A hibakezelés tesztelése: Hibainjektálás

Majzik István és Micskei Zoltán
Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék
http://www.inf.mit.bme.hu/

Tartalomjegyzék

Motiváció

- Hibakezelés tesztelése
  - Hibakezelési funkciók megvalósításának ellenőrzése
  - Példa: Biztonságkritikus rendszerek
    - Fail stop rendszerek: Hibadetektálás és leállás
    - Fail operational rendszerek: Hibatűrés megvalósítása
- Megvalósítási lehetőségek
  - Valós hibák hatásának megfigyelése (naplózás): Nehézségek véletlen hibák esetén:
    - Vagy hosszú idejű működtetés,
    - vagy nagyszámú megfigyelés szükséges
  - Hibainjektálás: Valóságban várható hibák bevitele
    - Prototípuson elvégezhető
    - Valós hibák „gyorsított módon” előidézhetők
    - Célzott hibainjektálás adott hibamodell esetén

A hibainjektálás további alkalmazásai

- Megbízhatóság értékelése
  - Véletlen hibák injektálása (nem célzott)
  - Statisztikai jellemzés
    - Hibadetektáló technikák hibafedése
    - Kritikus hatású hibák aránya
    - Megbízhatóság, rendelkezésre állás felmérése
- Szoftver tesztelés hatékonyságának értékelése
  - Hibabeültetés
  - Becslés:

$$\frac{\text{megtalált beültetett hibák száma}}{\text{összes beültetett hibák száma}} \approx \frac{\text{megtalált valódi hibák száma}}{\text{összes valódi hiba száma}}$$

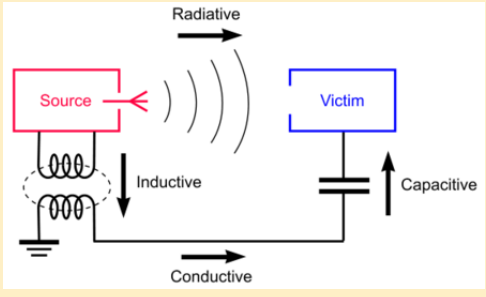
  - Sok feltétel kell a jó becslőhöz:
    - Beültetett hibák reprezentatívak, eloszlásuk megfelel a valódi hibákénak, …

Tartalomjegyzék

- Motiváció
- A hibainjektálás megvalósítása
  - Hardver hibainjektálás
  - Szoftver hibainjektálás
  - Modell alapú hibainjektálás
- A hibainjektálás tervezése
  - Hibamodellek
  - Absztrakciós szintek
  - A végrehajtás szempontjai
- Összefoglalás

Hardver hibainjektálás

- Célkitűzés:
  - Valós hibaok (fault) injektálása és hatásának felmérése
    - Korlát: Milyen közel mehetünk a valós hibaokhoz?
- Technológiák és eszközök
  - Jelek közvetlen befolyásolása: RIFLE, GOOFI környezet
  - Belső állapot megváltoztatás (pl. JTAG interfészen)
  - Tápfeszültség befolyásolása (tüskék, kimaradás)
  - Besugárzás (nehéz-ion, neutron)
  - EMI: Electromagnetic interference



Szoftver hibainjektálás

- Célkitűzés:
  - Hardver hibákra: Hibaállapot (error) injektálása (SWIFI)
    - Korlát: Milyen jól tudjuk felmérni a hatásokat?
  - Szoftver hibákra: Mutáns kódrészlet (bug) injektálása
    - Korlát: Milyen becslésünk van a tipikus hibákról?
- Megvalósítási technológiák
  - Futtató rendszer által nyújtott támogatás
    - Unix rendszerhívás: ptrace() – memória, regiszterek, verem elérhető
    - JVM Tool Interface: Speciális felműszerezés (ClassFileLoadHook)
  - Kód mutáció:
    - Közvetlen: Forráskód mutációs operátorok definiálása
    - AOP technológiák: Forráskód vagy bájtkód szinten
- Eszközök
  - Hardver hibahatások emulációja: FIAT, FERRARI, FTAPE
  - Alacsony szintű emuláció: DOCTOR, Xception
  - Kód mutációs eszközök: FINE, DEFINE, G-SWFIT
  - Protokoll rétegek hibái: ORCHESTRA, Neko, WS-FIT

