Szolgáltatásbiztos rendszerek: Architektúra tervezési példák

Majzik István
majzik@mit.bme.hu

Feladatátvételi fürtök (Failover clustering, High availability clustering)

Hardver architektúra

- Egyszeres hibapont (SPOF) kiküszöbölése elfogadható áron
  - Redundáns autonóm szerverek
  - Diszk: RAID, SAN: blokkos (iSCSI), NAS: fájl szintű adattárhálózat
  - Hálózat: Redundáns kapcsolat kliensekkel és a fürtön belül is
  - Környezeti hibaforrások: Áramellátás, légkondicionálás, …
- Megosztás: Állapottal rendelkező szolgáltatások
  - Shared disk (shared SCSI, serial attached SCSI, Fibre Channel)
    - Fizikai szintű sorosítás, globális zármenedzser szükséges
  - Shared nothing (csak hálózati kapcsolat)
    - Logikai szinten biztosított a kizárólagos hozzáférés
  - Replicated disks (tükrözött lemezek)
    - Bináris (blokkos), fálj szintű (cluster file system), vagy alkalmazás szintű replikáció (pl. adatbázis szintű változások); többféle teljesítmény opció
- Topológia:
  - Pár, N+1, N+I, gyűrű topológiák

Szoftver architektúra

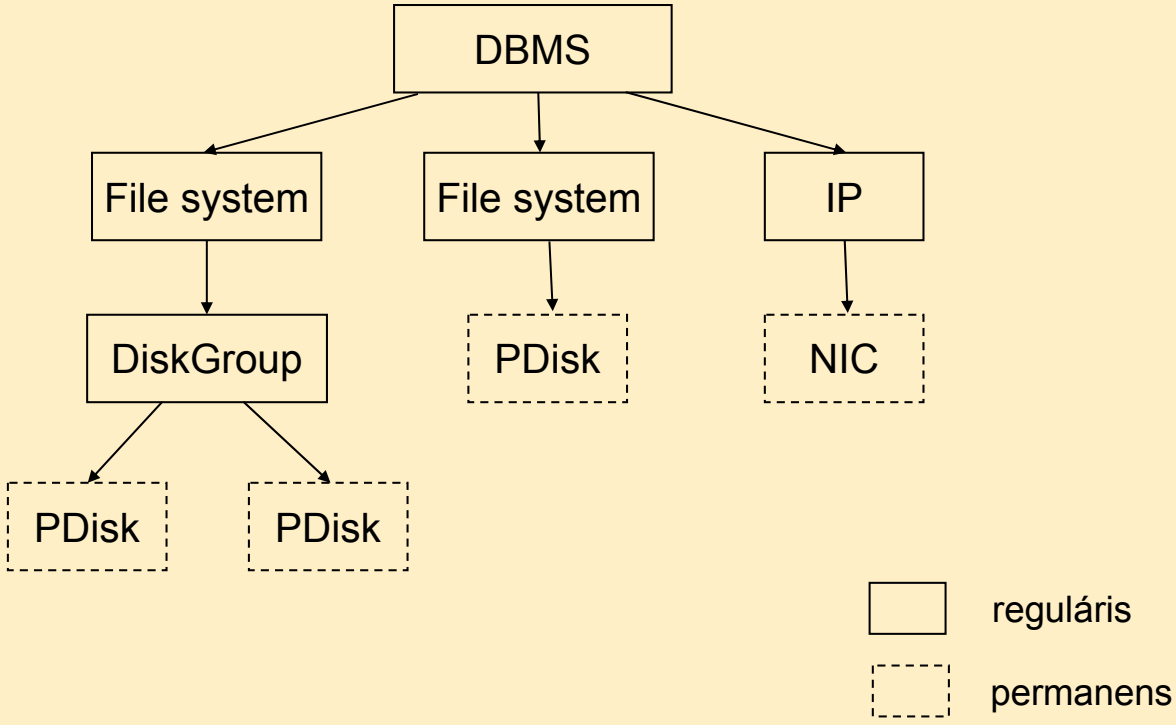
- Operációs rendszer
  - Általában nem fürt specifikus
  - Keretprogrammal integrálható
- Fürt keretprogram (HA keretprogram): Fürt menedzselése
  - Hibadetektálás
    - Heartbeat, challenge-response protokoll
  - Hibakezelés:
    - Újraindítás (tranziens hiba esetén, STOMITH)
    - Feladatátvétel (failover), feladat-visszavétel (failback)
  - Értesítések (újabb hiba már szolgáltatás kimaradást okozhat)
  - Újrakonfigurálás (pl. terheléselosztás)
    - Átkapcsolás (switchover), visszakapcsolás (switchback), ki/beléptetés
- Alkalmazói keretprogram
  - Alkalmazás-specifikus hibadetektálás
  - Alkalmazás szintű állapotmentés, helyreállítás, átkonfigurálás

