



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

Ingineria Calculatoarelor

21. Ipoteze ale defectelor unităților hardware înlocuibile

8.4 IPOTEZE ALE DEFECTELOR UNITĂȚILOR HARDWARE ÎNLOCUIBILE

Cei care dezvoltă *software*-ul sistemului de operare presupun, în mod tipic, că *server*-ele CPU, de intrare/ieșire și ale *controller*-ului de comunicații au semantici ale defectelor de tip cădere, că elementele de memorie au semantici ale defectelor de tip omitere la citire, că discurile au semantici ale defectelor de căutare și semantici ale defectelor de tip omitere la citire/scriere și că magistralele de comunicație și liniile de comunicație au semantici ale defectelor

de tip omitere sau omitere/performanță. Statisticile disponibile din acest domeniu arată că, pentru sisteme de procesare a tranzacțiilor comerciale *on-line*, caracterizate prin timpi de cădere de ordinul minutelor sau orelor pe an, aceste presupuneri ale defectelor sunt adecvate. Toate sistemele menționate anterior fac aceste presupuneri.

Astfel, semantici puternice ale defectelor *hardware* permit proiectanților sistemului să folosească tehnicile cunoscute de mascare ierarhică, pentru a masca defectele *server-ului hardware*, cum ar fi duplexarea stocării pentru mascarea pierderii datelor replicate pe două *server-e* de memorie sau *server-e* de disc, cu semantici ale defectului de tip omitere a citirii/scrierii, sau circuite virtuale pentru mascarea defectelor de tip omitere sau performanță a comunicației, prin folosirea *time-out*-urilor, secvențelor numerice, confirmărilor și reîncercărilor. Mai mult, când mascarea nu este posibilă, semantici puternice ale defectelor *server-ului hardware*, cum ar fi omiterea și căderea, permit programatorilor sistemului să asigure faptul că sistemul de operare și serviciile de comunicație pe care ei le-au implementat, prezintă semantici puternice ale defectelor.

De exemplu, **CPU** și *controller*-ele de disk cu semantici ale defectelor de tip cădere și semantici ale defectelor de tip omitere a citirii/scrierii, asigură un nivel ridicat al serviciului stocării stabile, cu operații de scriere care sunt atomice și care respectă căderile: orice întrerupere a scrierii, datorată unei căderi, este, mai degrabă, eliminată prin completare sau nu este efectuată deloc. Un astfel de serviciu al stocării stabile poate fi folosit de către alte *server-e* de nivel înalt pentru implementarea tranzacțiilor, prin bazarea pe tehnici de refacere a bazei de date, cum ar fi log-area înaintea scrierii și realizarea în două faze. În mod similar, serviciile de comunicație de nivel scăzut cu datagrame, cu semantici ale defectelor de tip omitere/performanță, permit implementarea circuitelor virtuale de nivel înalt, cu semantici ale defectelor de tip cădere.

În spatele acestor ipoteze restrictive referitoare la semantica defectelor unor unități *hardware* înlocuibile primare, complexitatea și costul detecției, diagnozei și refacerii din defectele elementare ale *server-ului hardware* cresc nivelele pe care mulți le consideră ca neacceptabile pentru multe aplicații comerciale în procesul de tranzacții *on-line* și în telecomunicații.

În scopul detectării defectelor în magistrale, linii de comunicație, memorii și *server-e* de disc, toate arhitecturile despre care am discutat anterior folosesc coduri detectoare de erori. Această tehnică de detectare a erorilor este bine înțeleasă. Există o literatură de specialitate bogată pentru codurile detectoare de erori, acest subiect părând să fi atins cu fermitate starea de maturitate. Pentru a detecta defectele în *server-ele hardware*, sisteme cum ar fi IBM, VAX și sisteme de vârf TANDEM, folosesc coduri detectoare de erori, în timp ce sistemele mai noi, cum ar fi STRATUS, SEQUOIA, TANDEM CLX și DEC VAX-3000 folosesc duplicarea pasului de închidere cu comparații.

