

Data Center Basics



No Time for Downtime



Requirements

- 24 hours a day availability
- Rapid data growth
- Data dependency
- Downtime costs \$ and business survivability
- Reliability
- Availability



The cost of unavailability

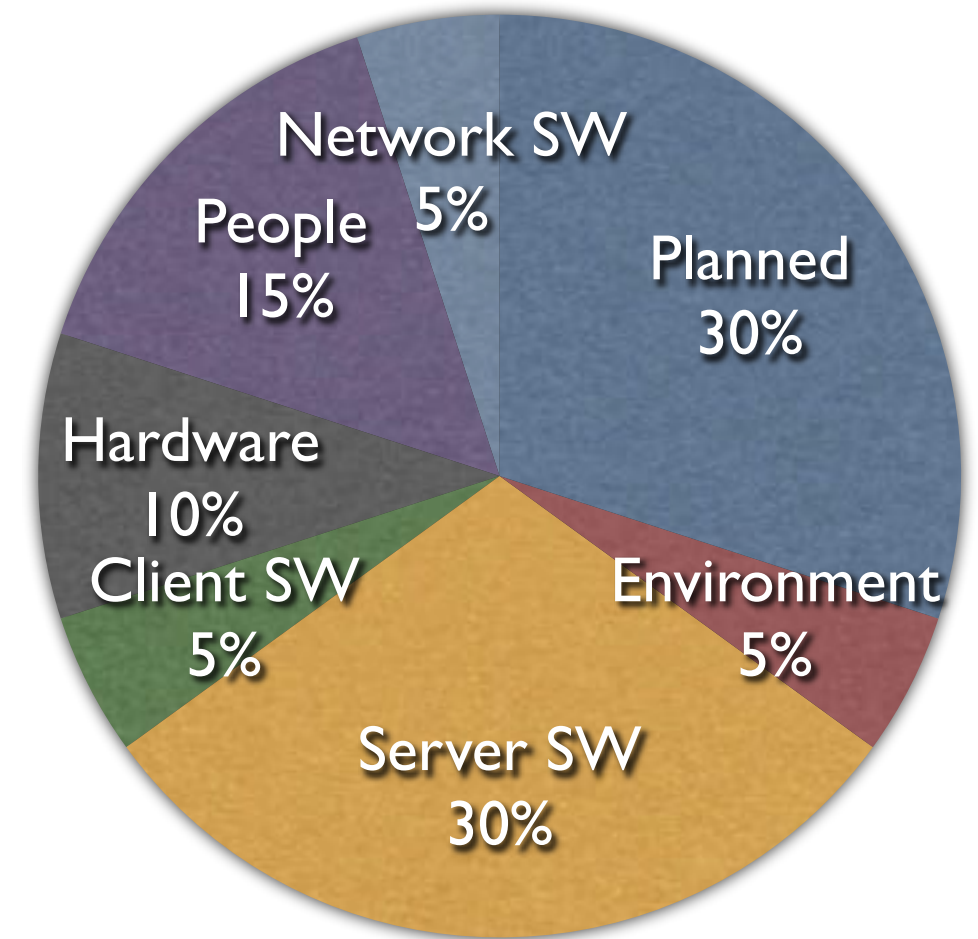
- What happens when:
 - Applications are inaccessible?
 - Data is unreachable?
 - Servers refuse to boot up?
- Can you:
 - Shutdown your business
 - Tell costumers to go elsewhere
- Have you planned for this scenario?
 - Can you recover from it?
 - How long it will take and how much will it cost?
 - What about your reputation among costumers? will they come back?



Downtime

Causes of Downtime

| Percentage uptime | Percentage downtime | Downtime / year | Downtime/ month |
|-------------------|---------------------|-----------------|-----------------|
| 98% | 2% | 7,3 D | 14H 46min |
| 99% | 1% | 3,65 D | 7H 18 min |
| 99,9% | 0,1% | 8 H 45min | 43min 45s |
| 99,99% | 0,01% | 52,5 min | 4min 22s |
| 99,999% | 0,001% | 5,25 min | 26s |



Causes of planned downtime

- Backup
- Replace or upgrade hardware
- Software or application upgrade
- Software maintenance or reconfiguration
- Operating system upgrade
- patch installation



Causes of unplanned downtime

- Extended planned downtime
- Human error
- Application failure
- Operating system failure
- Hardware failure such as disk, CPU, memory
- Incompatibility/conflict between application parameters



Remarks

- Availability:
 - Business view availability as the single metric of overall system performance
- Unplanned service outages are expensive and caused by:
 - Hardware and software failures
 - Human errors
 - Internet viruses and attacks
 - natural disasters
 - human caused crises



The HA Continuum



Definitions

- **Reliability**

- Represents the probability of a component or system not encountering any failures over a time span.

Is a measure of “not breaking down, once it is put in use”

- **Resiliency**

- The property of a component that allows it to continue with full or partial functionality after one or more faults.

Highly resilient components are able to detect and quickly compensate for faults



Definitions

- **Availability**

- measures the ability of a system or group of systems to keep an application or service up and running .

Designing for availability assumes that systems will fail, and the systems are configured to mask and recover from component or server failures.

- **Serviceability**

- is the probability of a service being completed with a given time window

- **Fault Tolerant Systems**

- Systems that have redundant hardware components and can operate in the presence of individual component failures.



Definitions

- **High Availability Clusters (HAC)**
 - Clusters of two or more nodes with a number of external interconnections such as shared disks and private heartbeat network. Are managed by special software with a goal of providing uninterrupted service despite nodes failures.
- **High Performance Clusters (HPC)**
 - Also called parallel computing clusters. Each cluster has a group of computers, tied together to work at the same time on a problem, not as backups to each other.
- **Disaster Recovery (DR)**
 - In the context of online applications, is an extended period of outage of mission critical service or data, caused by events such as fire and terrorist attacks that damage the entire facility.



