Elosztott rendszerek: Alapelvek és paradigmák
Distributed Systems: Principles and Paradigms

Maarten van Steen[1] Kitlei Róbert [2]

[1]VU Amsterdam, Dept. Computer Science
[2]ELTE Informatikai Kar

1. rész: Bevezetés

2015. május 24.

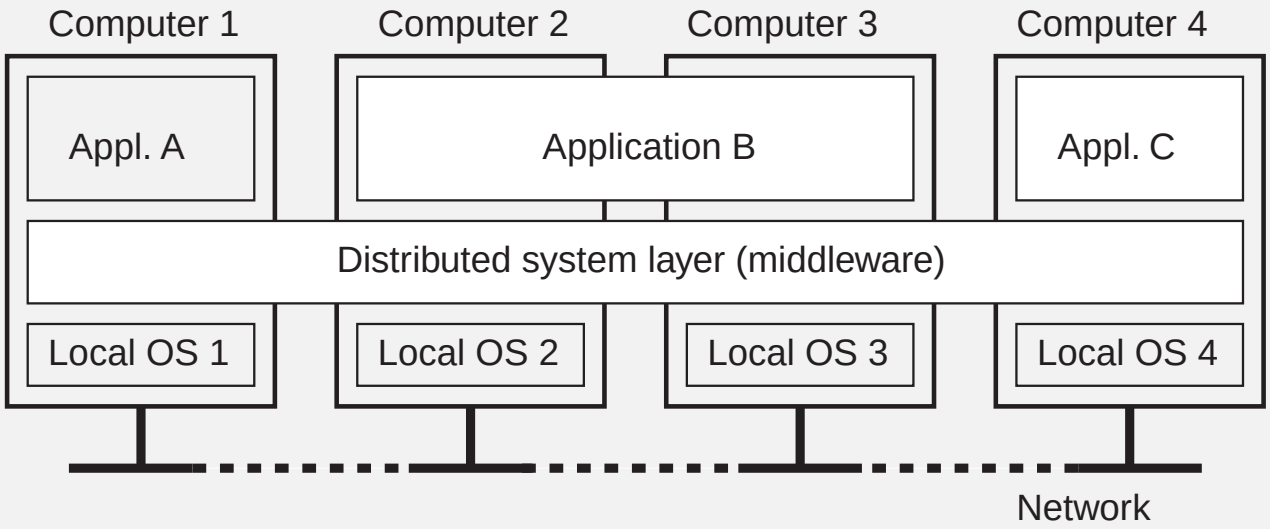
Tartalomjegyzék

Definíció: Elosztott rendszer

Az elosztott rendszer

> *önálló számítógépek olyan összessége, amely kezelői számára egyetlen koherens rendszernek tűnik.*

Két szempont: (1) független számítógépek és
(2) egyetlen rendszer $\Rightarrow$ köztesréteg (middleware).



Az elosztott rendszer céljai

- Távoli erőforrások elérhetővé tétele
- Átlátszóság (distribution transparency)
- Nyitottság (openness)
- Skálázhatóság (scalability)

Átlátszóság

Fajta	Angolul	Mit rejt el az erőforrással kapcsolatban?
Hozzáférési/elérési	Access	Adatábrázolás; elérés technikai részletei
Elhelyezési	Location	Fizikai elhelyezkedés
Áthelyezési	Migration	Elhelyezési + a hely meg is változhat
Mozgatási	Relocation	Áthelyezési + használat közben is történhet az áthelyezés
Többszörözési	Replication	Az erőforrásnak több másolata is lehet a rendszerben
Egyidejűségi	Concurrency	Több versenyhelyzetű felhasználó is elérheti egyszerre
Meghibásodási	Failure	Meghibásodhat és újra üzembe állhat

Megjegyzés

A fentiek lehetséges követelmények, amelyeket a rendszerrel kapcsolatban támaszthatunk. A félév során megvizsgáljuk, hogy melyik mennyire érhető el.

Az átlátszóság mértéke

Mekkora átlátszóságot várhatunk el?

