

Elosztott rendszerek: Alapelvek és paradigmák

Distributed Systems: Principles and Paradigms

Maarten van Steen¹ Kitlei Róbert²

¹VU Amsterdam, Dept. Computer Science

²ELTE Informatikai Kar

12. rész: Elosztott webalapú rendszerek

2015. május 24.

Tartalomjegyzék

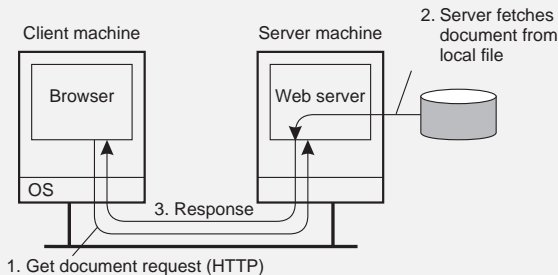
| Fejezet |
|--|
| 01: Bevezetés |
| 02: Architektúrák |
| 03: Folyamatok |
| 04: Kommunikáció |
| 05: Elnevezési rendszerek |
| 06: Szinkronizáció |
| 07: Konzisztencia & replikáció |
| 08: Hibatűrés |
| 10: Objektumalapú elosztott rendszerek |
| 11: Elosztott fájlrendszerek |
| 12: Elosztott webalapú rendszerek |

Elosztott webalapú rendszerek

A WWW (világháló, World Wide Web) olyan szerverek összessége, amelyek HTTP protokollon keresztül különféle tartalmakat szolgáltatnak ki. A dokumentumokat **hiperhivatkozások** kapcsolják össze.

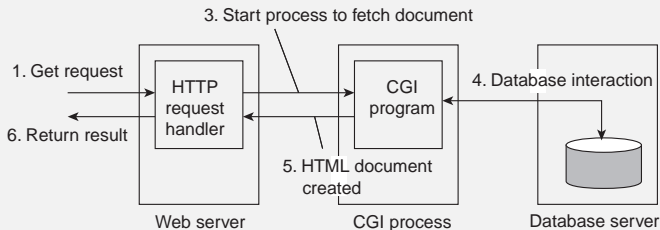
- Sok dokumentum szövegalapú: szövegfájl, HTML, XML
- Egyéb fajták: képek, audio, videó, dokumentum (PDF, PS)
- A tartalmak lehetnek a kliensoldalon végrehajthatók (Javascript)

Jelenleg **~1 milliárd weboldal** létezik, amelyek közül kevesebb mint 200 millió aktív; **egy felmérés szerint** a szerverek népszerűsége: Apache (40%), IIS (29%), nginx (15%), egyéb (16%).



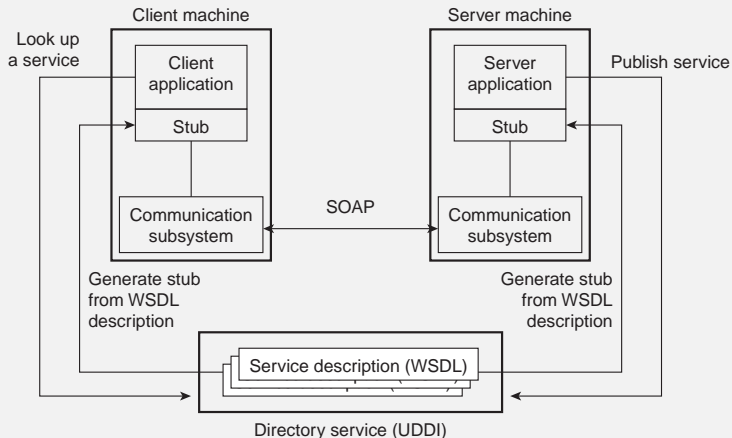
Többrétegű architektúrák

Már a kezdeti kiszolgálók is gyakran három rétegbe tagozódtak.