Failover és failback

- Failover és failback alapja: Erőforrás függőségi fa
  - Erőforrások: hardver, szoftver, vagy alkalmazás
    - Erőforrás csoportok: Hierarchikus egymásra épülés (ld. példa)
  - Erőforrás típusok:
    - Reguláris: ki/be kapcsolható (pl. mount/umount)
    - Permanens: nem kapcsolható ki (pl. hálókártya)
  - Erőforrások kezelése:
    - Szkriptek: bekapcsolás, kikapcsolás (regulárishoz), monitorozás
- Alkalmazás jellemzői:
  - Használt erőforrás csoportok (függőségi fa)
  - Hol indítható, automatikusan indítható-e egy-egy erőforrás
- Alkalmazás failover és failback:
  - Leállítás: Erőforrások kikapcsolása – felülről lefelé
  - Elindítás: Erőforrások bekapcsolása – alulról felfelé
- Erőforrás szintű hiba detektálása esetén:
  - Alkalmazás failover, ahol a függőségi fában szerepel

Erőforrás függőségi fa

- Egy példa:

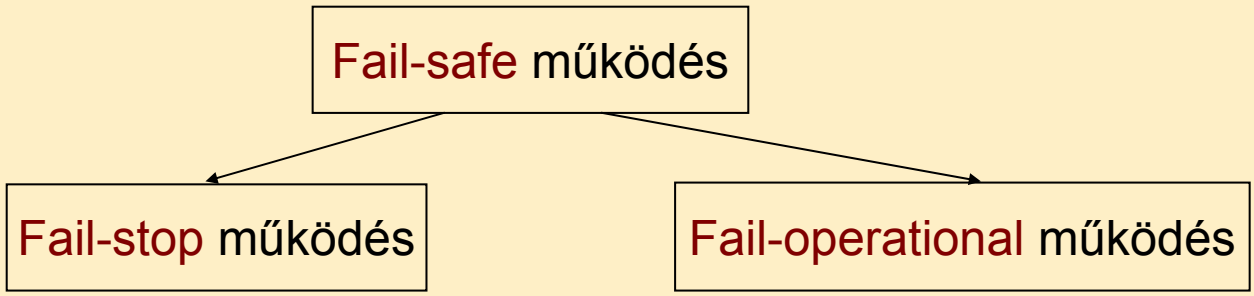


Jellegzetes problémák fürtökben

- Tudathasadás (split brain)
  - A fürt felbomlása (partíciók)
  - Quorum (többség) képzése szükséges
    - Szerverek többsége (páratlan számú szerver esetén)
    - Tanúlemez (witness disk) birtoklása
    - Tanú fájl (witness file) birtoklása (kijelölt fájlmegosztás, távoli lehet)
    - Általános: Szavazatok hozzárendelése szerverekhez, tanúlemezhez, tanú fájlhoz
- Amnézia (amnesia)
  - Meghibásodott, majd javított szerver visszaléptetése, közben az aktív is meghibásodik: aktuális konfiguráció elveszhet
  - Megoldás: Fürt konfigurációt közös adattárolóra írni
    - Fürt adatbázis: Konfiguráció, változás napló (quorum logging)
- Szoftverfrissítés
  - Gördülő frissítés (rolling upgrade): Kiléptetés, frissítés, visszaléptetés sorozat