În timp ce codurile detectoare de erori par să rămână metoda aleasă pentru

detectarea erorilor în stocări și în server-ele hardware de comunicație, cum ar fi memoriile, disk-urile, magistralele și liniile de comunicație, ele par să ia calea duplicării și potrivirii în circuite complexe, cum ar fi CPU și *controller*-ele de dispozitive și de comunicație, bazate pe microprocesoare separate (*off-the shelf microprocessor*). Un motiv pentru această comportare este acela că duplicarea și coincidența par să ofere o aproximare mai bună a semanticii defectelor de tip cădere pentru aceste *server*-e complexe, în comparație cu codurile detectoare de erori. De aceea, în timp ce pentru CPU și *controller*-ele de intrare/ieșire bazate pe coduri detectoare de erori există posibilitatea ca datele de pe o magistrală sau de pe un dispozitiv de stocare în timpul ultimilor “câțiva” cicluri înainte de defectare să fie eronate, duplicarea și coincidența, prin folosirea circuitelor comparatoare cu autotestare, elimină virtual posibilitatea unei astfel de manifestări. Costul acestei capacități excelente de detectare a defectelor este acela că mai este nevoie de două *server*-e hardware identice, plus compararea logică, în schimbul unui singur *server* elementar pentru cazul circuitului de detectare a erorilor.

Pe lângă prevederea unei garanții mai mari a semanticii defectelor pentru *server*-e complexe, cum ar fi CPU și *controller*-ele de intrare/ieșire, duplicarea și coincidența prezintă un număr de alte caracteristici alternative. Absența circuitului de detectare a erorilor în *server*-ele fizice elementare reduce complexitatea acestora, ducând la creșterea fiabilității și la o reducere a costurilor de testare și proiectare. Eliminarea circuitului de detecție a erorilor va face, pe de altă parte, ca *server*-ele să fie mai rapide. Un alt motiv pentru utilizarea duplicării pasului de închidere este disponibilitatea unor microprocesoare rapide și ieftine, care nu au multe circuite de detectare a erorilor. Împerechind aceste componente externe și adăugând un comparator, acestea pot fi mai ieftine decât dezvoltarea proiectării proprietăților procesorului cu capacități elaborate pentru detectarea erorilor. Mai mult, împerecherea procesoarelor externe permite fabricanților de sisteme de calcul să determine o îmbunătățire rapidă în viteza cipurilor și să-și actualizeze prompt liniile de produse, odată ce vor deveni disponibile pe piață noi cipuri.

Un ultim avantaj al duplicării pasului de închidere este creșterea îmbunătățirii calității *software*-ului. Când orice defect elementar al *server*-ului hardware este detectat prompt, ca o neconcordanță înainte de a apărea orice date defecte, este ușor să se folosească o metodă care să determine datele care sunt ambigue, pentru defectele cauzate de erorile proiectării *software*, nefiind observată nici o neconcordanță între procesoarele hardware și, atunci, cu o probabilitate crescută, defectul este datorat unei erori de proiectare a *software*-ului. Aceasta este în contrast flagrant cu ce se manifestă în arhitecturile de procesoare tradiționale, bazate pe coduri corectoare de erori. Întrucât în astfel de sisteme, defectele hardware poate duce la un defect nedetectabil al datelor, datorat latenței detectării defectului, o clasă largă de defecte ale sistemului atribuite *software*-ului sunt cauzate, de fapt, de către hardware. Cunoașterea

faptului că defectele sistemului pot, cu o probabilitate semnificativă, să “mascheze” defectele de proiectare a *software*-ului, poate crea dificultăți serioase în diagnosticarea corectă a defectelor sistemului. În acest caz, proiectanții sistemului de operare pot da vina pe proasta funcționare a *hardware*-ului pentru defectele sistemului pe care nu le pot diagnostifica și, de asemenea, ei pot da vina pe sistemul de operare pentru defectele pentru care *hardware*-ul funcționează aparent corect.