A teljes átlátszóságra törekvés általában túl erős:

- A felhasználók különböző kontinenseken tartózkodhatnak
- A hálózatok és az egyes gépek meghibásodásának teljes elfedése elméletileg és gyakorlatilag is lehetetlen
  - Nem lehet eldönteni, hogy a szerver csak lassan válaszol vagy meghibásodott
  - Távolról nem megállapítható, hogy a szerver feldolgozta-e a kérésünket, mielőtt összeomlott
- A nagymértékű átlátszóság a hatékonyság rovására megy, de a késleltetést is el szeretnénk rejteni
  - Ilyen feladat lehet a webes gyorsítótárak (cache-ek) tökéletesen frissen tartása
  - Másik példa: minden változás azonnal lemezre írása nagymértékű hibatűréshez

Elosztott rendszerek nyitottsága

Nyitott elosztott rendszer

A rendszer képes más nyitott rendszerek számára szolgáltatásokat nyújtani, és azok szolgáltatásait igénybe venni:

- A rendszerek jól definiált interfészekkel rendelkeznek
- Az alkalmazások hordozhatóságát (portability) minél inkább támogatják
- Könnyen elérhető a rendszerek együttműködése (interoperability)

A nyitottság elérése

A nyitott elosztott rendszer legyen könnyen alkalmazható heterogén környezetben, azaz különböző

- hardvereken,
- platformokon,
- programozási nyelveken.

Nyitottság: követelményrendszerek, mechanizmusok

A nyitottság implementálása

- Fontos, hogy a rendszer könnyen cserélhető részekből álljon
- Belső interfészek használata, nem egyetlen monolitikus rendszer
- A rendszernek minél jobban paraméterezhetőnek kell lennie
- Egyetlen komponens megváltoztatása/cseréje lehetőleg minél kevésbé hasson a rendszer más részeire

Nyitottság: követelményrendszerek, mechanizmusok

Példák

A nyitott rendszer követelményeket (policy) ír elő, amelyekhez jó, ha minél több megvalósító megvalósítás (mechanism) érhető el a rendszerben. Néhány szóbajöhető példa:

- Mennyire erős konzisztenciát követeljünk meg a kliensoldalon cache-elt adatokra? Legyen dinamikusan beállítható, hogyan döntse el a rendszer az adatról, milyen erősen legyen konzisztens.
- A letöltött kód milyen műveleteket hajthasson végre? A mobil kódhoz rendeljünk különböző megbízhatósági szinteket.
- Milyen QoS követelményeket támasszunk változó sávszéleségű rendszerben? Minden folyamra külön lehessen QoS követelményeket beállítani.
- Milyen mértékben titkosítsuk a kommunikációs csatornánkat? A rendszer kínáljon többfajta titkosítási mechanizmust, amelyek közül választani lehet.

Átméretezhetőség (scalability)

Átméretezhetőség

Ha egy „kis” rendszer megnő, az sokfajta kihívást jelenthet. Több különböző jellege is megnőhet a rendszernek:

- méret szerinti átméretezhetőség: több felhasználó és/vagy folyamat van a rendszerben
- földrajzi átméretezhetőség: a rendszert nagyobb területről veszik igénybe, pl. egyetemen belüli felhasználás→világméretű felhasználóbázis
- adminisztrációs átméretezhetőség: biztonsági, karbantartási, együttműködési kérdések merülnek fel, ha új adminisztrációs tartományok kerülnek a rendszerbe

Megjegyzés

A legtöbb rendszer a méret szerinti átméretezhetőséget kezeli.
(Nem mindig) megoldás: erősebb szerverek használata.
A másik két jellegű átméretezhetőséget nagyobb kihívás kezelni.