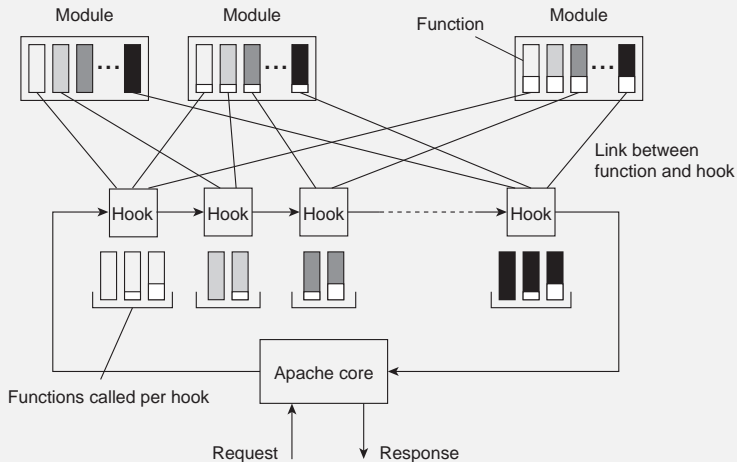
Webszolgáltatások

Felmerült az is, hogy a **felhasználó** ↔ **weboldal** interakció mellett az oldalak is igénybe vehetnek **szolgáltatásokat** más oldalakról ⇒ fontos, hogy a szolgáltatások **szabványosak** legyenek.



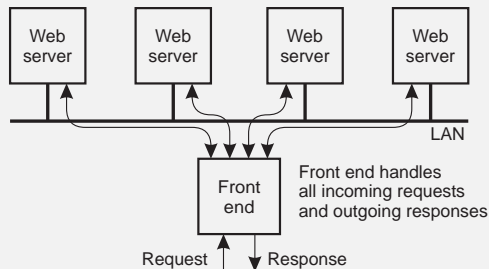
Webszerverek

A szerver szerkezetét a tartalmak kiszolgálásának menete szabja meg. A szerverekbe beépülő modulok telepíthetők, amelyek a kiszolgálás egyes fázisaiban aktivizálódnak.



Szerverfürtök

A teljesítmény és a rendelkezésre állás növelésének érdekében a szerverek sokszor (a felhasználó számára átlátszó módon) többszörözve vannak.

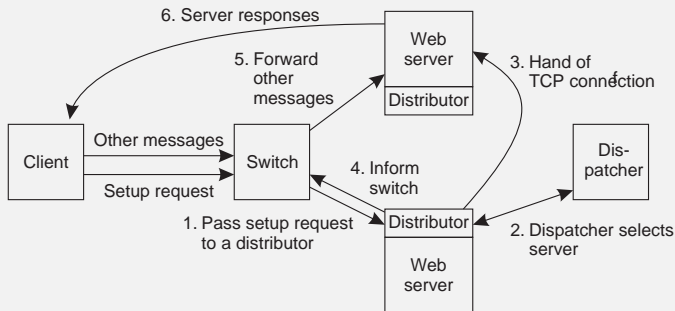


A kapcsolattartó (front end) szűk keresztmetszetté válhat, ennek elkerülésére több lehetőség van.

- **TCP átadás:** Valamilyen metrika alapján kiválasztunk egy szervert, és a kliens kiszolgálását az a szerver folytatja.
- **Tartalomérzékeny kéréselosztás** (content aware distribution): Lásd következő oldal.

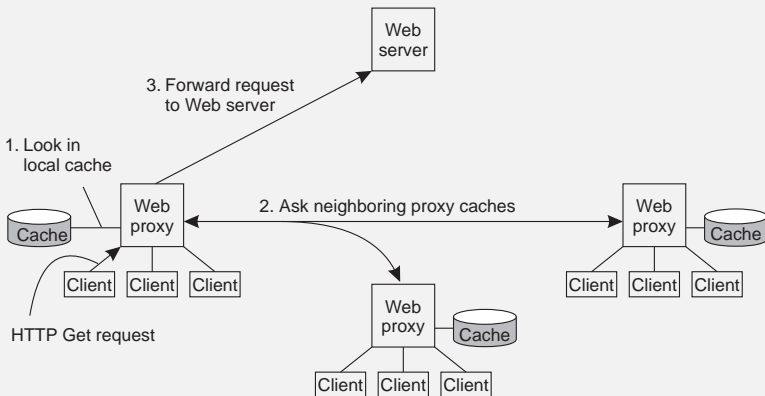
Szerverfürtök

Tartalomérzékeny kéréselosztás (content aware distribution): A HTTP kérés tartalmát is figyelembe vesszük a szerver kiválasztásánál. Ez megnöveli a kapcsolattartó terhelését, de sok előnye van: segítségével hatékonyabb lehet a szerveroldali cache-elés, és lehetnek bizonyos feladatokra dedikált szervereink.



