Hálózatba kapcsolt adatbázisok – Magas rendelkezésreállás

Erős Levente, TMIT
eros@tmit.bme.hu
2011.

Tartalom

- Mi az, hogy rendelkezésreállás?
- Miért fontos?
- Hogyan mérjük?
- Mitől sérül?
- Védelmi szintek
- Rendelkezésreállási technológiák
- Példa – Oracle Data Guard
- Brewer CAP elmélete

Mi az, hogy rendelkezésreállás?

- „*the degree to which a system or component is operational and accessible when required for use*”
- Magas rendelkezésreállás
  - Amikor használnunk kell a rendszert, az nagyon nagy valószínűséggel rendelkezésre áll.

Miért fontos?

- 1 óra kiesés mekkora veszteséget okoz egy cégnek?
  - 2001-es Cost of Downtime Survey alapján

46%	Kevesebb, mint 50 000 USD
28%	51 000 USD – 250 000 USD
18%	251 000 USD – 1 000 000 USD
8%	Több, mint 1 000 000 USD

Miért fontos?

- N óra kiesés költsége nem N*1 óra kiesés költsége!
  - Nem lineáris
  - Csökkent termelés
  - Kevésbé elégedett ügyfelek
  - Elveszített ügyfelek

Miért fontos?

- Cégek átlagosan legfeljebb 2-3 napos kieséseket élnek túl.
- Mekkora kieséstől megy csődbe a cég?

40%	72 óra
21%	48 óra
15%	24 óra
8%	8 óra
9%	4 óra
3%	1 óra
4%	1 óránál kevesebb

Tartalom

- Mi az, hogy rendelkezésreállás?
- Miért fontos?
- Hogyan mérjük?
- Mitől sérül?
- Védelmi szintek
- Rendelkezésreállási technológiák
- Példa – Oracle Data Guard
- Brewer CAP elmélete

Hogyan mérjük?

- „Kilencesek száma”
  - 5 kilences – Az idő 99,999%-ában rendelkezésre áll
  - 2 kilences – Az idő 99%-ában rendelkezésre áll
    - Soknak tűnik
    - Évente több mint 3 nap kiesés!
      - 10 db 9 órás leállás
    - Soknak tűnik?

Hogyan mérjük?

- Rendelkezésreállási osztályok

Osztály	Rendelkezésreállás	Évenkénti kiesés	Példa
1	90% - 99%	3,65 nap – 1,2 hónap	???????
2	99% - 99,9%	8 óra 45 perc – 3,65 nap	Irodai PC
3	99,9% - 99,99%	52 perc 30 mp – 8 óra 45 perc	Üzenetküldő rendszerek
4	99,99% - 99,999%	5 perc 15 mp – 52 perc 30 mp	Ügyfélszolgálat, e-kereskedelem
5	99,999% - 99,9999%	31,5 mp – 5 perc 15 mp	Telekommunikáció, navigáció, ATM-ek
6	99,9999% - 99,99999%	3 mp – 31,5 mp	Védelmi rendszerek, repülőgép-számítógépek