Technikák az átméretezhetőség megvalósítására

A kommunikációs késleltetés elfedése

A válaszra várás közben más tevékenység végzése:

- Aszinkron kommunikáció használata
- A beérkező választ külön kezelő dolgozza fel
- Probléma: nem minden alkalmazás ültethető át ilyen megközelítésre

Technikák az átméretezhetőség megvalósítására

Elosztás

Az adatokat és a számításokat több számítógép tárolja/végzi:

- A számítások egy részét a kliensoldal végzi (Java appletek)
- Decentralizált elnevezési rendszerek (DNS)
- Decentralizált információs rendszerek (WWW)

Technikák az átméretezhetőség megvalósítására

Replikáció/cache-elés

Több számítógép tárolja egy adat másolatait:

- Replikált fájlszerverek és adatbázisok
- Tükrözött weboldalak
- Weboldalak cache-elése (böngészőkben, proxy szervereken)
- Fájlok cache-elése (a szerver- és kliensoldalon)

Átméretezhetőség – a probléma

Megjegyzés

Az átméretezhetőség könnyen elérhető, de ára van:

- Több másolat fenntartása (cache vagy replika) inkonzisztenciához vezet: ha módosítunk egy másolatot, az eltér a többitől.
- A másolatok konzisztensen tartásához globális szinkronizációra van szükség minden egyes változtatás után.
- A globális szinkronizáció viszont rosszul skálázható nagy rendszerekre.

Következmény

Ha feladjuk a globális szinkronizációt, akkor kénytelenek vagyunk bizonyos fokú inkonzisztenciát elviselni a rendszerünkben.
Az, hogy ez milyen mértékben elfogadható, rendszerfüggő.

Elosztott rendszerek fejlesztése: hibalehetőségek

Megjegyzés

Az elosztott rendszer környezetéről kényelmes lehet feltételezni, hogy megbízható. Ha ez tévesnek bizonyul, az a rendszer újratervezéséhez vezethet. Néhány ilyen feltételezés:

- a hálózat hibamentes
- a hálózat biztonságos
- a hálózat homogén
- a hálózati topológia nem változik
- a kommunikációnak nincsen időigénye
- a sávszélesség korlátlan
- a kommunikációnak nincsen költsége
- csak egy adminisztrátor van

Elosztott rendszerek fajtái

- Elosztott számítási rendszerek
  - grid
  - cloud
  - információs rendszerek
- Elosztott információs rendszerek
- Elosztott átható (pervasive, ubiquitous) rendszerek

Elosztott számítási rendszerek

Számítási rendszer

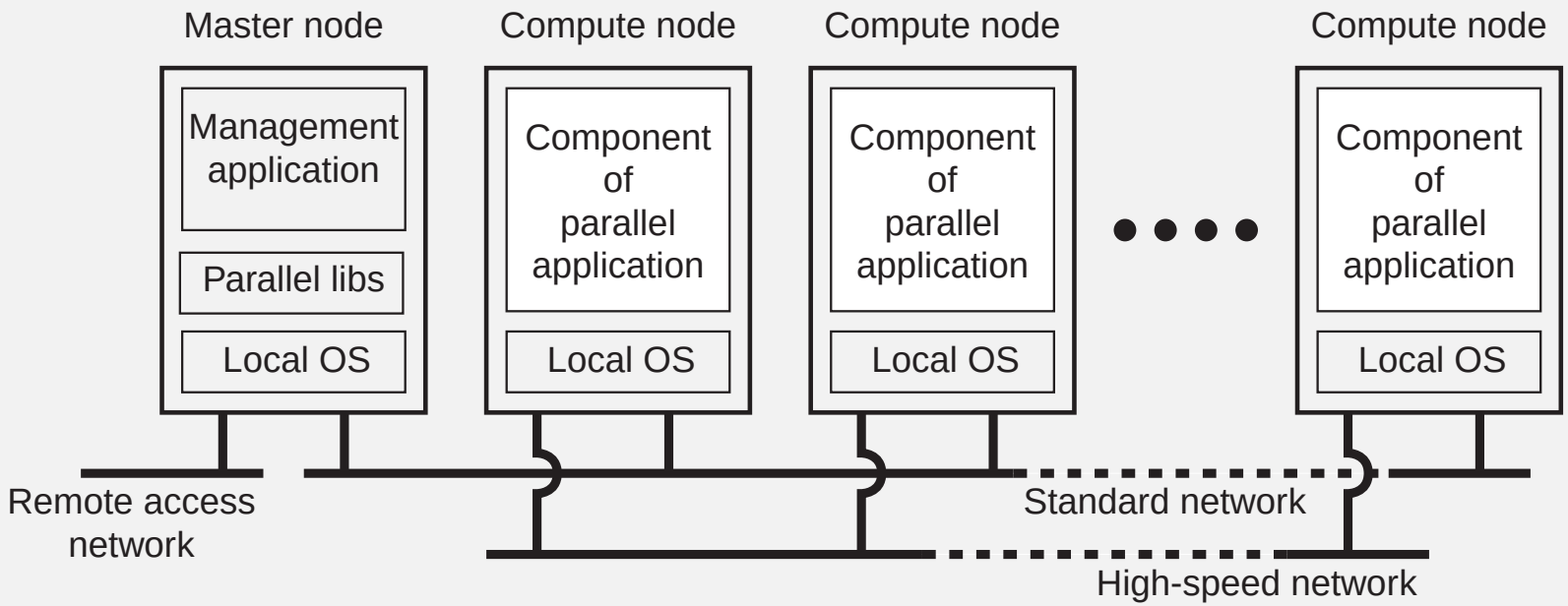
Sok elosztott rendszer célja számítások végzése nagy teljesítménnyel.

