



Hibadiagnosztika elosztott rendszerekben

Bevezető

A mérés során megvalósítandó feladatok két csoportba oszthatók: determinisztikus és valószínűségi diagnosztikai algoritmusokra.

- A **determinisztikus algoritmusok** adott feltételek teljesülése esetén képesek teljes és konzisztens diagnosztikai képet előállítani. *Teljesnek* akkor nevezünk egy diagnosztikai eredményt, ha a rendszer minden egysége kapott valamilyen (hibátlan/hibás) minősítést, *konzisztensnek* pedig akkor, ha minden egység valós és az algoritmus által feltárt hibaállapota megegyezik. A teljes és konzisztens diagnosztika eléréséhez teljesítendő további feltétel(ek)re példa a mérési segédletben tárgyalt, a hibás egységek számára adott felső korlát, az ún. t -korlát.
- A **valószínűségi algoritmusok** nem garantálnak biztosan teljes és konzisztens megoldást, de nagy valószínűséggel helyes eredményt szolgáltatnak. A tévedés lehetőségért cserébe ezek a módszerek csak a teszteredményekből kinyert információt használják fel, azaz nem igényelnek a rendszerre nézve olyan további előírásokat, mint a t -korlát a determinisztikus módszereknél.

A valószínűségi algoritmusok lehetséges diagnosztikai tévedései három csoportba sorolhatók:

1. Az *ismeretlen egység* egy olyan processzort jelent, amelynek hibaállapotát az algoritmus nem tudta meghatározni. Ebben az esetben a diagnosztika nem teljes.
2. *Jóindulatú tévedés* esetén egy hibátlan processzort hibásnak minősítünk. Így csökkentjük ugyan a rendelkezésre álló erőforrások számát, de a biztonság javára tévedünk.
3. *Rosszindulatú tévedés* esetén egy hibás processzort hibátlannak minősítünk. Ekkor a rendszerből nem távolítjuk el a hibás komponens, ami hibás adatokkal és hibaterjesztéssel az egész folyamat hibáját okozhatja. Az erőforrások számát nem csökkentjük.

A mérési feladat

A mérési feladat a következő pontban leírt determinisztikus diagnosztikai algoritmus megvalósítása egy C nyelven megírt diagnosztikai szubrutin formájában. A szubrutint a szimulációs környezetbe ágyazva le kell fordítani, majd az algoritmus helyességét próbafuttatásokkal ellenőrizni. (A futtatások eredményét statisztikailag értékelve a mérési jegyzőkönyvben is fel kell tüntetni!)

A mérés kiértékelése

A determinisztikus algoritmusok az adott topológiára érvényes t -korlát értékével megegyező számú hibás egységig helyesen (teljes és/vagy konzisztens módon) *kell* működniük. A t -korlát felett azonban már nincs erre garancia. Ezért a determinisztikus algoritmusok ellenőrzésekor:

1. Több pontban elvégzett mérések alapján (adott rendszer méret esetén növekvő hibaszám, vagy adott számú hiba mellett növekvő rendszer méret, stb.) igazolnia kell, hogy a t -korlátig az algoritmus nem téved!

2. Ha ez teljesült, adjon statisztikát (táblázatos és diagram alakban) a t -korlát feletti hibaszámokra a jóindulatú és rosszindulatú diagnosztikai tévedések (és ha van, az ismeretlen egységek) alakulásáról.

Szorgalmi feladat: ha tud, adjon ötleteket az adott algoritmus továbbfejlesztéséhez (ha van ideje, valósítsa is meg)! Cél, hogy a diagnosztikai hatékonyság növekedjen (ilyen lehetőség pl. a felfedezett ellentmondások alapján meghozható biztos következtetések felhasználása).

A megvalósítandó algoritmus leírása

Maestrini és Chessa (konzisztens, de nem teljes) algoritmusában kihasználja azt, hogy a rendszerekben többnyire léteznek *kölcsönös* tesztek. Ha ezeket a teszteredményeket páronként tekintjük, akkor több információ szűrhető le belőlük, mint az egyedülálló teszteredményekből, ugyanis:

u_i teszteli u_j -t	u_j teszteli u_i -t	Következtetés
$a_{ij} = 0$	$a_{ji} = 0$	u_i és u_j azonos hibaállapotúak
$a_{ij} = 0$	$a_{ji} = 1$	u_i biztosan hibás
$a_{ij} = 1$	$a_{ji} = 0$	u_j biztosan hibás
$a_{ij} = 1$	$a_{ji} = 1$	u_i, u_j közül legalább egy hibás

Ezeket a következtetéseket felhasználva a diagnosztikai algoritmus első lépésében az egységeket három csoportba soroljuk:

1. a *biztosan hibás* vagy F (faulty) egységek,
2. a *gyanús* vagy D (dual) egységek, és
3. a *valószínűleg hibátlan* vagy H (healthy) egységek.

A biztosan hibás egységeket a fenti táblázat második és harmadik sora, a gyanús egységeket a táblázat utolsó sora jelöli ki. Az első lépésben tehát összegyűjtjük a 2. és 3. szabály alapján a biztosan hibás egységeket, majd az 1. szabály alapján további biztosan hibás egységeket találunk: az eddigi biztosan hibásakkal azonos hibaállapotúakat. A második lépésben a 4. szabály szerint gyanús egységeket, valamint az ezekkel azonos hibaállapotúakat keressük meg a megmaradó egységek között. Végül az F és D csoportokba addig be nem sorolt egységek lesznek a valószínűleg hibátlan, azaz H egységek.

Miután minden egységet besoroltunk a három csoport valamelyikébe, az egymással szomszédos valószínűleg hibátlan egységek összefüggő területeit ún. *aggregátumokba* gyűjtjük. Ha a H csoport üres, azaz nincsenek valószínűleg hibátlan egységek, akkor a gyanús, azaz D csoportra tesszük meg ugyanezt.

A legnagyobb méretű aggregátumba tartozó processzorokat (nevük „Fault-Free Core”, azaz hibátlan mag) hibátlanak tételezzük fel. Ezután a hibátlan magba tartozó egységek által elvégzett tesztek eredményeiből levonható következtetések segítségével minősítjük a többi egység hibaállapotát. A fennmaradó egységek az eredeti algoritmus szerint „ismeretlen egység” címkét kapnak az algoritmus végén, azaz a diagnosztika nem lesz feltétlenül *teljes*.

Valósítsa meg a Maestrini, Chessa algoritmust!