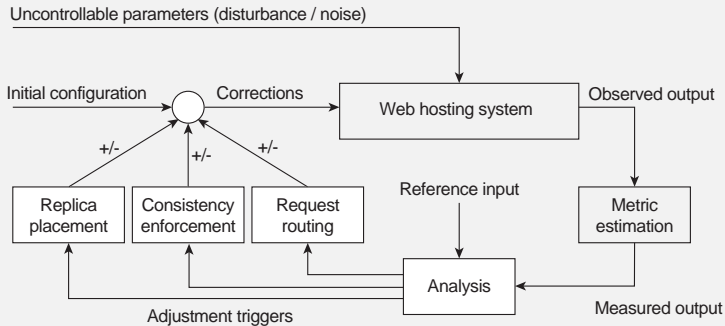
Webhelyettes

A kimenő kapcsolatok kezelésére **webhelyetteseket** (web proxy) telepíthetünk. Ezek cache-elik a kiszolgált tartalmakat; csak akkor fordulnak a szerverekhez, ha sem náluk, sem a többi helyettesnél nincsen meg a kért tartalom.



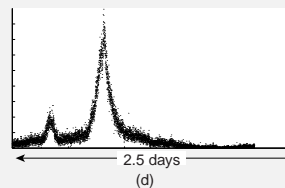
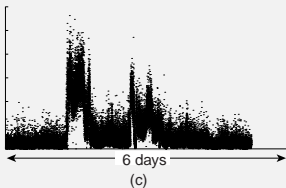
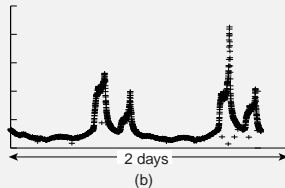
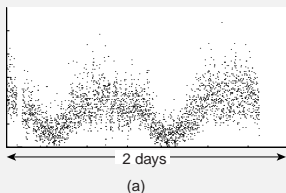
Replikáció webkiszolgálókban

A replikáció célja a teljesítmény növelése. A rendszer paraméterei (hová célszerű a replikátumokat elhelyezni, konzisztencia megkövetelt erőssége, kérések útvonalválasztása) változóak lehetnek, ezeket célszerű **önszabályozással** beállítani.



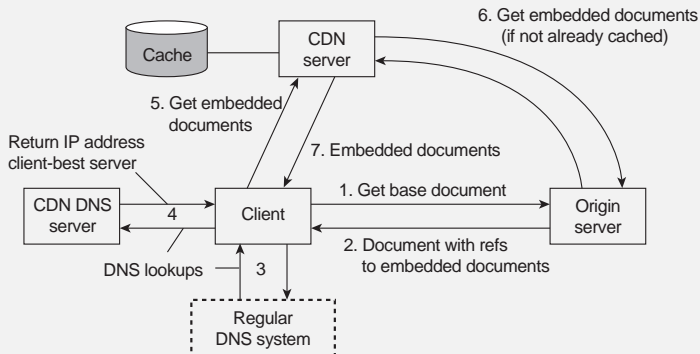
Hirtelen terhelés

A terhelés néha ugrásszerűen megemelkedik (flash crowd, flash mob), ezt még akkor sem könnyű kezelni, ha az erőforráskezelés dinamikus.



Szerveroldali replikáció

A tartalomkézbesítő hálózatok (Content Delivery Network, CDN) nagy teljesítményű és rendelkezésre állású elosztott rendszerek, amelyeknek célja dokumentumok hatékony kiszolgálása.



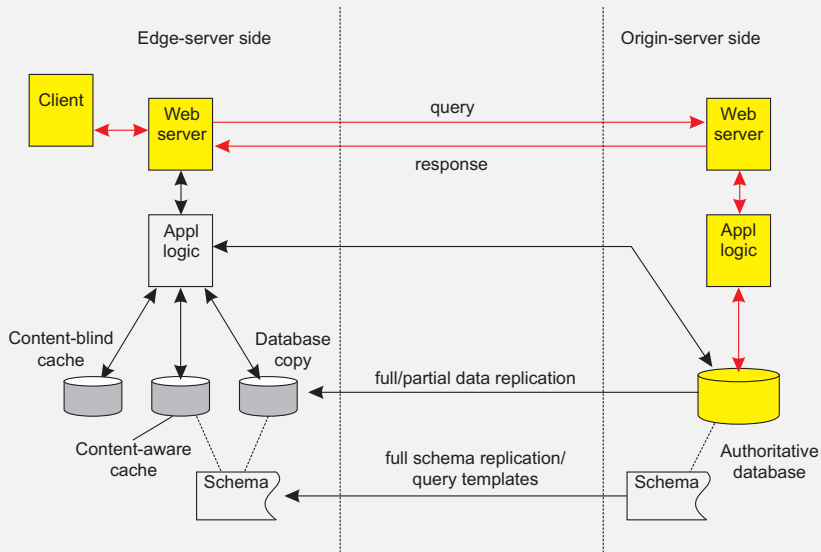
Replikáció webalkalmazásokban

Ha a CDN tárolt adataiban változás következik be, ez először az eredetszerveren jelenik meg. A változásokat el kell juttatni a CDN szerverekhez; ennek a célszerű módja a rendszer jellegétől függ.