HA Metrics

- **Mean Time Between Failures (MTBF)**
 - The average time interval (usually in hours) between two consecutive failures of a certain component or system
- **Mean Time To Repair (MTTR)**
 - The average length of time required to complete a repair action
- **Availability**

$$\frac{MTBF}{MTBF + MTTR} \text{ or } \frac{Uptime}{Uptime + Downtime}$$



Levels of Availability

- Depend on built-in hardware reliability
 - Redundancy in: hot swap power supplies, disks, fans
- Data Protection
 - Data is protected using RAID volumes. RAID-5 protect against disk failures but not against controller or subsystem failure
- Fault Tolerant Servers
 - Provides a fully replicated hardware design that allows uninterrupted service in the event of a component failure.
 - The system itself is a SPOF



Levels of Availability

- Server Redundancy or Clustering
 - Protects applications against any server level problems. 2 or more systems are clustered together using failover management software.
- Disaster Recovery
 - Build and maintain an off-site facility with duplicated hardware and software.
 - Many organisations may find this cost prohibitive.



Remarks

- Protecting against failure is **expensive** as is downtime. It is important to **identify** the most serious **causes of service unavailability** and build cost-effective safeguards against them
- A high degree of **component reliability**, data protection via redundant disks and adaptors, fault-tolerant servers, clustering and disaster recovery decreases the odds of service outage.



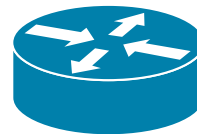
Elevado desempenho / disponibilidade



Infraestrutura para aplicações web 3-tier



Clientes



Servidor WEB



Activos de rede



Servidor de Aplicações



Base de dados



Confiabilidade



Clientes



Servidor WEB



Activos de rede



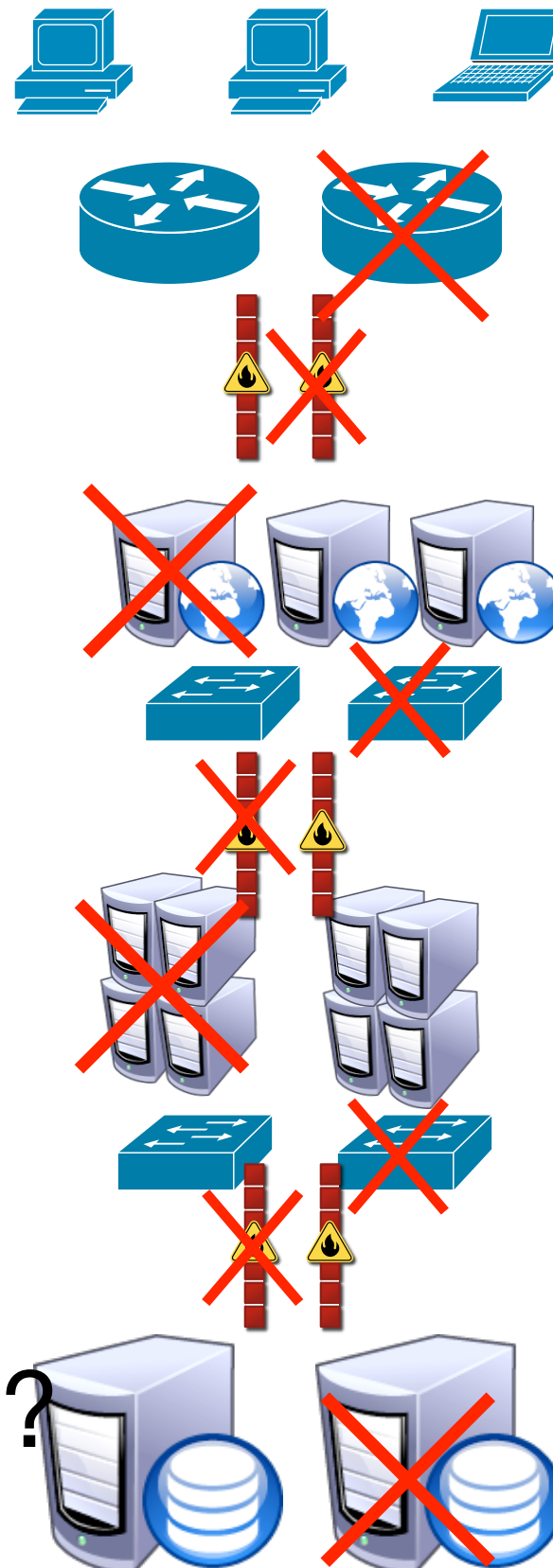
Servidor de Aplicações



Armazenamento de dados



Confiabilidade



Clientes

Servidor WEB

Servidor de Aplicações

Armazenamento de dados

Activos de rede

Como lidar com
a duplicação dos
recursos?

Quais estão activos?



Confiabilidade de Redes de Dados









Entraves à Confiabilidade da Rede

- Falhas são transientes
- A definição de rede é difusa
- Aumentos de latência podem confundir-se com falhas
- Ataques por negação de serviço (DoS)
- Balanceamento aumenta a escalabilidade não a confiabilidade
- Falha de segurança no acesso



Tipos de Falha da rede

-  Interface local
-  Cabos
-  Equipamento da infraestrutura
-  Routers e informação de routing
-  Falha de serviços fundamentais
-  Latência



Falhas Físicas

- Complicadas de diagnosticar
 - Cabo estragado
 - Falha num switch
 - Falha numa placa de rede
- É necessário identificar o ponto exacto da falha









Falhas na camada IP

- ❐ Problemas de configuração
 - ❐ Endereço errado
 - ❐ Duplicação de endereços
 - ❐ Problemas intermitentes
- ❐ Problemas de routing
 - ❐ Rotas assimétricas
 - ❐ Muitos hops entre as rotas alternativas
 - ❐ RIP timeouts de 30s



Falhas por Congestão de Rede

-  Carga de tráfego superior ao que a rede suporta
-  Tentar identificar a causa da carga
-  Tráfego ponto-a-ponto
-  Broadcasts
-  Multicast
-  DoS / DDoS