Modell alapú hibainjektálás

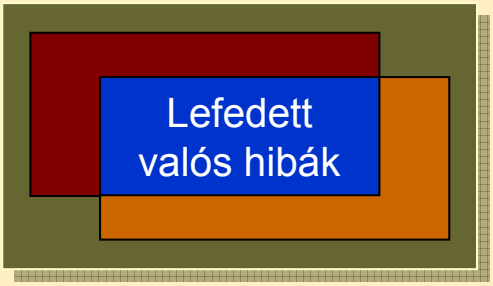
- Célkitűzés:
  - Nem közvetlenül a tesztelés része
  - Valós (hardver, szoftver) hibainjektálás optimalizálása modell alapú előzetes értékeléssel
    - Biztonsági szempontból kritikus hatású hibák azonosítása: Ezeket célszerű tényleges teszteléssel vizsgálni
    - Hatás nélküli hibák kiszűrése
  - Modell szinten injektált hibák alapján történő hatásfelmérés
    - Korlát: Mennyire jól modellezhetők a hibák és hatások?
- Eszközök
  - VHDL, HDL szintű modellezés: FOCUS, MEFISTO
  - Komponens (CPU, diszk, memória) szintű modellezés: DEPEND
  - Általános modell alapú vizsgálatok: Modellellenőrzés, szimuláció

Tartalomjegyzék

- Motiváció
- A hibainjektálás megvalósítása
  - Hardver hibainjektálás
  - Szoftver hibainjektálás
  - Modell alapú hibainjektálás
- A hibainjektálás tervezése
  - Hibamodellek
  - Absztrakciós szintek
  - A végrehajtás szempontjai
- Összefoglalás

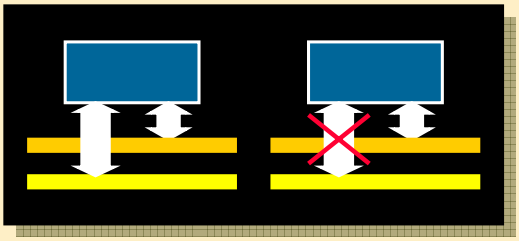
Hibainjektálás tervezése

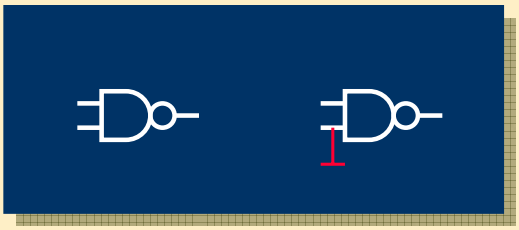
- Hibamodell:
  - Egy kép a rendszerben tesztelendő hibákról
  - Hardver hibák, szoftver hibák
- Hibamodell jósága
  - Feltételezett és valós hibamodell viszonya
  - Redundáns elemek: növelik a költséget
  - Nem lefedett hibák: rontják a tesztelés minőségét

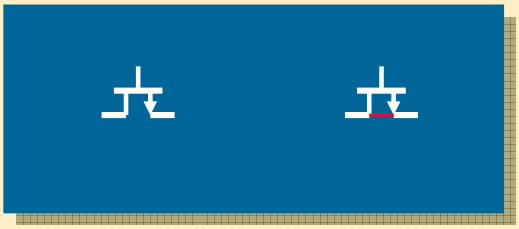


A hibainjektálás absztrakciós szintjei

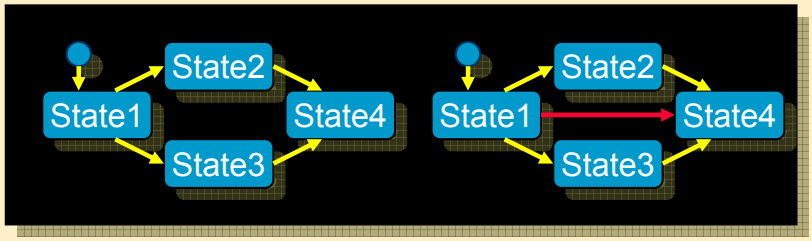
Hardver:







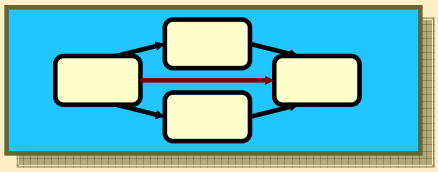
Szoftver:



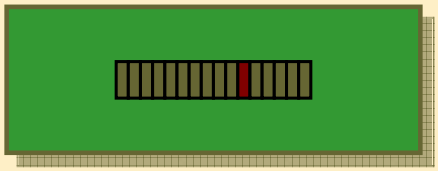

```
if (EVENT1 == event)
     state = STATE1;
if (EVENT2 == event)
     state = STATE2;
```

```
cmp eax, 1
je LABEL1
cmp eax, 2
je LABEL2
```