Cluster (fürt)

Lokális hálózatra kapcsolt számítógépek összessége.

- Homogén: ugyanaz az operációs rendszer, hardveresen nem vagy csak alig térnek el
- A vezérlés központosítva van, általában egyetlen gépre

Elosztott számítási rendszerek



Elosztott számítási rendszerek

Grid (rács)

Több gép, kevésbé egységesek:

- Heterogén architektúra
- Átívelhet több szervezeti egységen
- Nagyméretű hálózatokra terjedhet ki

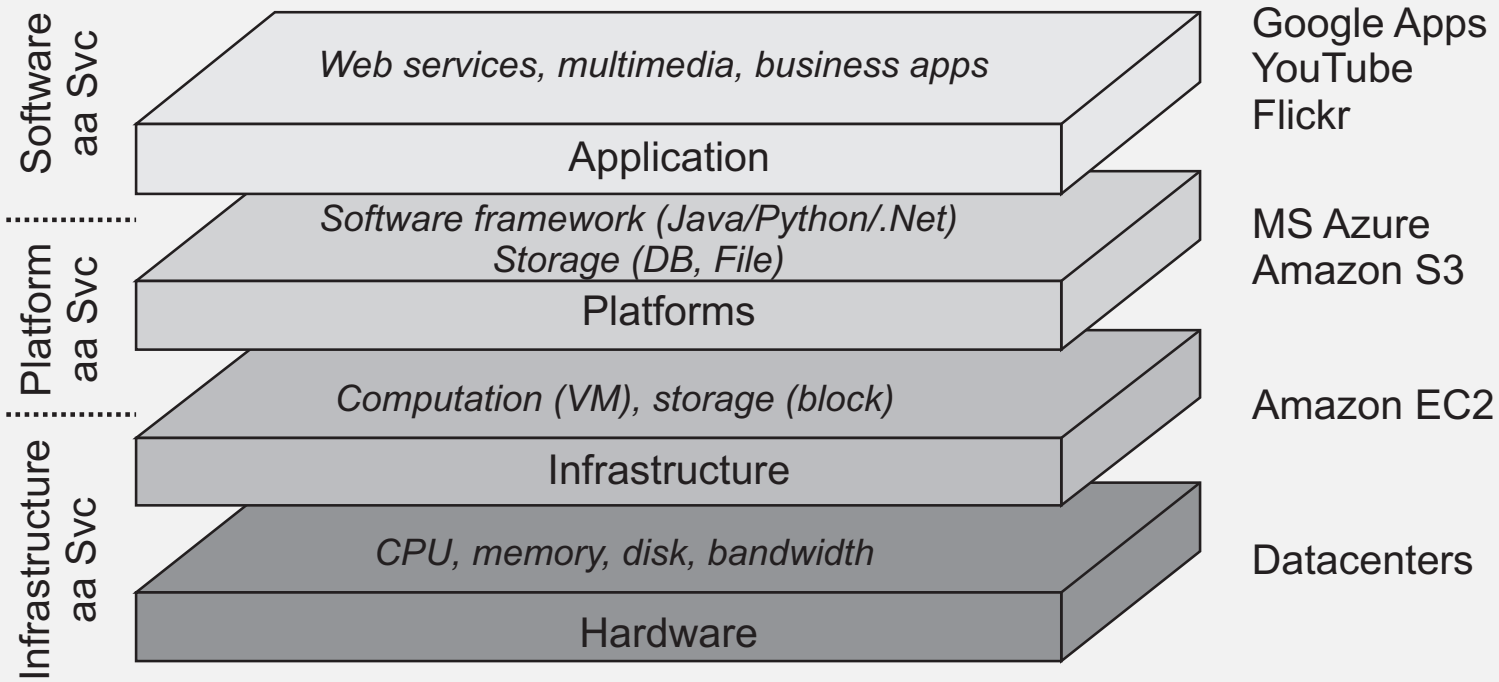
Megjegyzés

Az együttműködést sokszor virtuális szervezetek kialakításával segítik. Ez a felhasználókat csoportosítja, akiknek így egységesen lehet erőforrásokat kiutalni.

Elosztott számítási rendszerek

Felhő (cloud)

Többrétegű architektúra.



Elosztott számítási rendszerek

Felhő

Négy különböző réteg:

- Hardver: Processzorok, útválasztók (routerek), áramforrások, hűtőberendezések. A felhasználók közvetlenül nem látják.
- Infrastruktúra: Virtuális hardvert tesz elérhetővé: szerver, adattároló, hálózati kapcsolat, számítási kapacitás lefoglalása és kezelése.
- Platform: Magasabb szintű absztrakciókat biztosít. Pl. az Amazon S3 társzolgáltatás különböző fájlműveleteket biztosít; a felhasználónak vödrei (bucket) vannak, ebbe feltölthet, letölthet stb. fájlokat egy API segítségével .