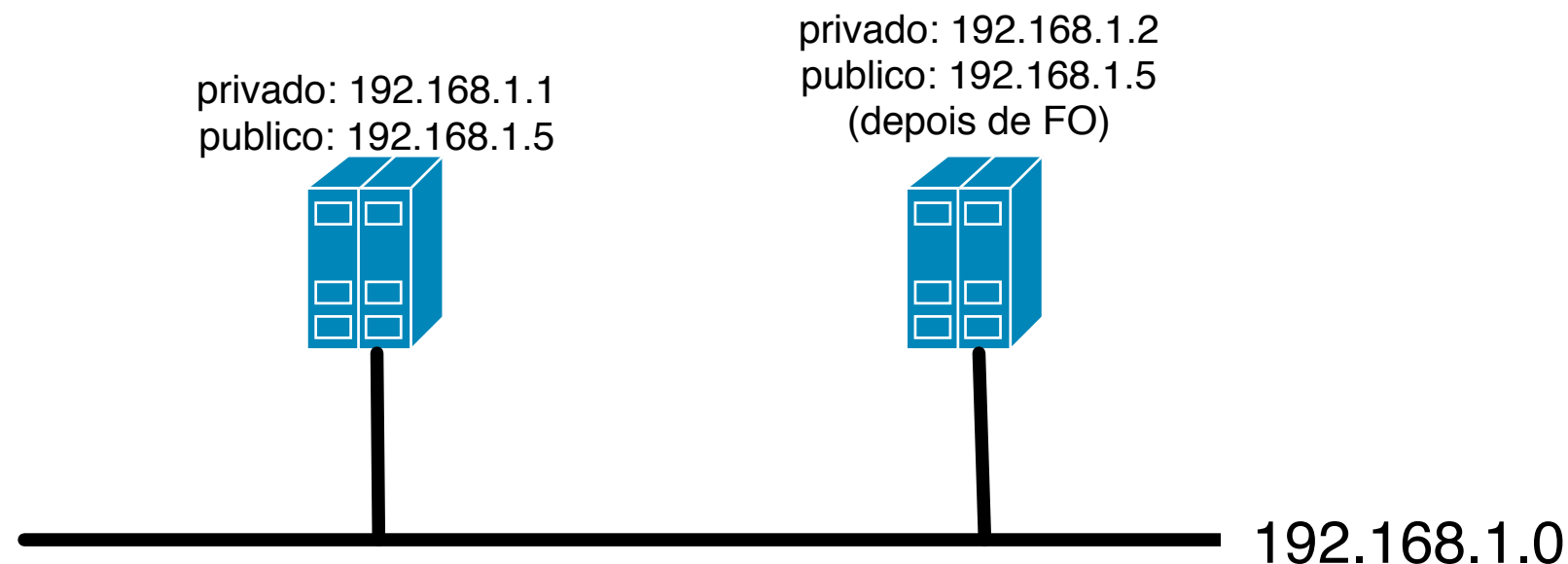
Orientações para o Desenho e Manutenção da Rede

- Conhecer a rede, os equipamentos e os protocolos utilizados
 - Monitorizar
 - Distinguir padrões de utilização
 - Saber o que pode ser medido e configurado
- Utilizar routers, filtros e firewalls para protecção
 - Permitem evitar congestionar a rede com tráfego não solicitado
- Utilização de Caches
- A separação de tráfego não garante o descongestionamento



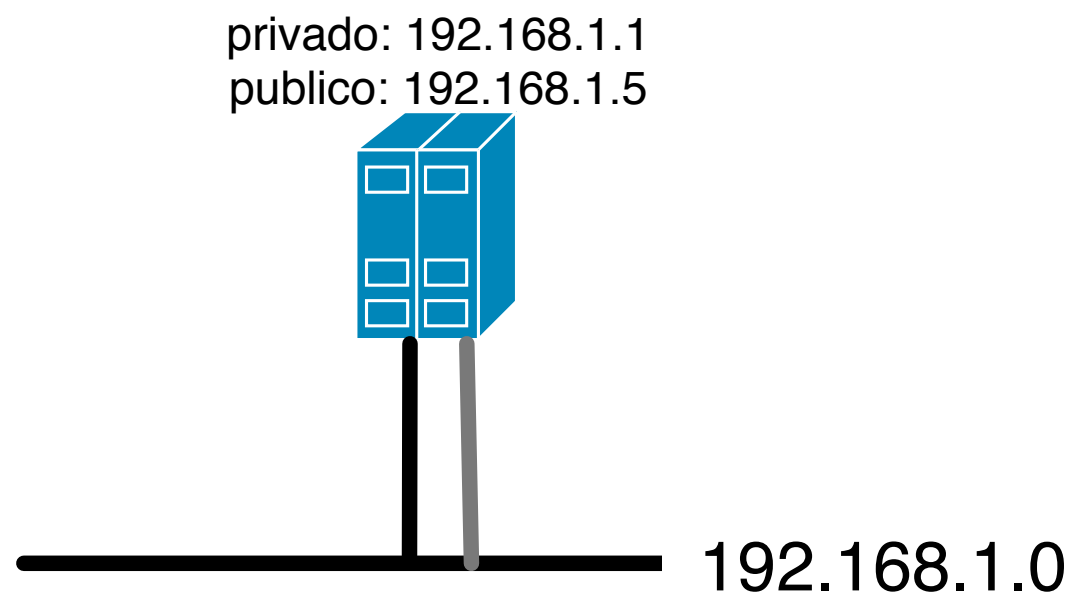
Endereços IP virtuais

- Distinguir a máquina que fornece um serviço do serviço fornecido
- Cada máquina tem um IP/nome (privado/para adm)
- Os clientes usam um mapeamento nome->IP para aceder ao serviço (público/lógico)
- Os nomes privado/público podem coincidir.



