Výkon enterprise aplikací, škálovatelnost, vysoká dostupnost

Základní koncepty

- pro některé aplikace je kritická určitá úroveň dostupnosti, konzistence, výkonosti apod.
- např. pro banky, web mail, částečně i KOS/CW

- **Mission-critical application**
  - aplikace nezbytná pro přežití business/organizace
  - selhání/porucha by na ně měla značné dopady
  - jsou pro ni důležité:
    - **škálovatelnost** – Jak dobře se adaptuje na zvládnutí většího množství práce?
    - **dostupnost** – Jak moc poskytuje užitečné zdroje za jednotku času?
    - **výkon** – Jaká je míra zpracování dané zátěže během dané jednotky času?

Škálovatelnost

- definuje jak lehce se dá aplikace rozšířit, aby splňovala zvyšující se nároky na síť, processing, přístup k DB, file-systém apod.
- jak dobře zvládá zvyšující se množství práce
- jsou dvě možnosti škálování:
  - **horizontální (scaling out)** = přidání nových nodů (strojů) se stejnou funkcionalitou jako ty současné
  - **vertikální (scaling up)** = přidání procesorů, paměti, úložiště atd. k nodu

Dostupnost

- **Uptime (downtime)** = čas, kdy aplikace běží (neběží)
  - občas se používá k vyjádření pravděpodobnosti

- **dostupnost** = procento času, kdy aplikace nabízí požadovanou funkcionalitu

$$A = \left(1 - \frac{t_{unplanned\_downtime}}{t_{uptime}}\right) * 100$$

- **high-availability** – aplikace, které musí mít skoro 100% dostupnost

- 90% = „one nine“, 99% = „two nines“, 99.9% = „three nines“,…

Service Level Agreement (SLA)

- definuje závazky zúčastněných stran týkající se dodávání a užívání aplikace, např.:
  - minimální/cilená úroveň dostupnosti
  - maintenance windows
  - výkonnost a metriky pro její vyhodnocení
  - účtování
  - důsledky nedodržení závazků

Techniky

Load balancing

*** response time** = čas, který systém potřebuje ke zpracování requestu po jeho obdržení

*** latency** = response time – (server) processing time

*** throughput = propustnost** = počet transakcí / vteřinu, které aplikace může zvládnout

*** load balancing =** technika pro minimalizaci response time a maximalizaci propustnosti, deleguje requesty mezi vícero nodů

*** load balancer =** odpovědný za směrování requestů k dostupným nodům dle plánovacích pravidel

- distribuuje požadavky klienta či zatížení sítě efektivně mezi několik serverů
- existují HW i SW load balancery
- **Round Robin Load Balancer** – distribuuje požadavky serverům sekvenčně
- **Least connections** – požadavky jsou směrovány k serverům s nejmenší zátěží
- **IP hash** – IP adresa klientova požadavku rozhodne o cílovém serveru
- **Persistent/Sticky Session** - stavové aplikace se server-side session vyžadují, aby požadavky z jedné session chodily na stejný server

- časté vlastnosti load balancerů:
    - **asymetrická distribuce zátěže** – různá zátěž přiřazena různým nodům
    - **prioritní aktivace** – při příliš vysoké zátěži se aktivují některé standby nody
    - **dynamická konfigurace** – přidání/odebrání serverů ze server-poolu rychle a za běhu
    - **filtrování obsahu** – upravuje procházející traffic
    - **firewall** – rozhoduje, jestli traffic projde nebo ne, na základě bezpečnostních pravidel

Caching

- technika pro sdílení dat mezi vícery spotřebiteli dat
- dobré, jsou-li data náročná na výpočet nebo se příliš nemění
- implementace pomocí indexovacích tabulek
    - pomocí klíče dostaneme cacheovaný objekt (datum)
    - dotaz na datum může vést ke cache hit (ok) či cache miss (not ok)
- cache je pro klienta transparentní

Typy cache

- **application cache**
- **web cache**
    - client side (prohlížeč) vs. server side caching
    - proxy cache – obsluhuje request pro klienty přistupující ke stejnému zdroji
- **distributed cache** – multiple systems, multiple customers, multiple resources

Cache strategie

- **Read-through** = data se čtou přes cache; miss → čtení z úložiště a uložení do cache
- **Write-through** = zapisují se se přes cache; aktualizace je synchronní v cache i v úložišti
- **Write-behind** = zapisovány do cache; aktualizace v úložišti pak probíhá asynchronně
- **Write-allocate/No-Write-Allocate**

Cache eviction

- **Index-based** = smazání cache na specifickém indexu
- **Random, Round Robin** = smazání na náhodné/vypočtené pozici
- **FIFO (TTL)** = nahrazení nejstaršího (nehledě na frekvenci)
- **LRU** = least recently used