Architektúrák biztonságkritikus rendszerekben

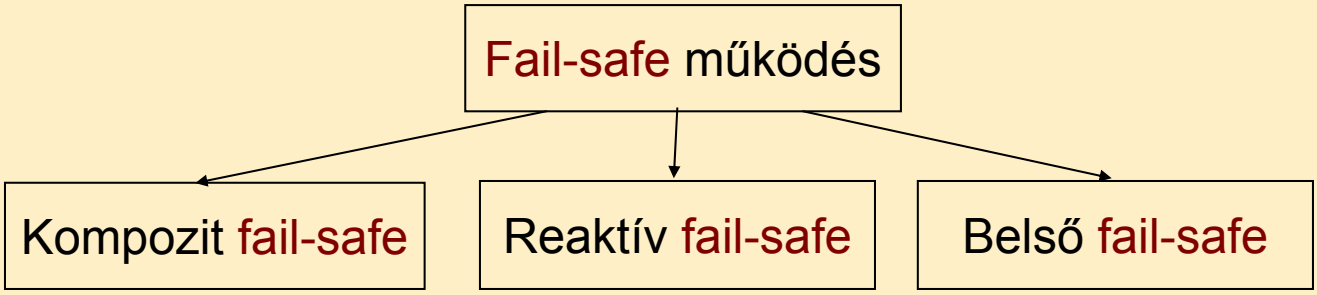
Célkitűzés



- A megállás (lekapcsolás) biztonságos állapot
- Detektált hiba esetén le kell állítani a rendszert
- Hibadetektálás a kritikus feladat

- A megállás (lekapcsolás) nem biztonságos állapot
- Detektált hiba esetén is szükséges szolgáltatás
  - teljes, vagy
  - csökkentett (degradált)
- Hibatűrés szükséges

Általános alapelvek



- Minden funkciót megvalósít legalább 2 független komponens
- A továbblépéshez (többségi) egyetértés szükséges

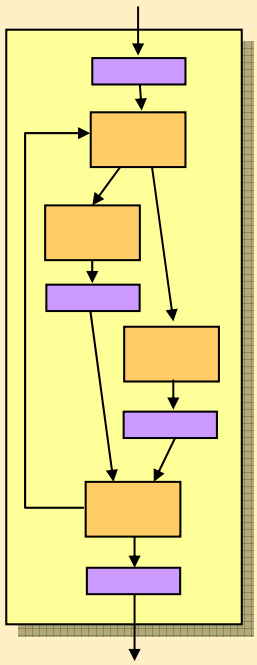
- Minden funkció mellé rendelhető független hibadetektálás
- A detektált hiba hatása negálható

- Minden hibamód veszélytelen
- „Természeténél fogva biztonságos”

Architektúrák biztonságkritikus rendszerekben: Jellegzetes megoldások fail-stop működéshez

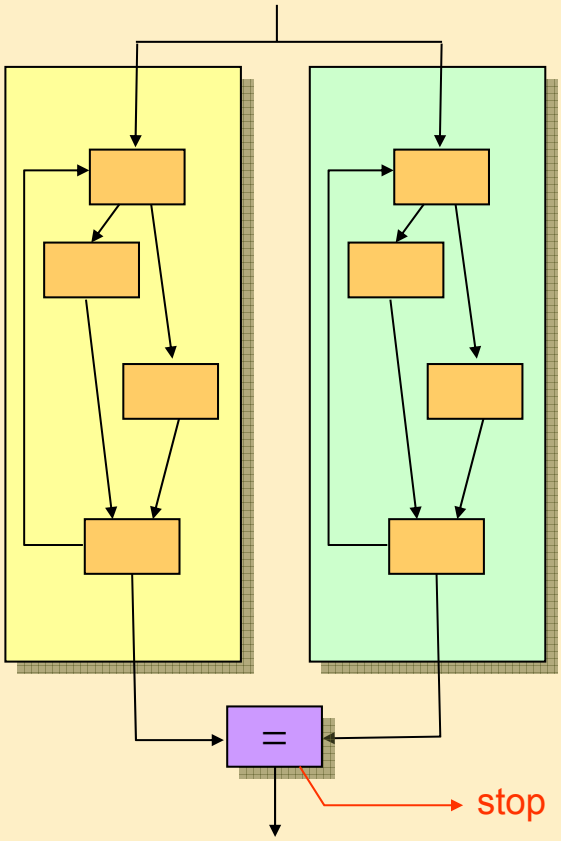
Egycsatornás feldolgozás öntészttel

- Egy feldolgozási folyamat
- Ütemezett hardver öntesztek
  - Induláskor részletes önteszt
  - Futás közben: Lappangó állandósult hibák detektálása
  - Ajánlott többféle módon
- Rendszeres szoftver önellenőrzés
  - Végrehajtási utak ellenőrzése
  - Alkalmazásfüggő hibadetektálás
- Hátrányok:
  - Önteszt hibafedése korlátos
  - Hibakezelés megvalósítása kérdéses (pl. lekapcsolás)