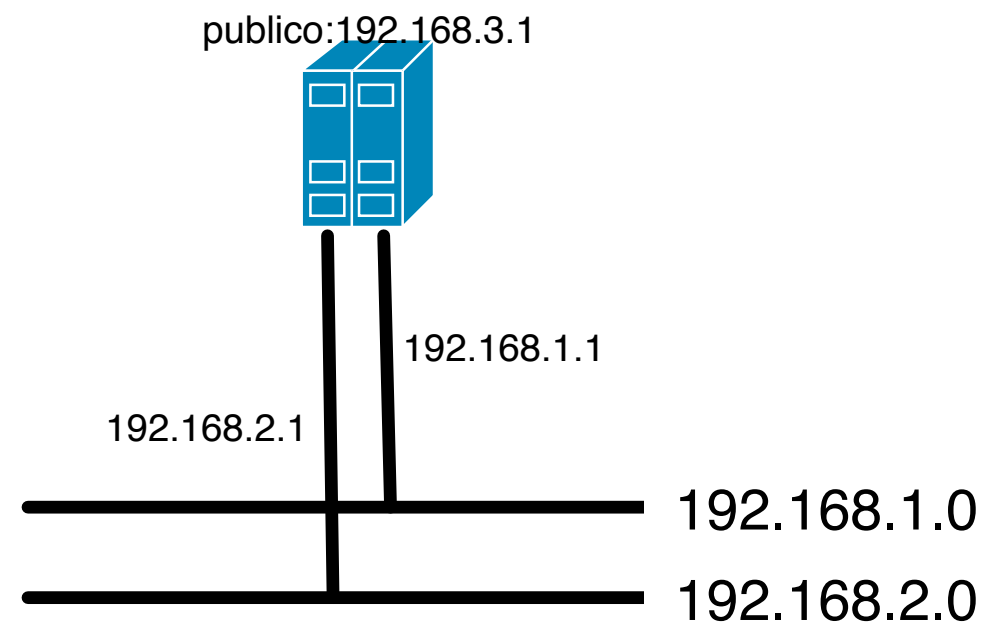
Ligações à Rede

- Redundância na ligação
- Duas placas de rede em failover
 - Se a primária falhar a secundária assume os endereços da que falhou.
- Na máquina devem estar em barramentos separados