- Alkalmazás: A végfelhasználónak szánt, jellemzően grafikus felületű alkalmazások.

Elosztott információs rendszerek

Elosztott információs rendszerek

Sok elosztott rendszer elsődleges célja adatok kezelése, illetve meglevő ilyen rendszerek elérése. Példa: tranzakciókezelő rendszerek.

```
BEGIN_TRANSACTION(server, transaction)
READ(transaction, file-1, data)
WRITE(transaction, file-2, data)
newData := MODIFIED(data)
IF WRONG(newData) THEN
    ABORT_TRANSACTION(transaction)
ELSE
    WRITE(transaction, file-2, newData)
    END_TRANSACTION(transaction)
END IF
```

Elosztott információs rendszerek: tranzakciók

Modell

A tranzakció adatok összességén (adatbázis, objektumok vagy más adattár) végzett művelet (lehetnek részműveletei), melynek az alábbi tulajdonságai vannak. A kezdőbetűk rövidítéséből ACID-nek szokás nevezni a követelményrendszert.

Oszthatatlan, elemi (atomicity): Vagy a teljes tranzakció végbemegy minden részműveletével, vagy pedig az adattár egyáltalán nem változik meg.

Konzisztens (consistency): Az adattárra akkor mondjuk, hogy érvényes, ha (az adattárra jellemző) feltételek teljesülnek rá. A tranzakció konzisztens, ha érvényes állapotot állít elő. Ez a követelmény csak a tranzakció végére vonatkozik: menet közben előállhat érvénytelen állapot.