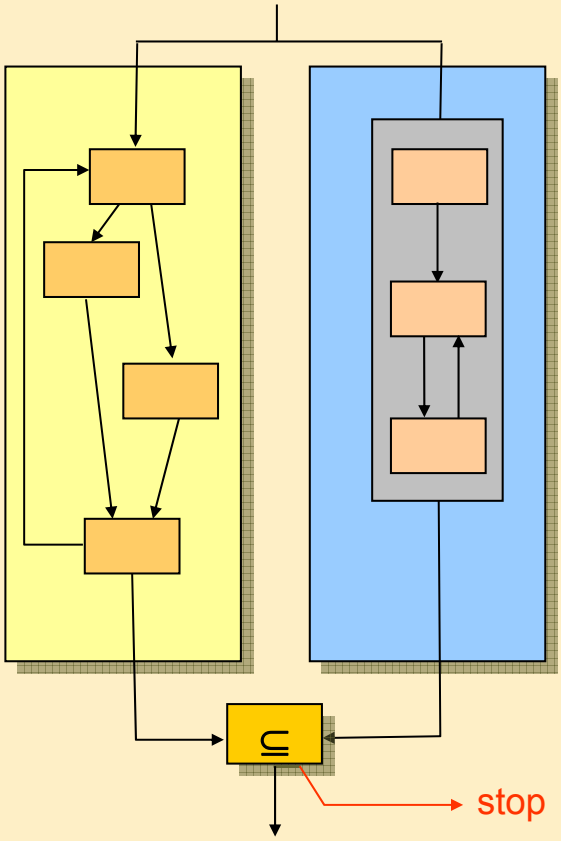
Két- vagy többcsatornás feldolgozás

- Két vagy több feldolgozási csatorna
  - Közös bemenet
  - Kimenetek komparálása
  - Eltérés esetén leállás
- Nagy hibafedés
- Komparátor kritikus elem
  - De egyszerű!
  - Kiváltása kódolt feldolgozással
- Hátrányok:
  - Közös eredetű hiba?
  - Hosszú lappangási idő