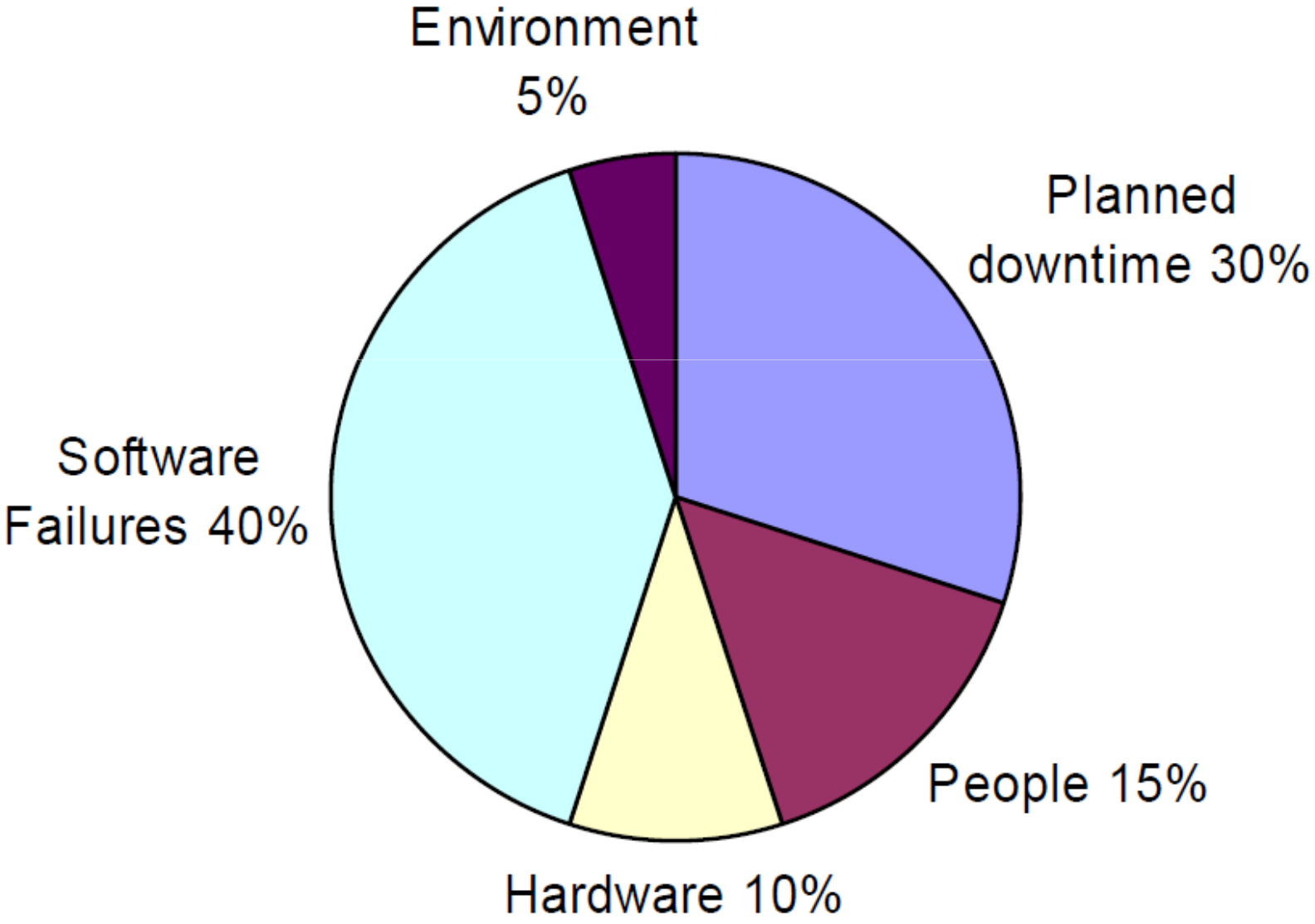
Hogyan mérjük?

- Pontosabb mérés érdekében két paraméter:
  - MTBF (Mean Time Between Failures)
    - Meghibásodások között eltelt átlagidő
  - MTTR (Mean Time To Repair)
    - Javítás átlagideje
  - Statisztikai átlagadatok, nem garantált!!!
- Rendelkezésreállás
  - (MTBF / (MTBF+MTTR)) * 100
    - A rendszer évente egyszer áll le, és 12 percbe kerül a javítása átlagban
    - Rendelkezésreállás = (8758,8 / 8760) *100 = 99,98

Mitől sérül?

- Tervezett leállás
  - Leállások 30%-a
  - Szoftverfrissítés, hardverfrissítés, egyéb karbantartás
  - Manuális leállítás
  - Nincs adatvesztés
- Nem tervezett leállás
  - Hardversérülés
  - Szoftverhiba
  - Környezeti hatások
    - Áramkimaradás
    - Katasztrófák
  - Emberi hiba

Mitől sérül?


Environment
5%
Planned
downtime 30%
Software
Failures 40%
People 15%
Hardware 10%

Tartalom

- Mi az, hogy rendelkezésreállás?
- Miért fontos?
- Hogyan mérjük?
- Mitől sérül?
- Védelmi szintek
- Rendelkezésreállási technológiák
- Példa – Oracle Data Guard
- Brewer CAP elmélete

Védelmi szintek

- 1 – Nincs védelem
  - Ritka biztonsági mentések
  - Hiba kieséshez és adatvesztéshez vezet
- 2 – Adatvédelem
  - Lényeg az adatok védelme
  - Kiesés lehet, adatvesztés nem
  - Redundáns adattárolás, pl. RAID

Védelmi szintek

- 3 – Rendszervédelem
  - Kiesés elkerülése a cél
  - Rendszerszintű reundancia
  - Tartalékszerver
- 4 – cégvédelem
  - Katasztrófa elleni védelem is
  - Távoli tartalék (szerverpark)
- Magasabb szintek drágábbak!