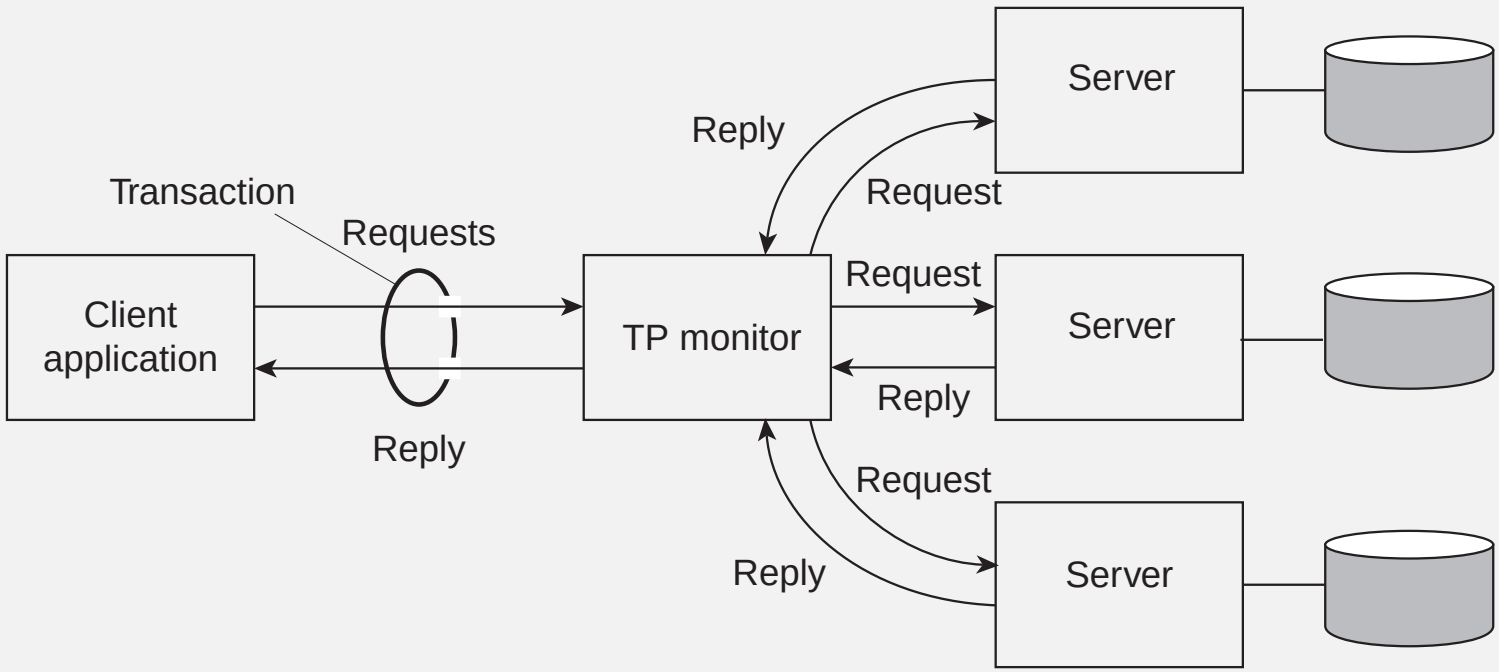
Elkülönülő, sorosítható (isolation): Egyszerre zajló tranzakciók "nem zavarják" egymást: olyan eredményt adnak, mintha egymás után sorban futottak volna le.

Tartósság (durability): Végrehajtás után az eredményt tartós adattárolóra mentjük, így a rendszer esetleges összeomlása után visszaállítható.

Tranzakciófeldolgozó monitor (Transaction Processing Monitor)