Clustering

- **cluster** = skupina výpočetních systémů, které spolupracují, ale z uživatelského hlediska se jeví jako jeden systém

- **Load-balancing cluster (Active/Active)** – distributes load to redundant nodes, while all nodes are active at the same time offering full-service capabilities

- **High-availability cluster (Active/Passive)** – improves service availability by redundant nodes eliminating single points of failures

Principy k dosažení větší dostupnosti

- **Elimination single points of failure** – přidání nadbytečnosti, takže selhání komponenty nevede k selhání celé aplikace
- **Reliable crossover** – schopnost přepnutí ze selhávajícího nodu na nový beze ztráty
- **Detection of failures as they occur**

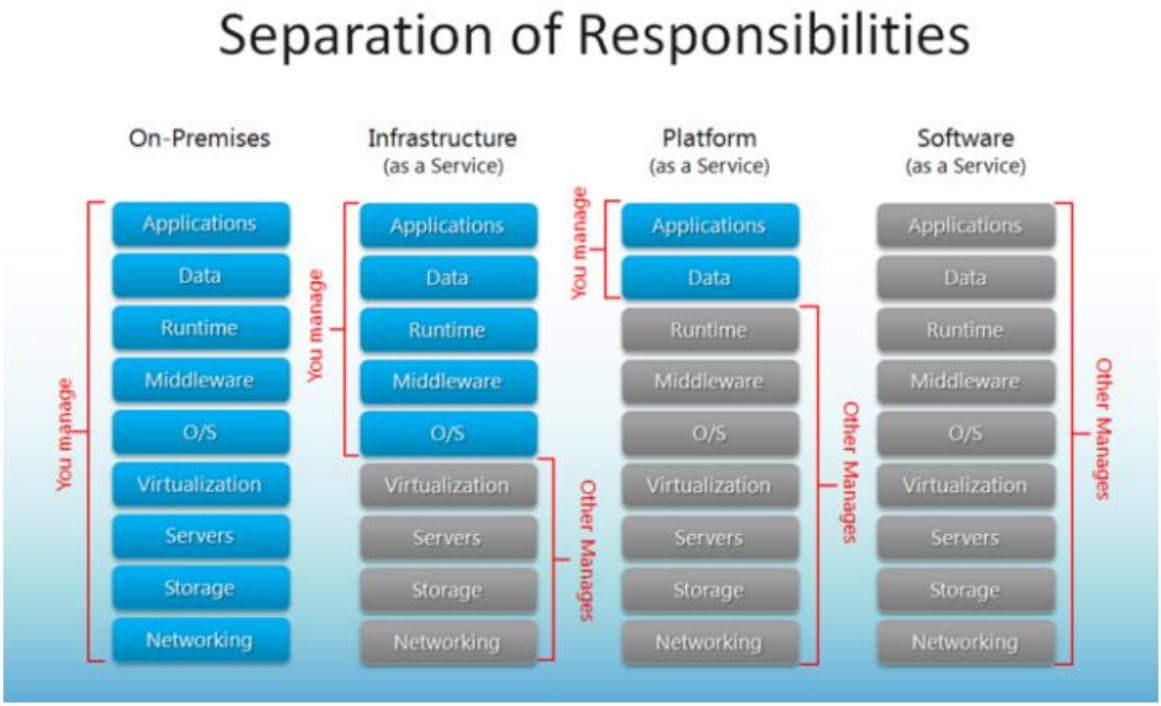
Cloud Computing

- typ internetového výpočtu, kde aplikace běží na distribuovaných resources spravovaných třetí stranou
- Pay-as-you-go billing

- service models:
    - **IaaS = Infrastructure as a Service**
        - využití nabízené infrastruktury – virtuální stroje, servery, sítě,…
        - např. Amazon EC2

    - **PaaS = Platform as a Service**
        - využití služeb/knihoven/nástrojů apod. poskytovatele
        - kontrola nad deploynutou aplikací (spouštění, DB, web-server, vývoj,…)
        - např. Google AppEngine, MS Azure

    - **SaaS = Software as a Service**
        - využití aplikace poskytovatele
        - slabá kontrola nad aplikací
        - např. Office 365, e-mail,…



System performance testing

- propustnost s danou zátěží za danou jednotku času
- specifikace výkonnosti jsou většinou sepsány v SLA
- různé typy testování:
    - **endurance testing** – identifikace úniků pod kontinuální, očekávanou zátěží
    - **load testing** – chování aplikace pod specifickou zátěží
    - **spike testing** – chování aplikace při dramatických změnách v zátěži
    - **stress testing** – identifikace bodu zlomu při dlouhodobých dramatických výkyvech v zátěži