Rendelkezésreállási technológiák

- Önjavítás
- Maszkolás
- Javítás támogatása
- Konzisztencia biztosítása
- Biztonsági másolat
- Izoláció
- Klaszterezés

Önjavítás

- A hibákat a rendszer észleli, és automatikusan javítja, mielőtt megfigyelhető következményük lenne.
- Pl. Hibajavító kódolás

Maszkolás

- Komponenshiba detektálása, másik komponens azonnali üzembe helyezése
- Pl. RAID
  - Hibás lemez feladatait automatikusan átveszi egy másik
  - Hibás egység cserélendő
  - Szolgáltatás folytonos

Maszkolás – RAID

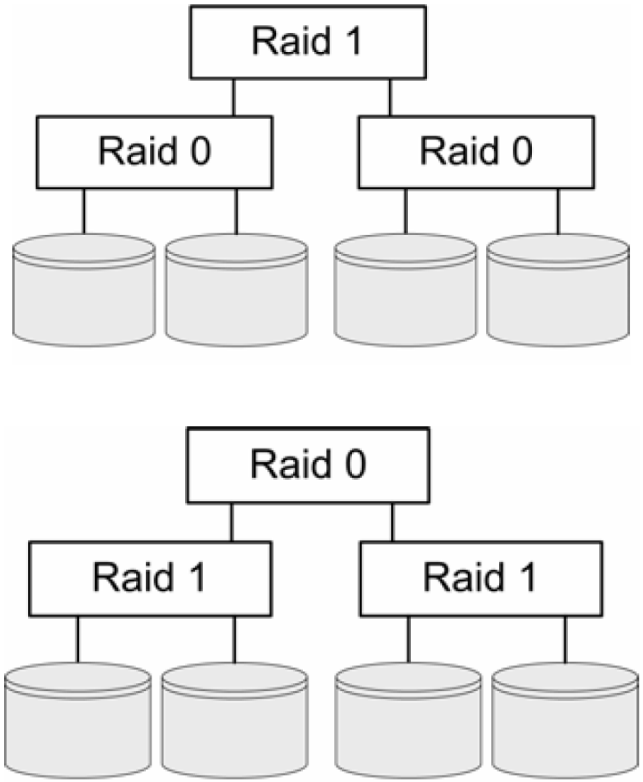
- Nem termék, hanem technológia
- Több lemez – 1 tárolási egység
- Redundant Array of Inexpensive Disks
- Redundant Array of Independent Disks
- Megvalósítás
  - Hardverszinten – RAID vezérlők
  - Szoftverszinten – Operációs rendszer feladata

Maszkolás – RAID

- Két alapvető RAID technológia
  - Összefűzés (RAID 0)
    - Lemezeken folytonosan tárolódik az adat
    - Nagyobb teljesítmény, kisebb rendelkezésreállás
  - Tükrözés (RAID 1)
    - Több lemezen tárolódik ugyanaz az adat
    - Kisebb teljesítmény, nagyobb rendelkezésreállás
  - Van még RAID 3,4,5,6

Maszkolás – RAID

- RAID 0+1
  - Összefűzött lemezek tükrözése
  - Egy lemez meghibásodásával az összefűzött tömb használhatatlan lesz, RAID 1 is sérül
- RAID 1+0
  - Tükrözött lemezek összefűzése
  - Egy lemez meghibásodásával csak a RAID 1 sérül, a rá épülő RAID 0 nem
  - RAID 1 elfedi a hibát



Javítás támogatása

- Monitoring rendszerek
  - Hibák előrejelzése statisztikai adatok, naplók alapján
- Hardverkarbantartás támogatása
  - Komponensek cseréje leállás nélkül

Konzisztencia biztosítása

- Leállás egyik legnagyobb veszélye
  - Adatintegritás elvesztése
- Naplózó fájlrendszerek
  - Változások naplózása
  - Konzisztens állapot visszaállítása
- Többfázisú commit
  - Mindenhol megjelenik az új adat, vagy sehol sem
  - Nem véd minden hiba ellen

Biztonsági másolat

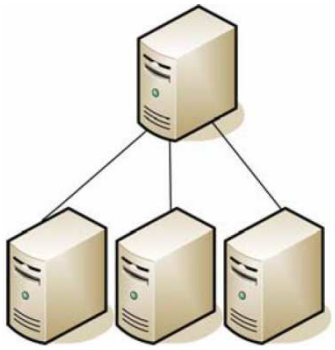
- Adatok mentése és biztonságos tárolása
- Fajtái
  - „Hot backup” – bármikor hozzáférhető
  - „Cold backup” – előkészítés után férhető hozzá
    - Szalagos biztonsági mentés
  - „Supplier backup” – szállító biztosítja adott hardvereszközök határidőn belüli szállítását

