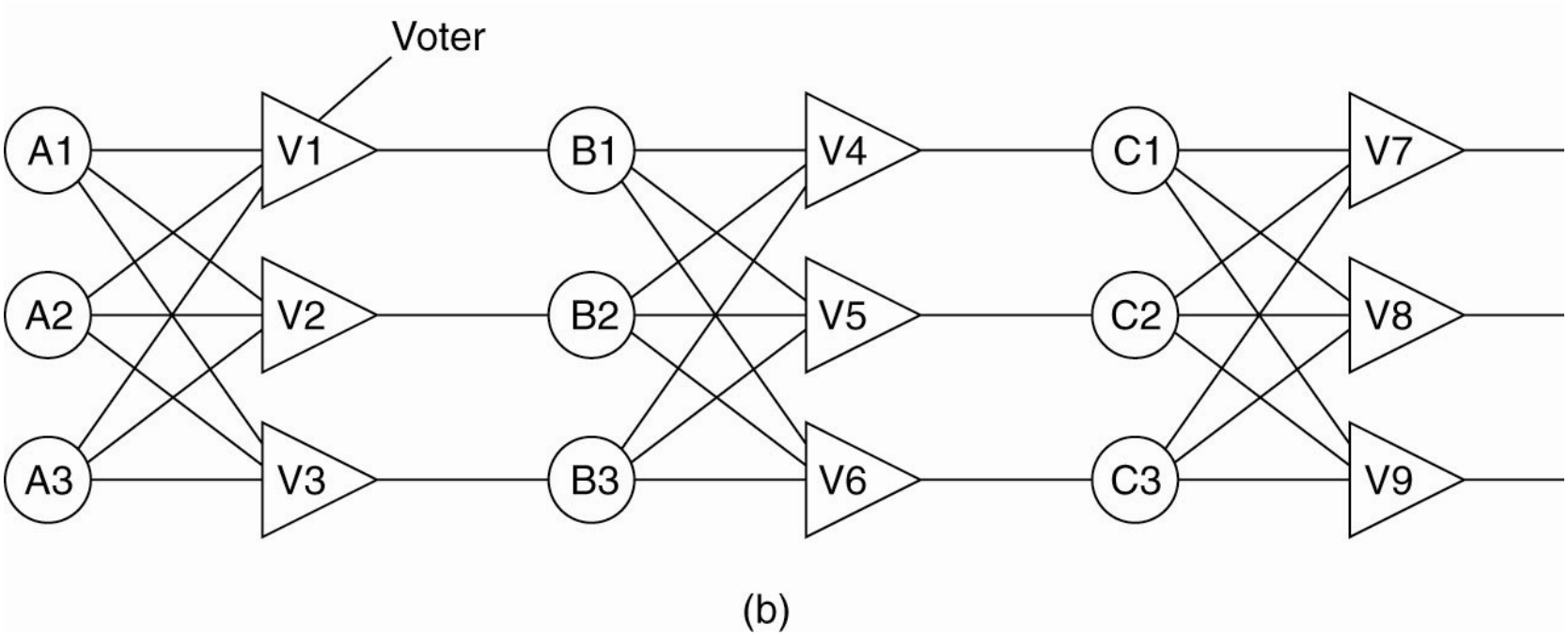
Triple Modular Redundancy (TMR)

- Organizar componentes de forma não-independente
- Votador: circuito simples, se duas ou mais entradas são iguais, repassa entrada, caso contrário nada



- O que ocorre se A1, B2 e C3 falham?
- Funciona mesmo com uma falha em todas as linhas

Triple Modular Redundancy (TMR)



- Por que temos tres votadores por nível?
  - um seria suficiente para corretude
- Votadores podem falhar!
  - redundância nos votadores
- O que ocorre se V1 e A1 falhar?
- O que ocorre de V1 e V2 falharem?

Tipos de Falhas

- Sistemas podem falhar por muitas razões
  - dependência entre as partes
- Classificação dos tipos de falha

Type of failure	Description
Crash failure	A server halts, but is working correctly until it halts
Omission failure *Receive omission* *Send omission*	A server fails to respond to incoming requests A server fails to receive incoming messages A server fails to send messages
Timing failure	A server's response lies outside the specified time interval
Response failure *Value failure* *State transition failure*	A server's response is incorrect The value of the response is wrong The server deviates from the correct flow of control
Arbitrary failure	A server may produce arbitrary responses at arbitrary times

Mais difícil de lidar!