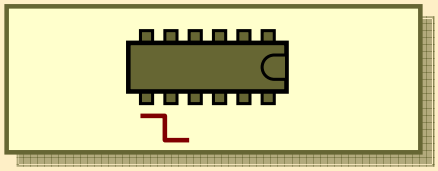
A hibainjektálás megvalósítása (hardver)



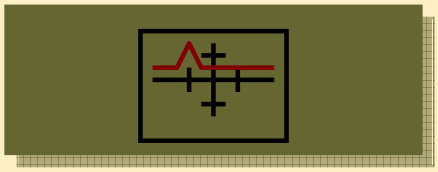
Hibaszimuláció a forráskódban vagy modell szinten a fejlesztőeszközben



Regisztertartalom módosítása, memóriakép felülírása



Síneken haladó vagy áramkörök lábán megjelenő jelek módosítása



Radioaktív sugárzás, ioninjektálás, tápfeszültség zavarása, hőmérséklet

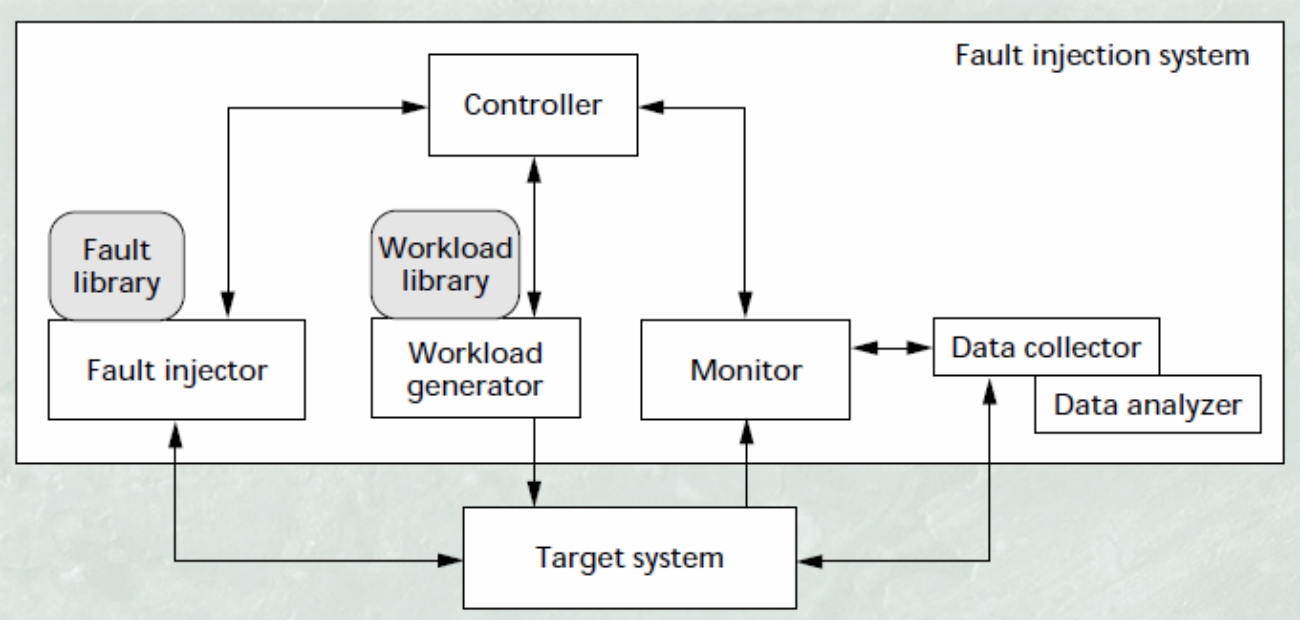
Ár, valósághűség

Hibainjektálási lehetőségek áttekintése

- Hardver hibainjektálás
  - Hibaok injektálása közelíthető
    - Besugárzás, elektromágneses zavarok, hőmérséklet
    - Tápfeszültség tüske, jelvezetékek leragadása, összeragadása, …
  - Rugalmatlan, drága, de valósághűsége jobb
- Szoftver hibainjektálás
  - Hibaállapot injektálása (hardver hibaok hatása)
    - Processzor regisztertartalom, memóriatartalom, fájlok, üzenetek,…
  - Mutáció bevitele (vezérlés, adatkezelés)
  - Rugalmas, olcsóbb, de valósághűsége kisebb
- Modell alapú hibainjektálás
  - Optimalizálásra alkalmazható
  - Tipikusan a komponens szintű hibajelenségek modellezése
    - Funkciók, interakciók megváltozása
  - Tervezési fázisban végrehajtható, de a valósághűség itt is kérdéses (legmagasabb absztrakciós szintű)

