



Kritikus architektúrák laboratórium

Megbízhatósági modellezés

Mérési feladatok

Készítette: Vörös András

Utolsó módosítás: 2016. március 29.

Verzió: 1.0

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

1 Bevezető

A mérés során a megbízhatósági modellezésben rejlő lehetőségekkel fogunk megismerkedni. Először a legfontosabb nem állapotter alapú modellezési lehetőségeket tekintjük át, majd az infrastruktúra modelljét Petri hálók segítségével is megvizsgáljuk.

A mérés során használt eszközök:

- **Sharpe**
 - hibafa analízis
 - megbízhatósági blokk diagram
- **TimeNET**
 - időzített sztochasztikus Petri hálók

Fontos megjegyezni, hogy a TimeNET eszköz nem csak megbízhatósági analízisre használható, különböző teljesítmény (rendszer áteresztő képessége, teljesítménye, bufferek telítődési valószínűsége, rendszerek szaturációja, stb...) és egyéb rendszerjellemzők vizsgálatára is alkalmas.

A mérésekhez használt eszközök a virtuális gépen az asztalon található ikonokkal érhetőek el, továbbá a SHARPE eszköz felhasználói dokumentációja is ott található.

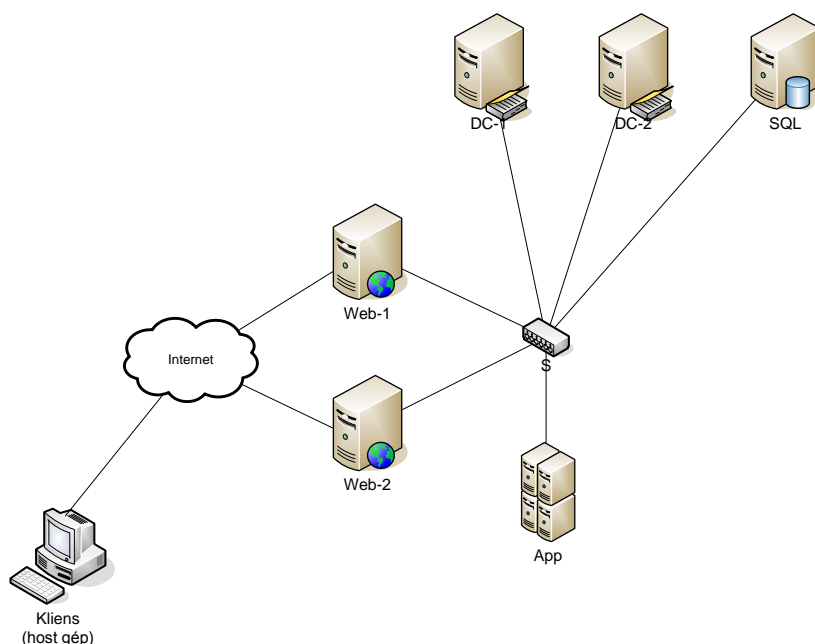
2 Jegyzőkönyv

A mérési jegyzőkönyvben minden döntést, amit a feladat esetleg elégtelen specifikációja miatt hoztatok, indokoljatok! A jegyzőkönyvben szerepelnie kell többek között:

- az elkészült modelleknek (magyarázattal),
- a számítások végeredményeinek is,
- a mérések értékelésével együtt!

3 Mérési infrastruktúra

A mérés során a következő ábrán látható infrastruktúra megbízhatósági modelljét fogjuk elkészíteni:



1. ábra: Infrastruktúra

Az ábrán látható webszerverek (Web-X), tartományvezérlők (DC-X) redundánsan vannak kötve a nagyobb megbízhatóság érdekében. Az alkalmazás szerverből (App), az adatbázis szerverből (SQL), és a hálózati eszközből (S) egy található a rendszerben.

Komponens	Elromlás (nap)	Javítás (óra)
Webszerver	20	24
Tartományvezérlő	25	
SQL szerver	30	48
Alkalmazás szerver	35	72
Switch (S)	60	12

4 SHARPE feladatok

1. Határozzuk meg az egyszerű hibapontokat a rendszerben!
2. Rajzoljuk fel a rendszer hibafáját, ahol a legfelsőbb szintű hiba a szolgáltatás kiesése!
3. Számoljuk ki a rendszer megbízhatóságát a fent megadott adatokkal!
4. Ábrázoljuk a megbízhatóságot az idő függvényében!
5. Keressünk az interneten elérhető forrásokból a fent látható komponensekhez tipikus értékeket (megbízhatóság, jellemző hibák, stb...)! Legalább egy komponensre vizsgáljuk meg az elérhető forrásokat!
6. Próbáljuk ki a SHARPE nyújtotta összes analízis lehetőséget! A fent felvázolt modellre melyik milyen értéket ad? Miért?
7. Rajzoljuk fel a fenti rendszer megbízhatósági blokk diagramját a SHARPE eszköz segítségével!

5 TimeNET

Tekintsük a fentebb látott infrastruktúra modellt. Rajzoljuk fel a rendszer Petri hálóját tartományvezérlők nélkül (DC-X)!

1. Számoljuk ki a modell rendelkezésre állását!
2. Használjuk javításra a következő módszert: amikor meghibásodik az adatbázis szerver vagy az alkalmazás szerver, akkor a tartalékból kiveszünk egy szert, amit azonnal beteszünk a helyére (modellezhetjük úgy, hogy az új gép azonnal helyettesíti a kiesőt). Az elromlott gépet (minden esetben) elküldjük javításra, amely átlagosan 3 napig tart. Hogyan változik a rendszer rendelkezésre állása, ha egy tartalék gépünk van. És ha kettő?
3. Költségoptimalizálás:
Rendelkezzenek a komponensek az alábbi éves fenntartási költségekkel:

Komponens	éves fenntartási költség (Ft)
Webszerver	200 000
SQL szerver	500 000
Alkalmazás szerver	1000 000
Switch (S)	20 000

Melyik komponensből lenne érdemes még egyet beszerezni, ha a cég ezen infrastruktúrájának egy napi kiesése 1000 000 Ft-ba kerül?

Milyen módosításokkal lehetne az éves költségeket minimalizálni (ha minden komponensből legalább egy kell a szolgáltatás működéséhez)?