Megjegyzés

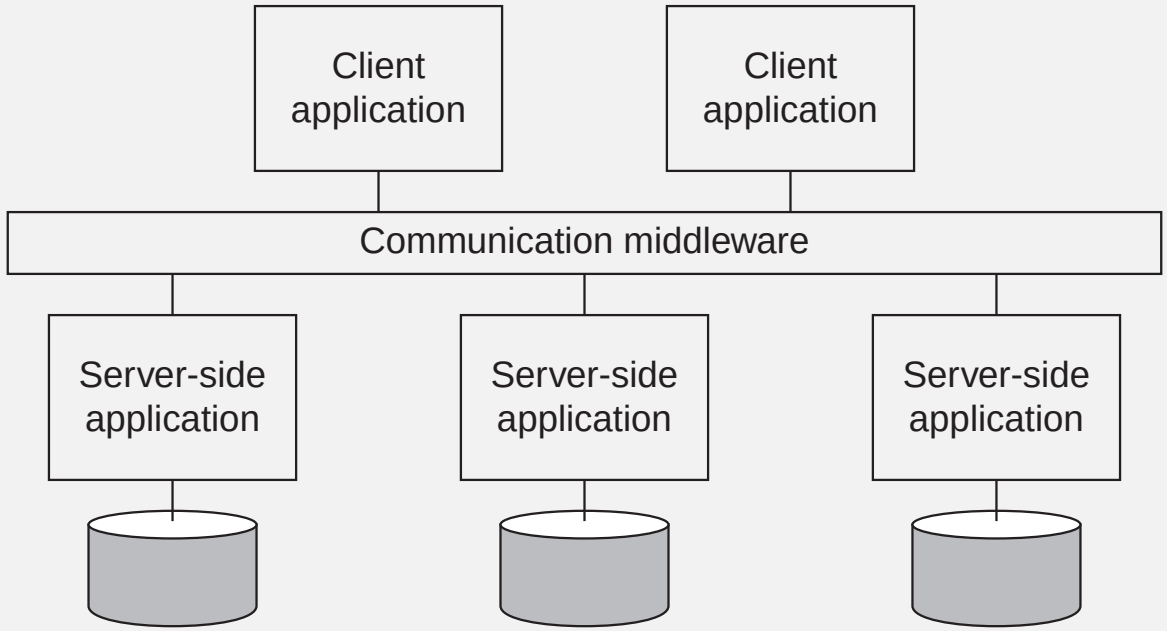
A tranzakciókat sokszor több szerver hajtja végre. Ezeket egy TP monitor vezérli.



Alkalmazásintegráció nagy rendszerekben

Probléma

A TP monitor nem választja el az alkalmazásokat az adatbázisoktól. Továbbá az alkalmazásoknak egymással is kommunikálniuk kell.



- Távoli eljáráshívás (Remote Procedure Call, RPC)
- Üzenetorientált köztesréteg (Message-Oriented Middleware, MOM)

Elosztott átható rendszerek

Átható rendszer

Sok modern elosztott rendszer kicsi, mobil elemekből áll.

Néhány jellemző

- A környezet megváltozhat: A rendszernek ezt követnie kell.
- Ad hoc szerveződés: A rendszer komponenseit nagyon különböző módokon használhatják a felhasználók. Ezért a rendszernek könnyen konfigurálhatónak kell lennie.
- Megosztott szolgáltatások: Mivel a rendszer nagyon változékony, az adatoknak könnyen kell áramlaniuk. Ennek elősegítésére a rendszer elemei általában nagyon egyszerű szerkezetűek.

