Topic: Load balancing
High Availability

- Key property: availability
  - “high” is relative
  - Important to determine correctly
    - Cost of unavailability
    - Business continuity: how long can one accept unavailability?
- Key concept: single point of failure: SPoF
- SPoF examples
  - Network (one ISP, one network card, one switchboard, ...)
  - Power (one switch, one provider, ...)
  - Physical access (one door, one gate, one road ...)
  - People (only one person knows how to ...)
- Principle:
  - “No single point of failure”
- Major method to achieve: redundancy
HA: active - passive

- Capacity: 100%
  - Passive
    - Hartbeat link
      - Active
        - Capacity: 100%
          - 100% overcapacity
          - 100% OK after switch
  - Capacity: 100% overcapacity
  - 100% OK after switch
- Capacity: 70%
  - Active
    - Capacity: 70%
      - 40% overcapacity
      - 70% OK after switch
    - Load balancing
HA: crosses
HA: Two “trains”/”streets”
Disaster recovery – business continuity

Disaster Recovery

• Respond to disasters
  – System down
  – Disk crash

• Measures:
  – Recovery point objective
    • the maximum time period in which data might be lost
    • “lose 1 hour of data”
  – Recovery time objective
    • The amount of time the business can be without the service, without incurring significant risks or significant losses
    • “system unavailable for 1 hour”

• Options
  – Back-up/restore

Business continuity

• Major outage
  – IT site unreachable, destroyed

• Survival mode: minimal functions to survive, able to continue business

• Provide critical operational support

• Options:
  – Physically separated sites
  – Rented locations
  – Home working

• Types:
  • Hot standby: up and running in no time
  • Cold standby: capability
  • Infra only: deploy when needed
Back-up - archiving

**Back-up**
- Use: protection against data modification
  - Delete
  - Update
- Redundant copy
- Full – incremental
- Mostly On-line
- Use: restore

**archiving**
- Use:
  - Reduce on-line storage needs
  - Record keeping (requirement)
- Master data
- Incremental
- Mostly off-line
- Use: search
Load balancing

• Solve resource/capacity problem
• One system cannot cope with load
  – Put bigger system in place
  – Put a cluster in place
  – Use multiple systems: load balancing
Load balancing

Virtual server

client

LB

Server 1

Server 2

server3

Server 4

4/03/2014
Load balancer

- Hides the fact that there are multiple servers
- Presents itself as THE server (virtual server)
- Determines “best” system to handle
  - “Best” system to handle requests based on request and system information
    - Expected load
    - Current load of systems
- But: may not look at either, and count on statistical balancing (round robin)
Load balancer: dealing with stickyness

• Servers may need to get multiple requests to not brake the transparency, or just plain work

• Issues:
  – Sessions, state

• Example: IP packet of one TCP session must go to same server in a standard implementation

• http sessions must be handled by the same server for vanilla web servers
Sticky sessions

First req

Next req

First req

Next req
State in client – stateless operation

First req

Next req
Client state

• Need to communicate all of the state each time
  – Overhead
  – size limitations
  – Multiple requests to different servers => independent state management possible (possible conflicts)

• Some state data is critical for server security
  – Logged-in user, authorizations

• State protection:
  – Protect confidentiality, integrity
  – Encryption, one key to manage them all?
Load balancing

• Servers have resource limitations

• Objectives:
  – Scalability
    • Servers have resource limitations, a single server has maximal load points
    • OS level (multiprocessor, clusters): expensive
    • Add or remove additional servers transparently
  – Availability
    • Replace defective components: only subset of users impacted
    • But not High-Availability (HA): other measures required

• Secondary
  – Transparency
    • “Virtual server”
Load Balancing solutions

• Effective load balancing:
  – Measure availability
    • Health monitoring
    • System availability: ping
    • Service availability: service ping
  – Measure “load”
    • Load = resource usage: CPU, disk, ...

• Transparency
  – Server side state
    • Managed on a server instance:
      – requires persistence
      – Each packet /message of connection to same server
      – Problem with higher level sessions (higher than LB mechanism)
    • State distribution
      – Central state store (new problem point)
      – Cost of state migration
General principle

Map:
C-S: VS-RS

C1-S: VS-Rsa
C2-S: VS-RSb
Load balancing systems

• Four main hooks for network load balancing:
  - ARP
    • Different MACs for single IP
  - NAT
    • Different mappings for same IP
  - DNS round robin
    • Different IPs for same DNS name
  - Proxy
    • Different back-ends for same proxy request

• Many vendors of solution: Alteon, F5, Cisco, Resonate, Radware, ...
ARP

• IP address: map one IP to multiple MAC addresses
• Keep track of sessions: which MAC for which TCP session
• Note: optimization
  – Short requests, long replies: only request via load balancer, replies: direct
• Examples: Cisco local director, F5 BIG IP
ARP: roundtrip

9.9.9.9 -> 34.1.1.1
M_{Rout} -> M_{LB-L}
M_{LB-L} 34.1.1.1

9.9.9.9 -> 34.1.1.1
M_{LB-R} -> M_{sysi}
M_{sysi} 34.1.1.1

9.9.9.9 <- 34.1.1.1
M_{rout} <- M_{sysi}
ARP: routing

Arp: 34.1.1.1 – MAC LB

Client

Router

34.1.1.254

34.1.1.1

-> 34.1.1.1

34.1.1.1

“noarp”

34.1.1.3

34.1.1.100

LB

34/03/2014

(c) A. Mariën
NAT based load balancing

- Virtual IP incoming
- Translate to multiple internal IP addresses
- Keep session state
- FW-1, BIG IP
NAT: roundtrip

- Set up client connection
- Decide back-end system
- Translate all packets
DNS round robin

- Do not allow caching of DNS – IP mapping
- Hand out IPs in round-robin mode
- Statistical loadbalancing
- Cisco Distributed Director, 3DNS
Proxy

• Connections forwarded to different IPs
• HTTP reverse proxies
  – Host: a = server a/ Host: b = server b
  – Map:
    • /images: http://a/images
    • /doc: http://b/doc