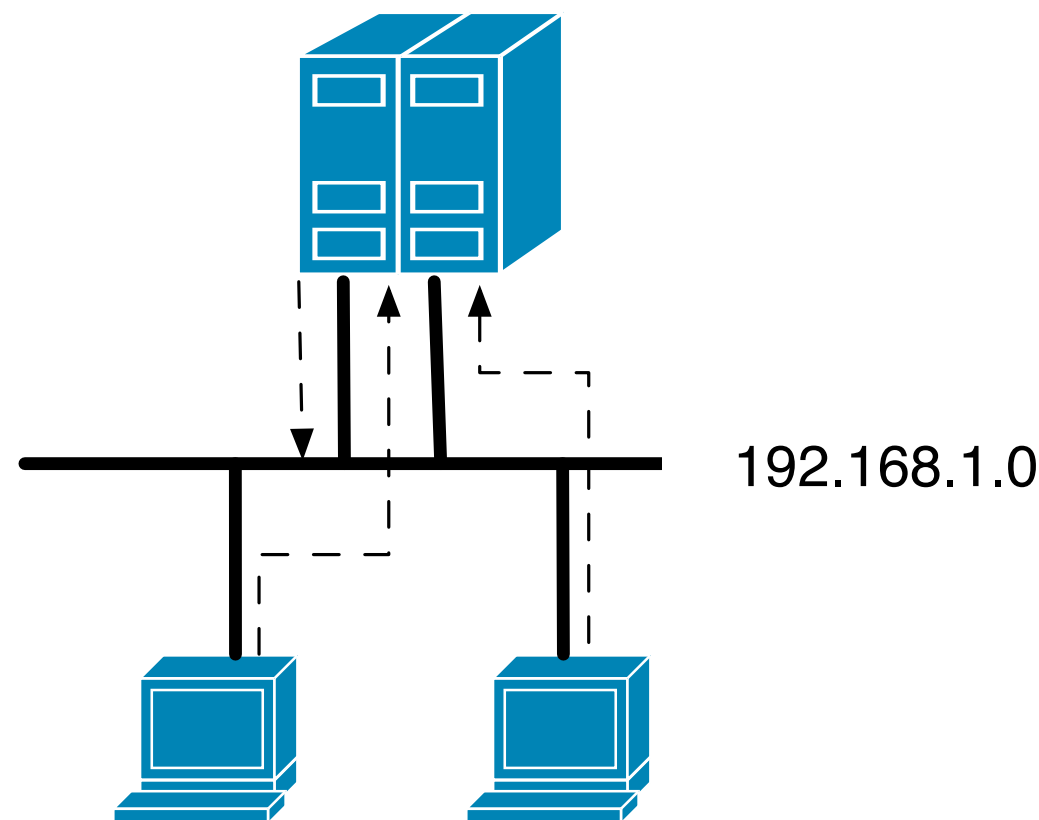
Ligações à Rede

- Ligações a várias redes
 - Migrar o ip público
 - Alterar DNS



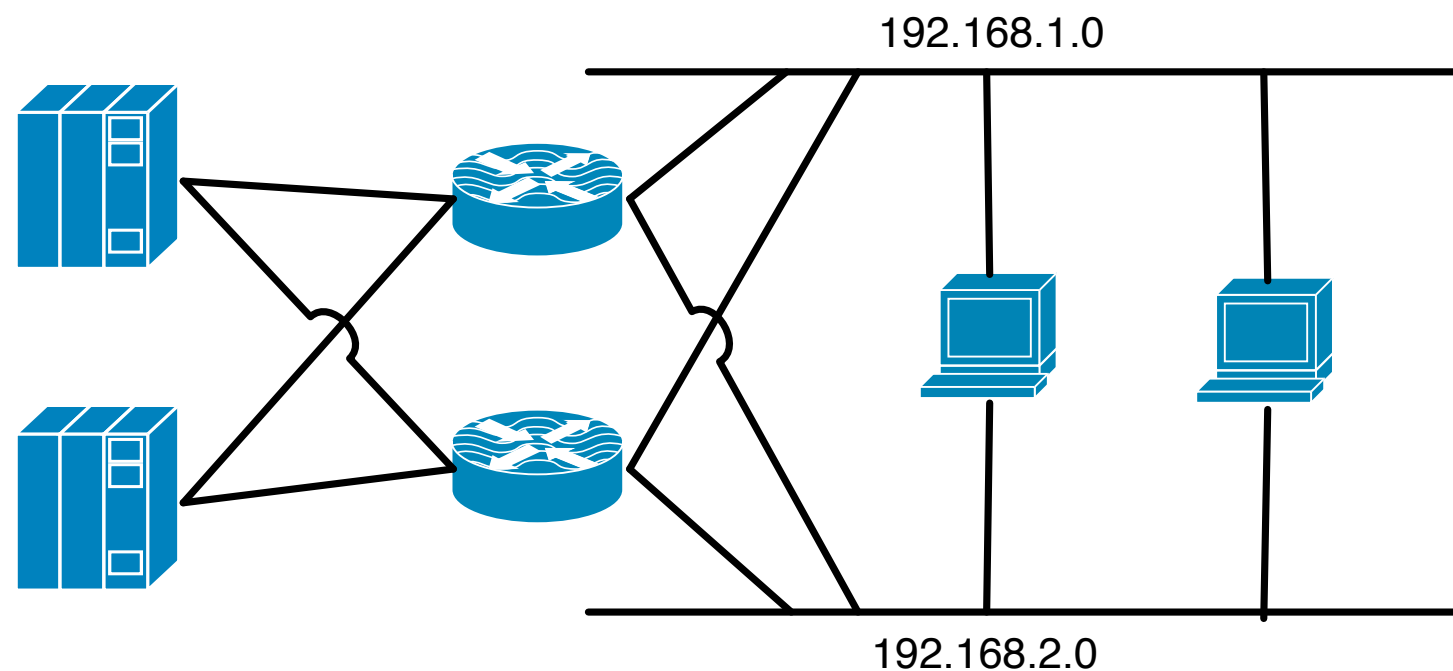
Ligações à Rede

- Bonding de interfaces
- Agregar várias interfaces numa só
- Cada interface deve ir para uma carta diferente no switch