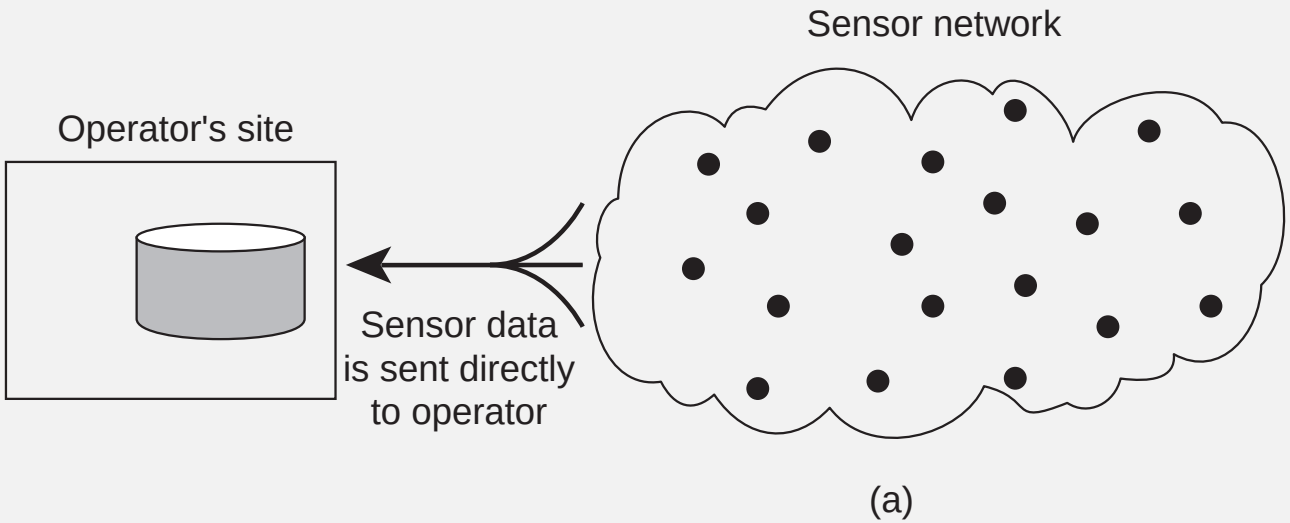
Érzékelőhálózatok

Jellemzők

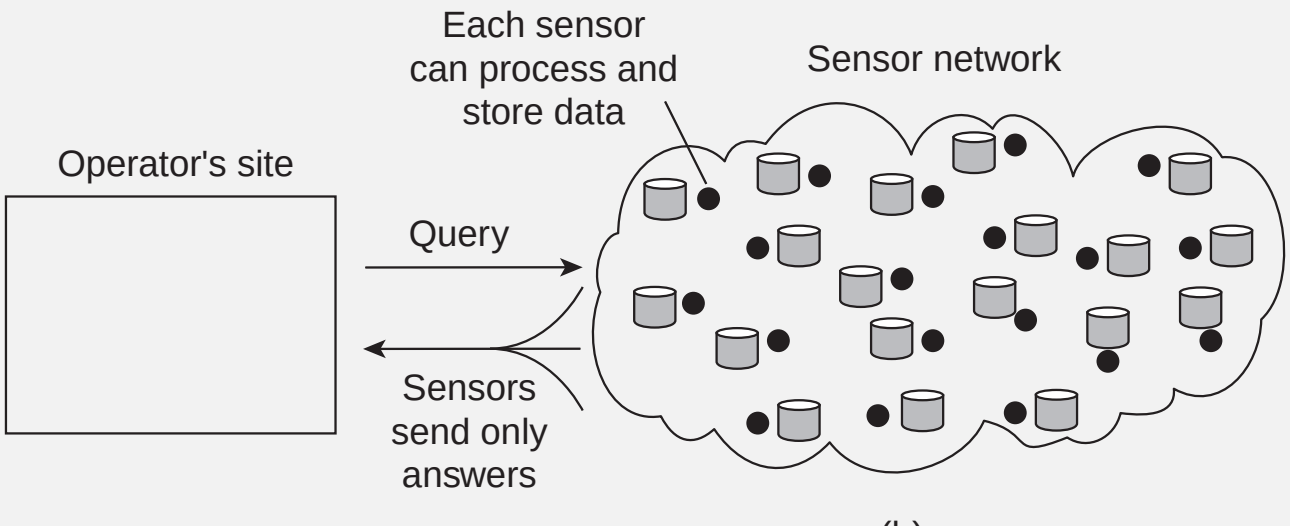
Az érzékelőket tartalmazó csúcsok

- sok van belőlük (nagyságrendileg 10-1000 darab)
- egyszerűek (kevés memória, számítási és kommunikációs kapacitás)
- sokszor elemről működnek, vagy áramforrás sem szükséges hozzájuk

Érzékelőhálózatok mint elosztott rendszerek



(a)



(b)

Példák

Tömegirányítás

- Helyzet: rendezvény fix útvonalakkal (kiállítás, fesztivál)
- Cél: az embereket a megfelelő helyre irányítani
  - a hasonló érdeklődésű emberek egy helyre menjenek
  - vészhelyzet esetén a fenti csoportokat ugyanahhoz a kijárathoz irányítani
- Cél: összetartozó embereket (pl. családokat) egyben tartani

Példák

Szociális játék

- Helyzet: Konferencia, a résztvevők különböző csoportokban érkeztek.
- Cél: A csoportok vegyítésének elősegítése.
- Megközelítés: A rendszer figyeli, hogy a csoportok hogyan viselkednek
  - Ha külön csoportokból származó embereket észlel, bónuszpontokat kapnak az emberek és a csoportok egyaránt.
  - A rendszer kiosztja a csoportszintű pontokat a tagok között.
  - Az eredményeket elektronikus kitűzők mutatják (feedback and social intervention).

Példa: Szociális játék

A szociális játék egy kitűzője.