Teszt környezet hibainjektáláshoz

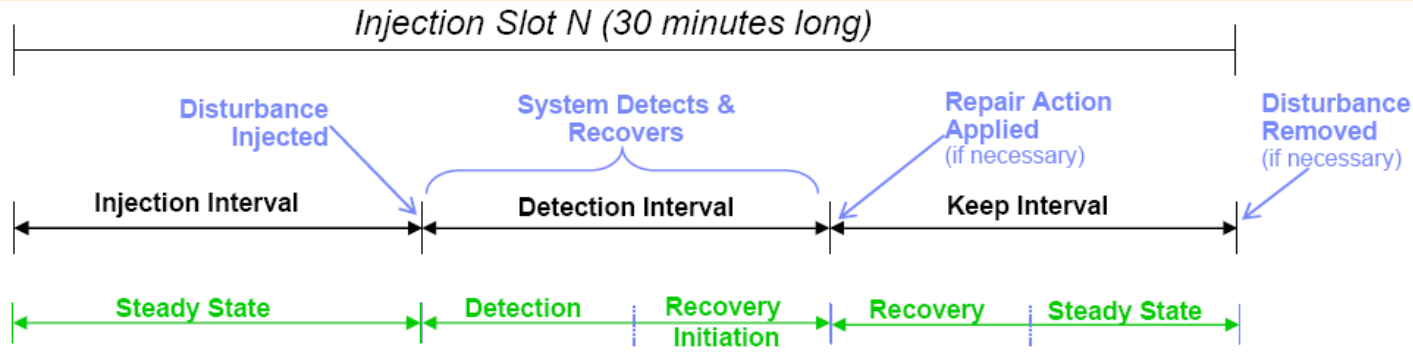
Általános blokkvázlat:



A hibainjektálás ütemezése

Egy teszt felépítése:

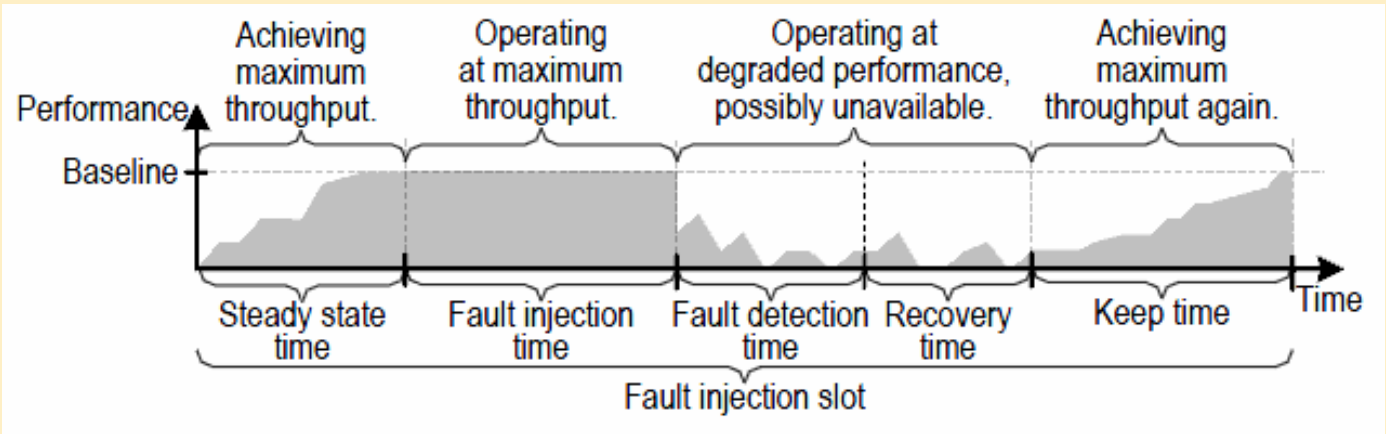
1. Normál állapot elérése (munkaterhelés)
2. Hiba injektálása
3. Hatások (teszt eredmények) felmérése
4. Hiba eltávolítása, alapállapot elérése



A hibainjektálás ütemezése

Egy teszt felépítése:

1. Normál állapot elérése (munkaterhelés)
2. Hiba injektálása
3. Hatások (teszt eredmények) felmérése
4. Hiba eltávolítása, alapállapot elérése