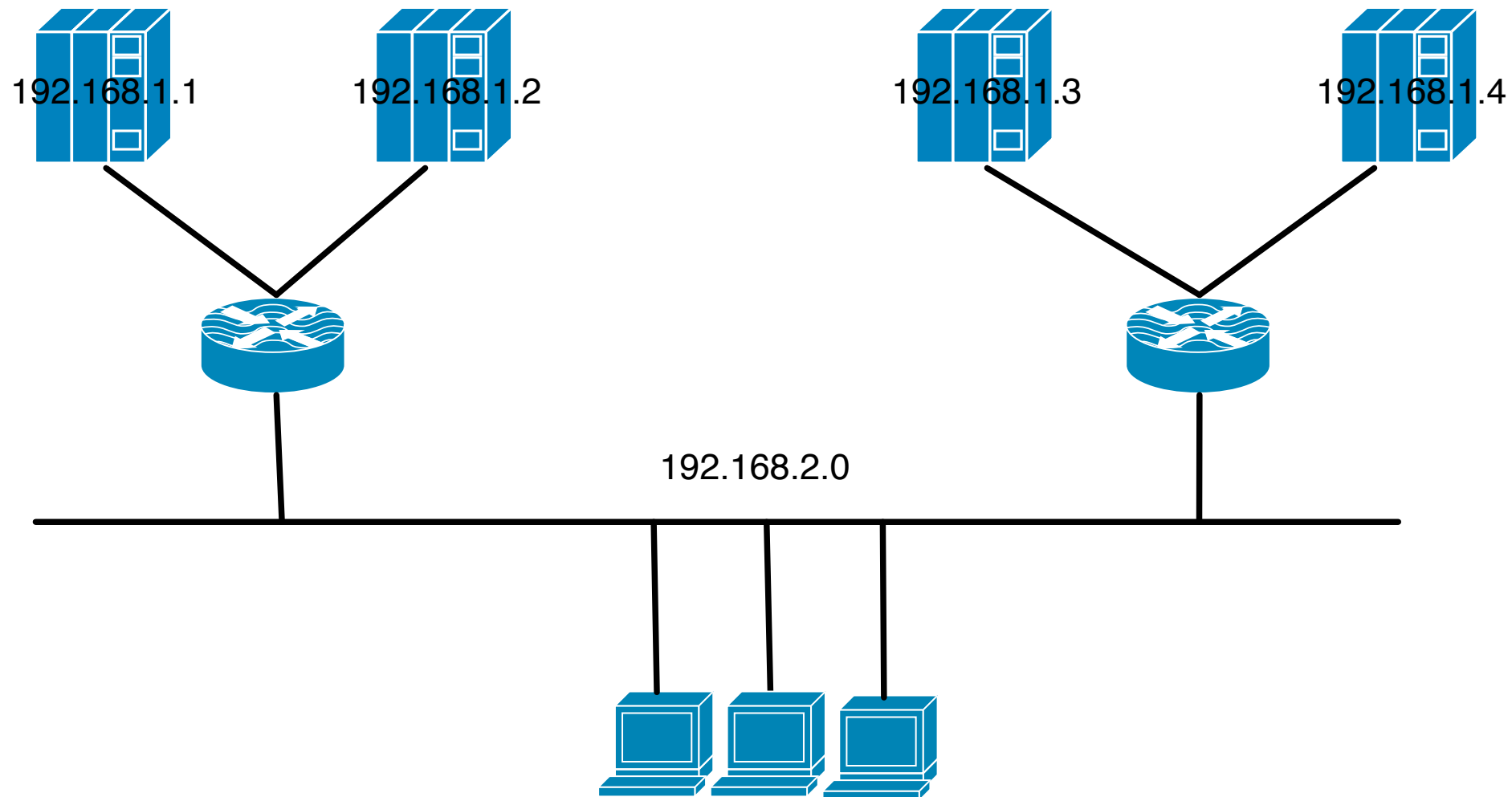


IP Network route

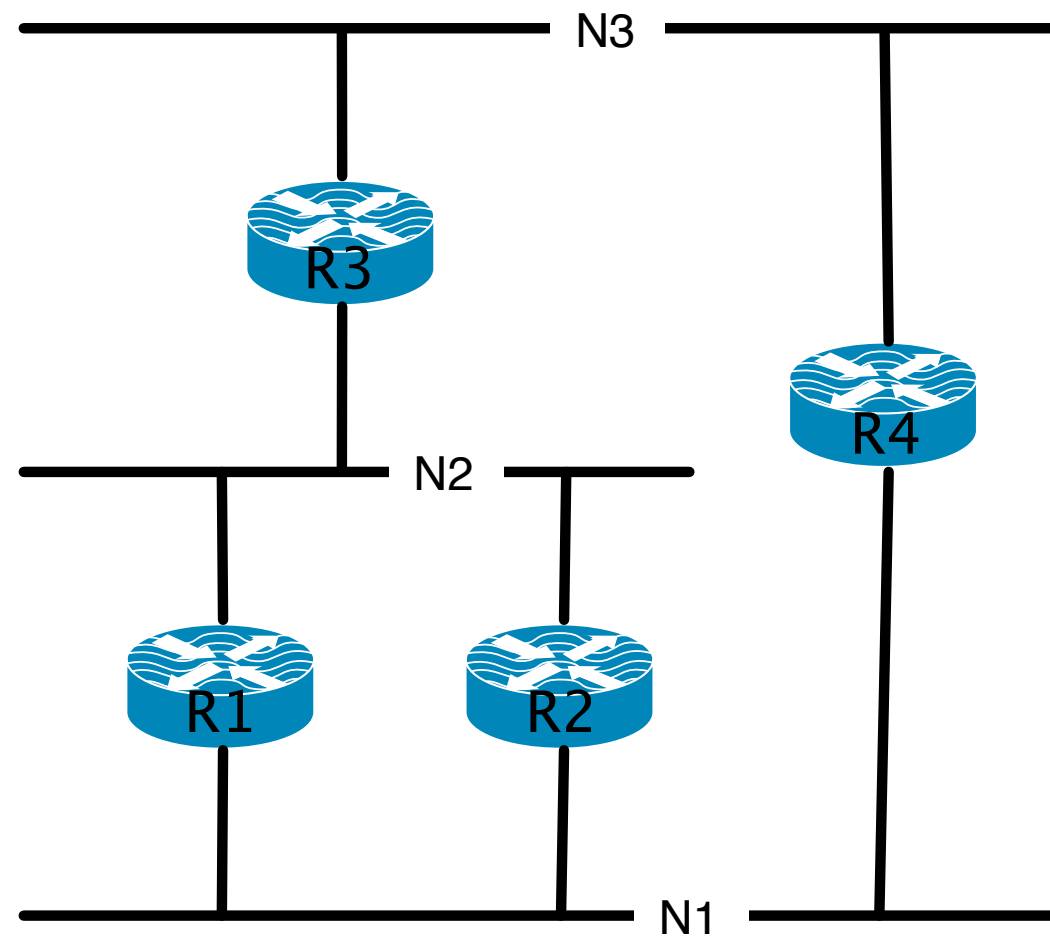
- Clientes seleccionam qual a ligação
- Só funciona com nº reduzido de clientes/servidores



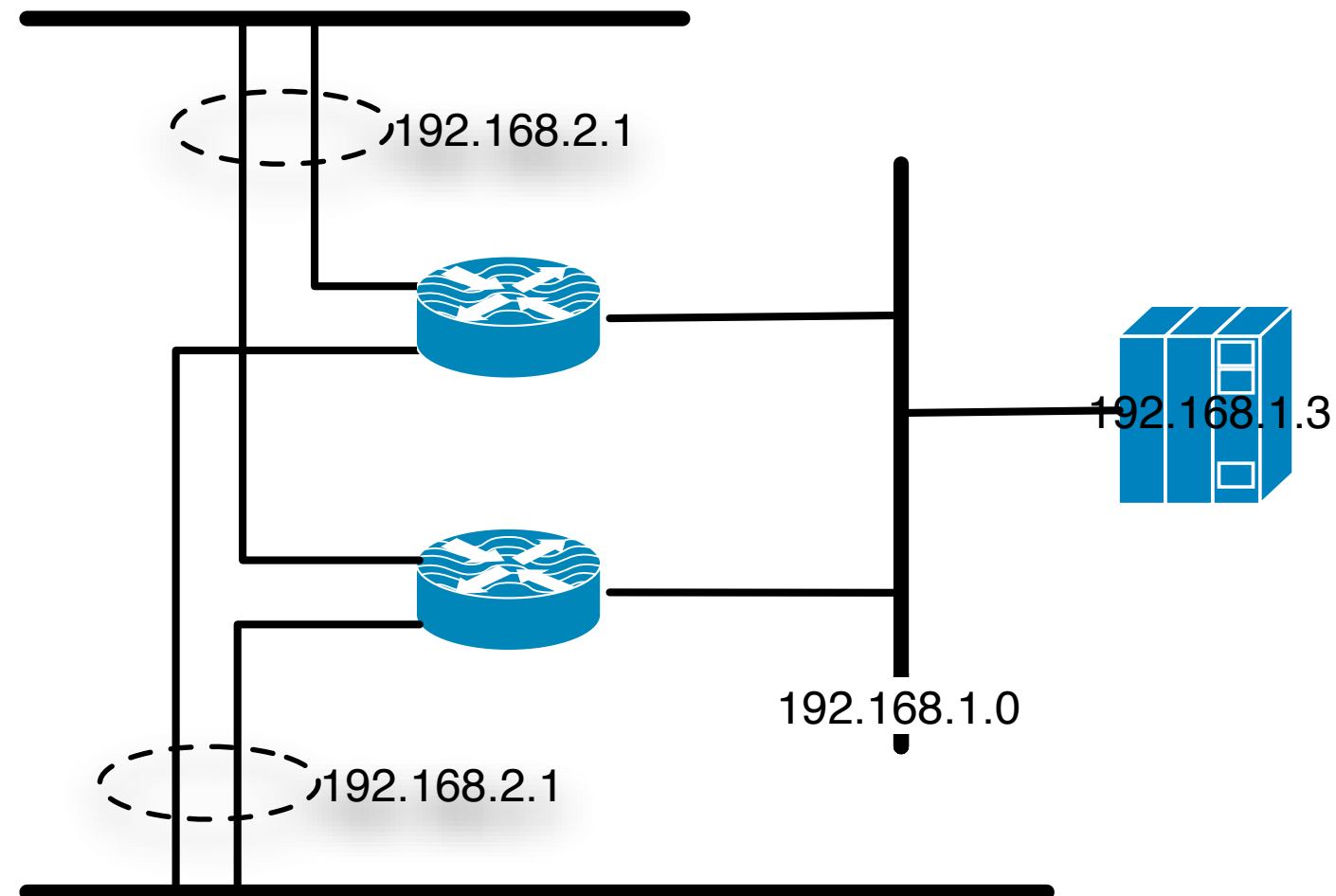
IP host route



Routing redundante



VRRP (Virtual Router Recovery Protocol)