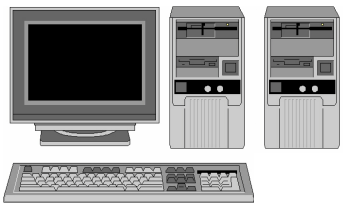


Kétcsatornás feldolgozás független ellenőrzéssel

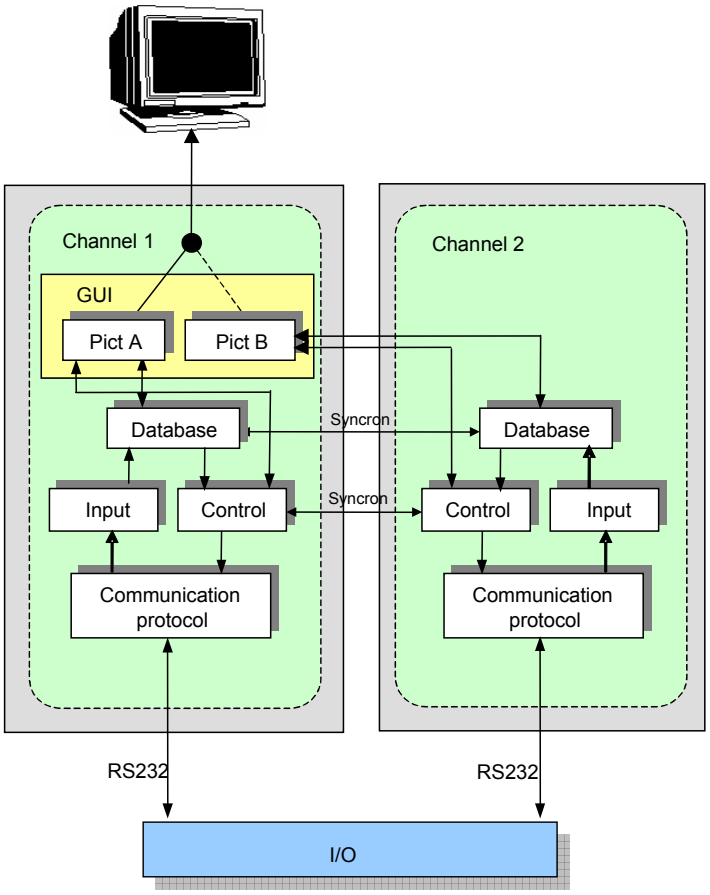
- Független csatorna
  - „Safety bag”: csak biztonsági ellenőrzés
  - Eltérő tervezés
  - Megengedhető viselkedés ellenőrzése
- Példa:
  - Alcatel (Thales) Elektra biztosítóberendezés
  - Szabályok az elsődleges csatorna működésének ellenőrzésére



Példa: SCADA rendszer

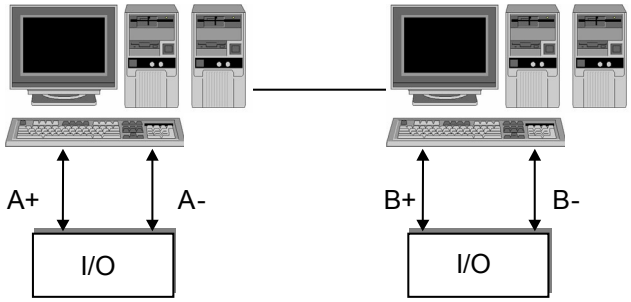


- Két csatorna
- Váltakozó bitmap megjelenítés (az operátor komparál :)
- Szinkronizáció: Belső hibadetektálás (mielőtt a kimenetre kerülne)



Telepítési opciók

- Két csatorna ugyanazon a szerveren
  - Statikusan linkelt szoftver modulok
  - Időben, memóriában és diszken elkülönülő végrehajtás
  - Diverz adattárolás
    - Bináris adatokra (jelek): Inverz adatábrázolás
    - Különböző adatbázis indexelés
- Két csatorna különböző szervereken
  - Szinkronizáció dedikált belső hálózaton
- Rendelkezésre állás növelése (hibatűrés):
  - Kétszer „2-ből 2” séma



Hibadetektálás

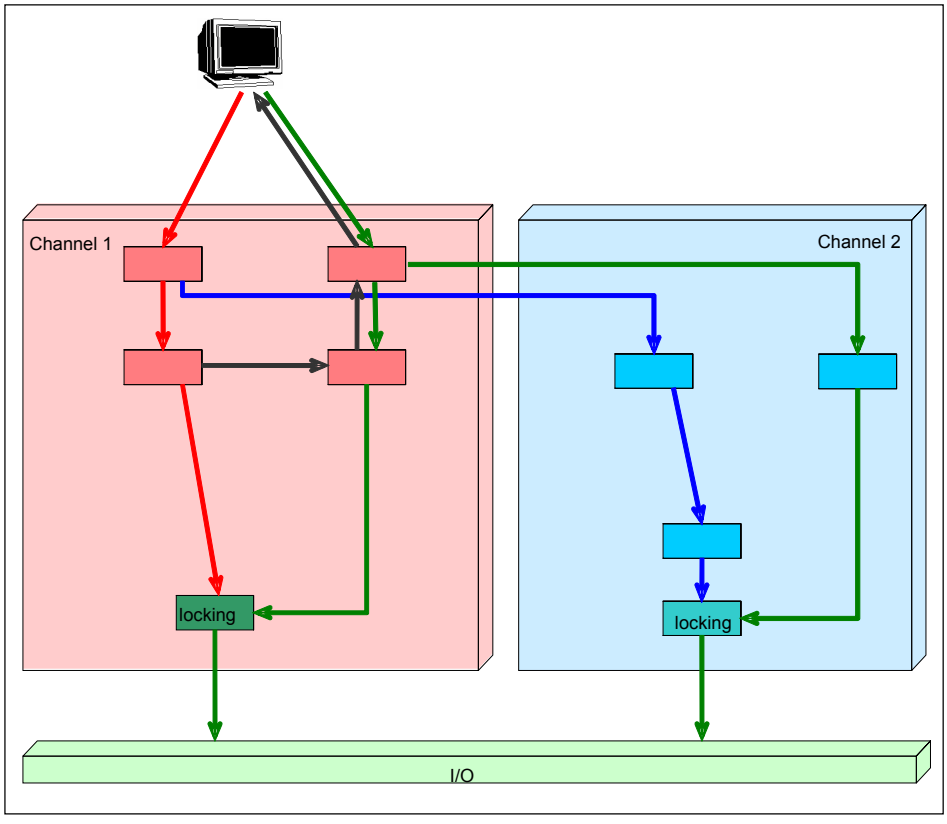
Véletlen hardver hibákra működés közben:

- Csatornák komparálása: Operátor illetve I/O
  - Operátornak: Villogó RGB-BGR szimbólum jelzi a frissítést
- Watchdog processz
  - Többi processz futásának ellenőrzése
- Az adatbázis tartalmának rendszeres összehasonlítása
  - Lappangó hibák detektálása

Szándékolatlan vezérlésre, közös módusú hibákra:

- Háromfázisú parancskiadás
  - Előkészítés (zárolva), visszaolvasás, független jóváhagyás
  - Diverz modulok a zárolásra

Háromfázisú parancskiadás

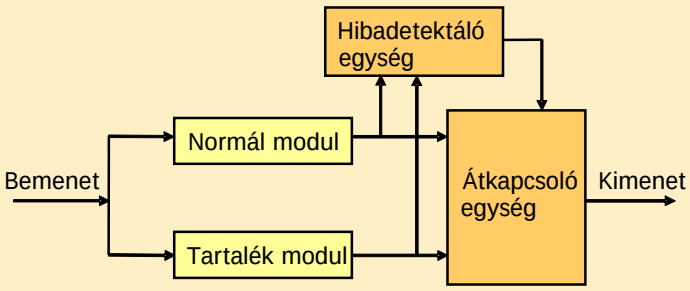


Architektúrák biztonságkritikus rendszerekben: Jellegzetes megoldások fail-operational (hibatűrő) működéshez

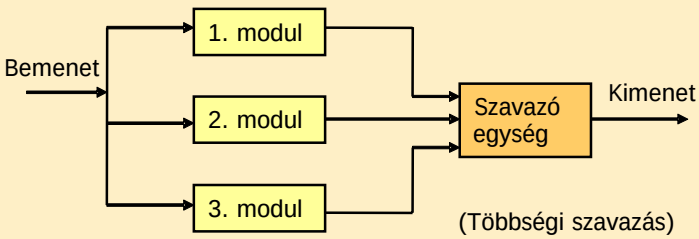
Ismétlés: Állandósult hardver hibák kezelése

Többszörözés:

- Kettőzés:
  - Hibatűrés: Diagnosztikai támogatás és átkapcsolás



- TMR: Triple-modular redundancy
  - Hiba maszkolása többségi szavazással
  - Szavazó kritikus elem (de egyszerű)



- NMR: N-modular redundancy
  - Hiba maszkolása többségi szavazással
  - Missziós idő túlélése nagyobb esélyű, utána javítás jöhet
  - Repülőgép fedélzeti eszközök: 4MR, 5MR

Ismétlés: Szoftver tervezési hibák kezelése

- Redundáns modulok: Eltérő tervezés szükséges
  - Variánsok: azonos specifikáció, de
    - eltérő algoritmus és/vagy adatstruktúrák (diverzitás)
    - más fejlesztési környezet, programnyelv
    - elszigetelt fejlesztés

    az azonos hibák bekövetkezésének csökkentésére
- Variánsok végrehajtásának technikái:
  - Aktív statikus redundancia:
    - N-verziós programozás (NVP: N-version programming)
  - Passzív redundancia:
    - Javító blokkok (RB: Recovery block)
  - Aktív dinamikus redundancia:
    - N-önellenőrző programozás (NSCP: N-self-checking programming)
  - Adaptív redundancia:
    - Önkonfiguráló optimista programozás (SCOP: Self-configuring optimistic programming)