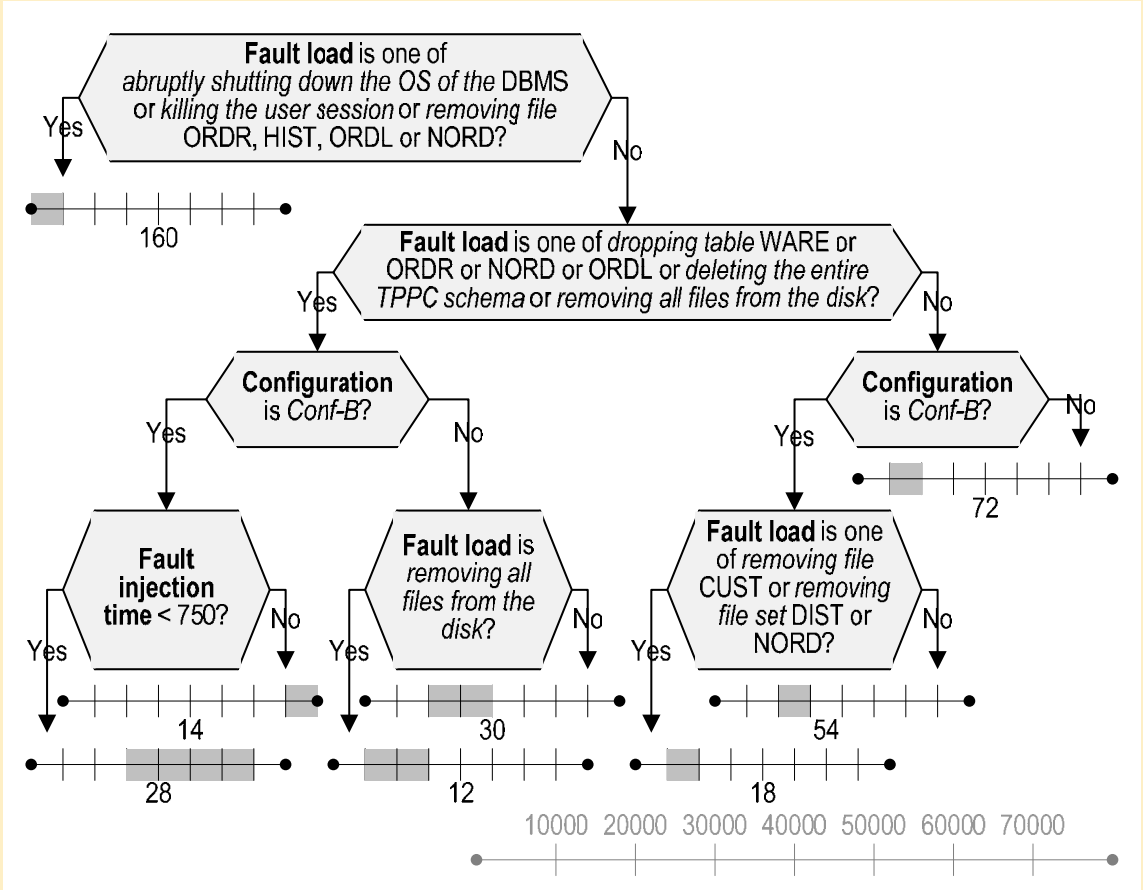


Az eredmények kiértékelése

- Nagy mennyiségű adat feldolgozása
  - Használható technológiák: OLAP, adatbányászat
  - Rejtett összefüggések is felismerhetők
- Példa (ld. következő dia)
  - Adatbáziskezelő tesztelése hibainjektálással
    - Helyreállítás idejének mérése
  - Eredmények osztályozása
    - Helyreállítás idejének felosztása 8 intervallumra (0 és 89999 sec között)
    - Hogyan függ az intervallum a hibainjektálás paramétereitől?
    - Adatbányász eszköz megvizsgálja, mi az összefüggés a hibainjektálás paraméterei és az intervallumok között (osztályozás)
    - Egy döntési fa nyerhető ki az osztályozás szempontjairól: fontosabb (meghatározó) szempontok a döntési fában feljebb
  - A döntési fában lévő döntések megadják a paraméterek és a helyreállítás idejének összefüggését (mitől függ?)

Példa: Összefüggések felderítése adatbányászattal

A hiba-injek-tálási paramé-terek és a helyre-állítás idejének össze-függése



Összefoglalás

- Motiváció
  - Hibakezelés tesztelése
- A hibainjektálás megvalósítása
  - Hardver hibainjektálás
  - Szoftver hibainjektálás
  - Modell alapú hibainjektálás
- A hibainjektálás tervezése
  - Hibamodellek
  - Absztrakciós szintek
  - A végrehajtás szempontjai
- Összefoglalás