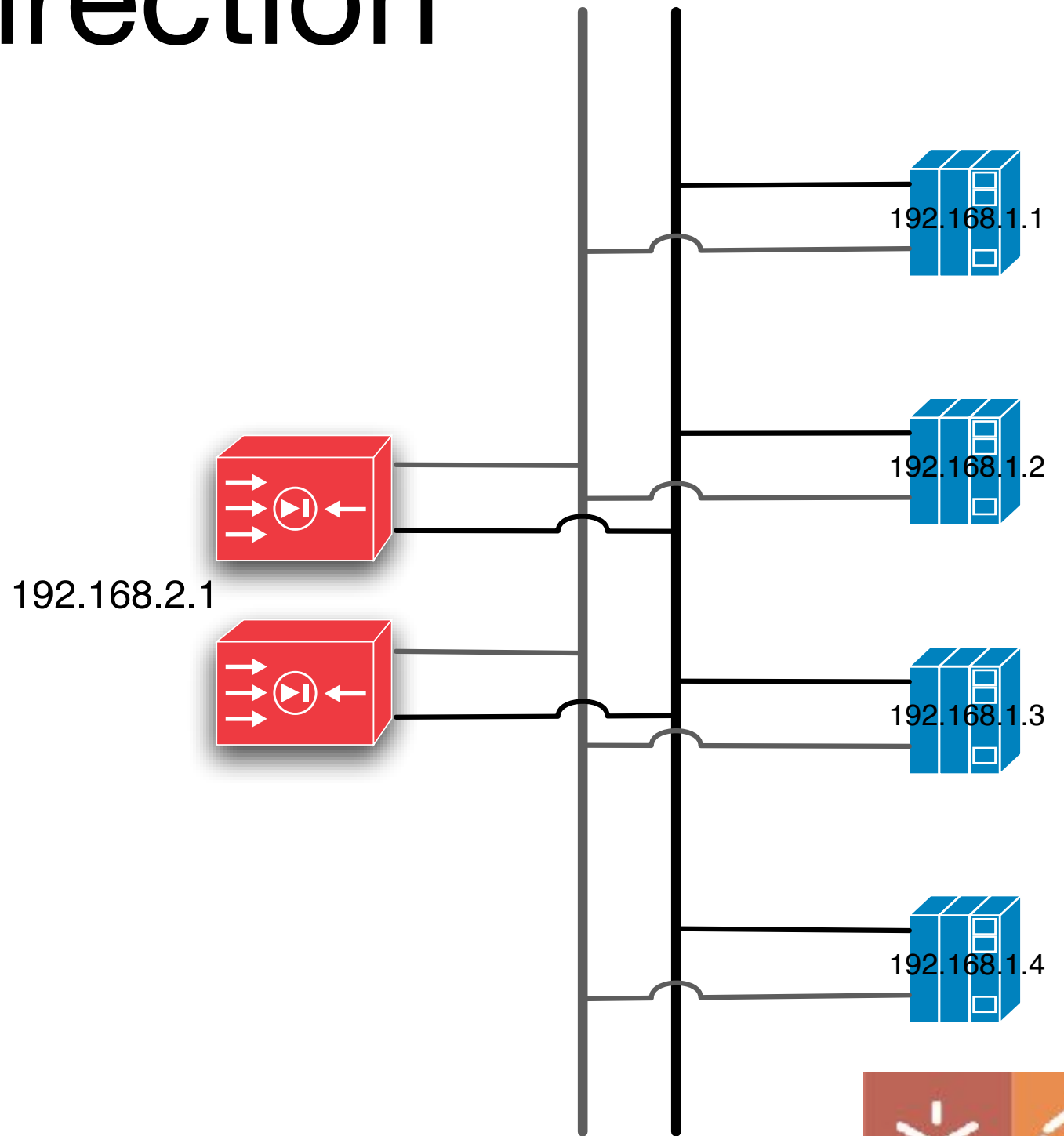
Orientações para evitar problemas de routing

- Evitar um único ponto de acesso
- Rotas por omissão desde que exista + que 1
- Redirecção dinâmica de pacotes (ICMP redirect)
- Monitorização e intervenção pro-activa



Load Balancing & Network Redirection

- Round-Robin DNS
- Network Redirector



Monitorização



Monitorização

- Permite identificar quais os servidores/serviços operacionais
- Antecipar falhas e agir pro-activamente
- Exemplo:
 - Monitorização de servidores num cluster LVS para inclusão na lista de disponíveis
- Consequência de erros de monitorização:
 - Alguns pedidos encaminhados para um servidor indisponível
 - Servidor disponível mas inutilizado



Monitorização

- Componentes físicos
 - Fontes de alimentação, ventoinhas, discos, memórias, periféricos, placas,....
- Componentes lógicos
 - Sistema operativo
 - Processos
 - Aplicações



Algoritmo genérico

- Periodicamente, cada monitor obtém uma resposta do monitorizado:
 - o monitor faz periodicamente pedidos
 - o monitorizado toma a iniciativa periodicamente
- Se não é obtida qualquer resposta durante um intervalo de tempo superior ao período de monitorização, o monitorizado é considerado indisponível
- Transições de estado desencadeiam acções
 - Incluir/remover servidores da lista de disponíveis



ICMP Ping/Echo

- O pedido é um pacote ICMP (ping)
- A resposta é enviada pelo kernel (echo)
- Verifica que:
 - a máquina está ligada
 - a rede está fisicamente ligada
 - a rede está configurada correctamente para pacotes ICMP
 - o kernel atende interrupções de hardware





Atenção !

- A existência de resposta ao pedido de monitorização apenas verifica a disponibilidade dos componentes utilizados na transmissão e processamento do pedido de monitorização!
- É frequente um computador responder a ICMP ping mesmo quando não está em condições de efectuar trabalho útil!