Hardver és szoftver hibák együttes kezelése

Hibrid architektúrák jelölése:

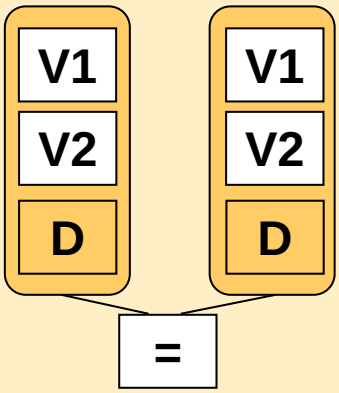
- Szoftver hibatűrés típusa /
  tolerált hardver hibák száma /
  tolerált szoftver hibák száma

Példák hibrid architektúrákra:

- RB/1/1
- RB/2/1
- NVP/2/1
- NSCP/1/1

Hibrid architektúrák 1.

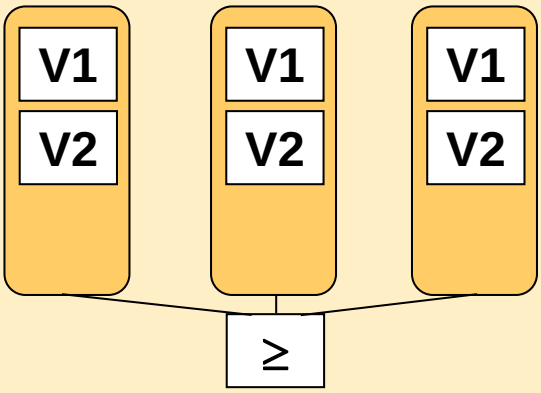
RB/1/1



RB/1/1

- Elfogadhatósági teszt után komparálás
- Diagnosztikai ellenőrzés után a hibás lekapcsol

RB/2/1

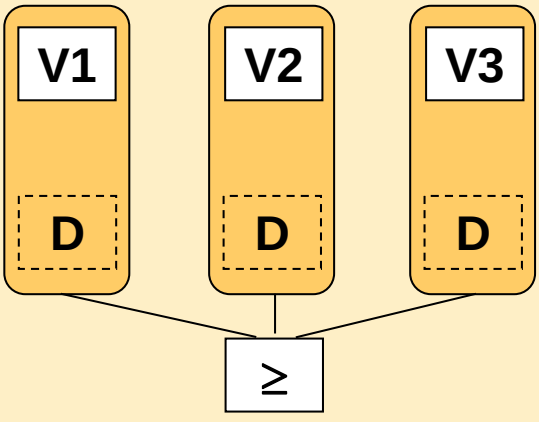


RB/2/1

- Elfogadhatósági teszt után szavazás
- Ismételten eltérő lekapcsol (RB/1/1 marad)

Hibrid architektúrák 2.

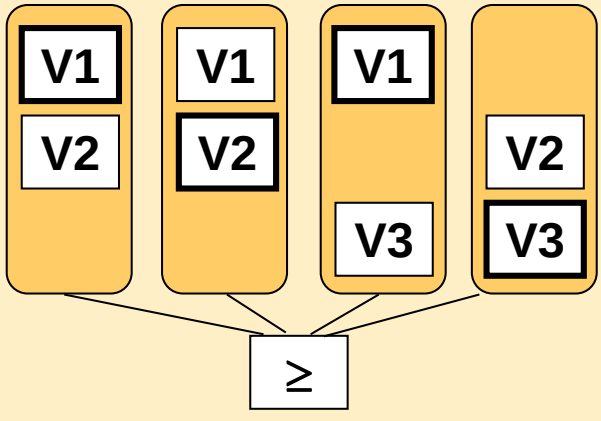
NVP/1/1



NVP/1/1

- Variánsok közötti szavazás
- Ismételten eltérő lekapcsol (komparálás marad)

NVP/2/1



NVP/2/1

- 4 variáns esetén szavazás
- Ismételten eltérő lekapcsol (NVP/1/1 konfigurálható)