Izoláció

- Cél: Hibaterjedés megelőzése
  - Egy komponens hibája ne veszélyeztesse egy másik helyes működését
- Pl.: Alkalmazások és adatok szeparálása fizikailag
  - Külső adatbázis használata

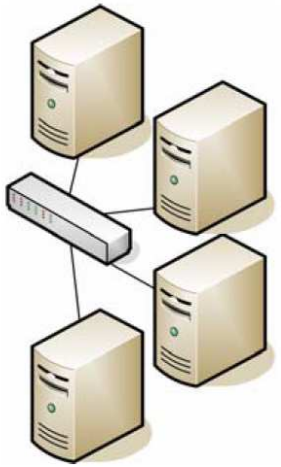
Klaszterezés

- Több összekötött számítógép, amelyek a felhasználó felé egyként jelennek meg
  - Egy gép meghibásodása esetén egy másik veszi át a feladatait – *failover*
  - Ha egy gép több vésztartalékaként is szolgál, a meghibásodás elhárítását követően a kijavított gépnek vissza kell vennie a feladatait – *failback*
    - Egyenrangú gépek esetén erre nincs szükség
      - A megjavított gép lesz a vésztartalék



← kell failback

nem kell failback →



Példa – Oracle Data Guard

- Biztonsági másolatok, tartalékadatbázisok karbantartása, failover-képességek
- Hangolható *rendelkezésreállás, megbízhatóság* és *teljesítmény* között
- Három működési mód
  - Maximum protection (adatvédelem)
  - Maximum performance (teljesítmény)
  - Maximum availability (rendelkezésreállás)

Példa – Oracle Data Guard

- Maximum protection működési mód
  - Adatvédelem mindenek előtt
  - Tranzakció committálását megelőzően a változtatásokat érvényesíteni kell legalább egy tartalékadatbázisban is.
  - Ha tartalékadatbázis kiesik, rendelkezésreállás sérül
    - Nincs commit

Példa – Oracle Data Guard

- Maximum performance működési mód
  - Teljesítményre optimalizál
  - Tartalékadatbázissal való szinkronizáció commit *után*, aszinkron módon történik
    - Elsődleges adatbázis teljesítménye nem romlik
    - Rosszabb adatvédelem

Példa – Oracle Data Guard

- Maximum availability működési mód
  - Adatvédelemre és teljesítményre optimalizál
  - Tranzakció committálását megelőzően a változtatásokat érvényesíteni kell legalább egy tartalékadatbázisban is.
    - DE ha tartalékadatbázis nem elérhető, maximum performance módban működik tovább addig, amíg újra elérhető nem lesz.
    - Elsődleges adatbázis rendelkezésreállásának fenntartása
    - Hibrid mód

Tartalom

- Mi az, hogy rendelkezésreállás?
- Miért fontos?
- Hogyan mérjük?
- Mitől sérül?
- Védelmi szintek
- Rendelkezésreállási technológiák
- Példa – Oracle Data Guard
- Brewer CAP elmélete

Brewer CAP elmélete

- Három rendszerszintű követelmény elosztott rendszereknél
  - Consistency
    - A rendszer mindig konzisztens
  - Availability
    - A rendszer rendelkezésreállása magas
  - Partition tolerance
    - Rendszer tűri, ha kommunikációs hiba miatt partíciókra esik
  - Ez a három követelmény egyszerre *nem biztosítható*
- Eric Brewer, 2000.
- Bizonyítás: Gilbert, Lynch, 2002.

Brewer CAP elmélete

- Megoldás
  - Szüntessük meg a partíciók lehetőségét (C+A)
    - Adott tranzakcióhoz tartozó valamennyi adatelem egy fizikai helyen
    - Nem skálázható (nem mindig kivitelezhető)
  - Áldozzuk fel a magas rendelkezésreállást (C+P)
    - Partíció esetén várjunk, amíg visszatér a konzisztencia
  - Konzisztencia feláldozása (A+P)
    - Inkonzisztenciából adódó hibák utólagos kijavítása
    - Skálázható? Újabb hiba előtt elvégezhető?
  - ACID helyett ellenétes BASE
    - Basically Available – általában rendelkezésre áll
    - Soft-State – állapotok között „lebeg”
    - Eventually Consistent – alkalmanként konzisztens (~soft-state)