UDP Echo

- O pedido é um pacote UDP para a porta 7
- A resposta é enviada pelo xinetd
- Verifica que:
 - o xinetd está a correr
 - o escalonamento de processos está a funcionar
 - a rede está configurada correctamente para pacotes UDP



UDP Heartbeat

- Usa um servidor próprio, o pedido pode ser implícito
- Funcionamento:
 - periodicamente é enviado um pacote para o monitor
- Usando multicast ou broadcast, a resposta pode ser enviada em simultâneo para vários monitores








Cuidado!

- Se a máquina ou a rede estão sobrecarregadas mas não indisponíveis, a monitorização pode errar
- É preciso balancear:
 - rapidez de detecção de indisponibilidade
 - probabilidade de cometer erros
 - impacto do erro cometido



Monitorização de aplicações

-  Os problemas ao nível da aplicação são uma das principais causas de indisponibilidade
-  A monitorização a baixo nível é incapaz de detectar a indisponibilidade de aplicações
-  A monitorização de aplicações não pode ser feita apenas com mecanismos genéricos



HTTP HEAD

- O pedido é o método HEAD sobre um ficheiro conhecido (por exemplo, /index.html)
- A resposta são os cabeçalhos, incluindo a data de modificação
- Verifica que:
 - o httpd está correctamente configurado e a correr
 - o disco em que o ficheiro está armazenado está (esteve?) disponível



HTTP GET

- O pedido é o método GET sobre um ficheiro conhecido (por exemplo, /index.html)
- A resposta são os cabeçalhos e os dados
- Verifica que:
 - o conteúdo do ficheiro está correcto, por exemplo, o título é o esperado ou o HTML bem formado



HTTP POST

- O pedido é o método POST sobre um objecto conhecido (por exemplo, /index.cgi)
- A resposta são os cabeçalhos e os dados
- Verifica que:
 - a aplicação que suporta o objecto está disponível (por exemplo, que existe espaço em disco para armazenar o estado decorrente do pedido)
- Dependendo do tipo de pedido, pode testar-se a aplicação a diferentes níveis de profundidade





Cuidado!

- A monitorização não deve interferir no funcionamento normal do serviço:
- em algumas aplicações, a execução de pedidos de monitorização que verifiquem todos os componentes do sistema podem ser perigosos
- a existência de numerosos pedidos de monitorização pode perturbar o funcionamento do sistema






Dependências

- É importante considerar dependências entre objectos monitorizados
- Exemplo:
 - um sistema constituído por um director, uma rede local (switch), vários servidores e serviços
 - a detecção da falha de um servidor deve desencadear a sua remoção da tabela de encaminhamento
 - a detecção da falha do switch deve desencadear a reconfiguração para um serviço alternativo



Estratégias para a monitorização

-  Uma máquina monitoriza-se a si própria
-  Uma máquina monitoriza outras máquinas
-  Várias máquinas monitorizam-se a si próprias



Local

- Monitor e monitorizado na mesma máquina
- Útil para detectar:
 - estouro da aplicação
 - bloqueio da aplicação (por exemplo, por esgotamento de um recurso)
- Não é eficaz quando:
 - o problema é ao nível do sistema operativo
 - o recurso esgotado é necessário para o funcionamento do monitor



Watchdog

- Hardware especializado que automaticamente reinicia a máquina em caso de indisponibilidade
- Funcionamento:
 - escrever algo periodicamente para `/dev/watchdog`
 - depois da primeira escrita, se o período é ultrapassado a maquina é reiniciada
- A escrita pode depender da monitorização de uma aplicação



fping

- Para monitorização de outra máquina
- Monitorização usando ICMP ping/echo de um conjunto de máquinas
- Funcionamento:
 - envia um conjunto de pings
 - espera até receber todos os echos ou até esgotado o limite de tempo
 - imprime o estado de cada uma das máquinas



Linux-HA mon

- Monitorização de aplicações com módulos para:
 - desencadear acções quando descobre indisponibilidade
 - detectar indisponibilidade ao nível da aplicação
- Vocacionado para avisar administradores e não para desencadear automaticamente acções de recuperação
- Alternativas:
 - monit, Big Brother, Nagios



ldirectord

- Para utilização num *director* LVS
- Monitorização ao nível da aplicação de serviços http, https e ftp
- Actualização automática das regras com ipvsadm quando é descoberta a indisponibilidade de um servidor



Linux-HA Heartbeat

- Para monitorização mútua
- Produz periodicamente pacotes em UDP multicast ou broadcast
- Produz periodicamente pacotes em ligação série
- Reencaminha pacotes recebidos em ligação série para formação de aneis
- Executa scripts em resposta à descoberta de máquinas indisponíveis



Disco partilhado

- Um sector de disco por cada máquina onde é guardado um inteiro
- Cada máquina periodicamente:
 - lê todos os sectores
 - escreve o seu sector, depois de incrementar o inteiro
- Se depois de um intervalo de tempo algum dos contadores pára, admite-se que a máquina está indisponível
- Vantagem: o recurso é o próprio meio de monitorização



Split-Brain

- Se ao fazer monitorização mútua o sistema de monitorização falha, cada uma das máquinas considera a outra indisponível
- As soluções possíveis são:
 - um mecanismo de monitorização redundante (por exemplo, porta série + rede)
 - mitigar as consequências do erro



Coerência

- Monitorização para escolha de um líder
- Exemplo:
 - Escolher qual o servidor num cluster que pode escrever num disco partilhado
- Consequências de erros de monitorização:
 - nenhuma dos servidores escreve no disco (tornando o serviço indisponível)
 - ambos os servidores escrevem no disco, corrompendo os dados!!!!







I/O Fencing

- Em situações em que está em jogo a coerência de dados, não se pode confiar na monitorização simples
- Mecanismo genérico:
 - permitir que uma máquina impeça outra de aceder a um recurso comum (e.g. um disco)
- Possível com alguns tipos de interliagação



STOMITH / STONITH

-  Mecanismo simples e genérico
-  Shoot **T**he **O**ther **M**achine / **N**ode **I**n **T**he **H**ead
-  Cada máquina controla a fonte de alimentação de outra
-  Após uma suspeita de indisponibilidade, a máquina é desligada