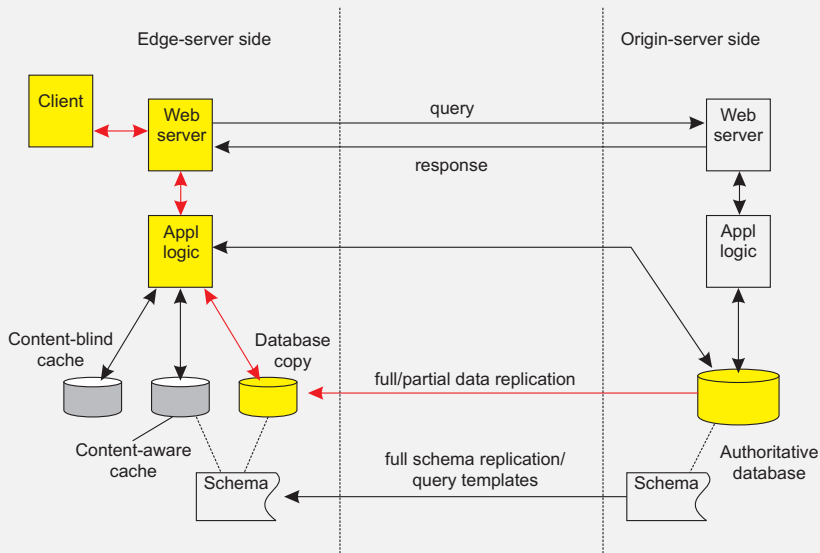
- **Teljes replikáció:** sok olvasás, kevés írás, **összetett** lekérdezések
- **Részleges replikáció:** sok olv., kevés írás, **egyszerű** lekérdezések
- **Tartalom szerinti gyorsítótárazás:** Az adatbázist az edge szerver módosított, a lekérdezésekhez illeszkedő alakban tárolja helyben, és feliratkozik a szerveren a frissítésekre. Jól működik **intervallumokra vonatkozó, összetett** lekérdezésekre.
- **Eredmények gyorsítótárazása:** Az edge szerver a korábbi lekérdezések eredményeit tárolja el. Jól működik **egyszerű** lekérdezésekre, amelyek **egydi adatokra** (nem intervallumokra) vonatkoznak.

Ha az írások számaránya megnő, akkor a replikáció akár ronthatja is a rendszer teljesítményét.

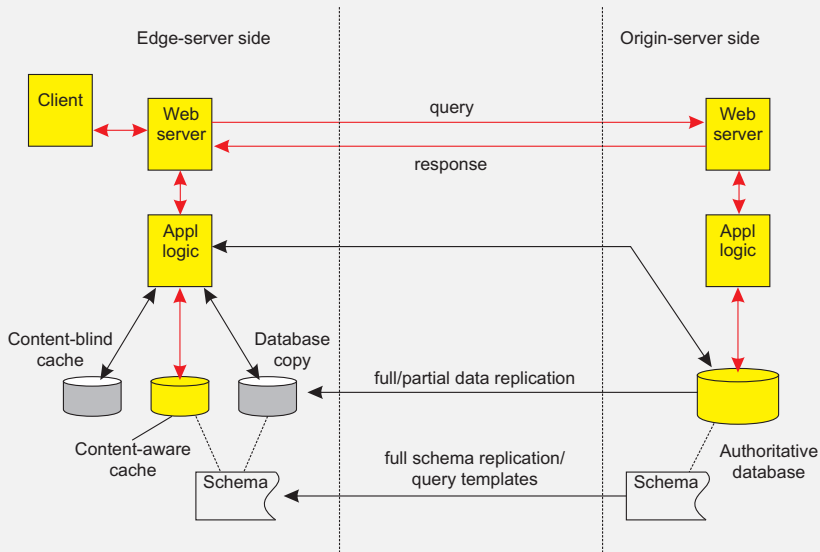
Replikáció webalkalmazásokban: nincs replikáció



Webalk. replikációja: részleges/teljes replikáció



Webalk. replikációja: tartalom szerinti gyorsítótárazás



Webalk. replikációja: eredmények gyorsítótárazása

