



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



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

Sisteme Tolerante la Defecte

26. Redundanța temporală - Checkpoint

REDUNDAŢA TEMPORALĂ (CHECKPOINT)

Multe dintre aplicațiile curente pot să se deruleze pe durate lungi de timp. În acest sens se pot cita aplicații mari consumatoare de timp ca fiind:

- simulările unor circuite complexe (microprocesoarele) înaintea producerii măștilor (sunt raportate durate de ordinul săptămânilor)
- aplicațiile de proiectare și optimizare automată a circuitelor digitale laborioase,
- calcul financiar-bancar previzional etc.

Orice eroare semnalată și detectată, la un moment dat, poate declanșa reluarea unei astfel de aplicații cu masive implicații asupra costurilor dar chiar și asupra oportunității reluării respectivei aplicații.

Duratele lungi de calcul fac întotdeauna ca probabilitățile de apariție ale unor erori să crească, indiferent de cauzele care favorizează ori produc respectivele erori.

Un model analitic al unei astfel de situații poate fi creionat astfel:

- Rularea completă a programului durează T ore. Sistemul este afectat de defecte tranzitorii cu o medie de l erori într-o oră.
- Defectul este tranzitoriu dar tot volumul de calcul până în momentul apariției defectului este iremediabil pierdut.

STUDII DE CAZ

Se presupune că sistemul este afectat de defecte tranzitorii modelabile printr-un proces Poisson cu viteza de λ defecte într-o unitate de timp.

Se consideră că timpul de reluare a procesului de calcul al sistemului, urmare a unei defectări aleatorii, este neglijabil.

Se notează prin E timpul mediu de execuție al proceselor de calcul, incluzând orice timp al unor calcule pierdute ca urmare a apariției erorilor tranzitorii.

Există două situații:

- nu are loc nici o defectare a sistemului pe durata procesului de calcul dorit,
- există cel puțin o defectare a sistemului pe durata procesului de calcul dorit.

În primul caz durata execuției procesului de calcul dorit este T .

Probabilitatea să nu apară nici un defect într-un interval de timp cu durata T este:

$$p = e^{-\lambda T}.$$

Astfel, durata medie a timpului de execuție a procesului de calcul fără defecte este de forma:

$$T_{\text{mediu}} = T e^{-\lambda t} \quad (01)$$

Dar dacă se consideră și apariția unui defect tranzitoriu în intervalul de timp $[0, T]$ la momentul $\tau \in [0, T]$, atunci timpul mediu de execuție are expresia:

$$E = \frac{e^{\lambda T} - 1}{\lambda} \quad (02)$$

Se poate aprecia că media timpului de execuție este deosebit de sensibilă cu T . În adevăr E crește exponențial cu T .

Penalizarea indusă de procesul de apariție al unui defect tranzitoriu poate fi evaluată prin expresia: $E - T$.

Adică, timpul suplimentar impus prin pierderea unui volum de calcule la un moment dat datorită defectului tranzitoriu.

Prin normalizarea acestei penalizări față de durata nominală de execuție se obține metrica adimensională a acestei penalizări:

$$\eta = \frac{e^{\lambda T} - 1}{\lambda T} - 1 \quad (03)$$

De remarcat faptul că penalizarea η depinde numai de produsul λT .

Produsul λT reprezentând că numărul mediu de defectări care va afecta procesorul pe durata unei execuții.

Este evident că pentru valori mici ale numărului mediu de defectări penalizarea η are valori modeste dar deîndată ce produsul λT va lua valori mai mari, penalizarea η atinge valori mult mai importante.

Faptul că în urma apariției unui defect tranzitoriu se pierde tot ceea ce s-a calculat până în acel moment devine foarte supărător atunci când apariția defectului tranzitoriu are loc cât mai aproape de încheierea procesului de calcul.

O soluție este introducerea periodică a salvării, undeva pe un mediu (eventual extern) suficient de fiabil, a tuturor rezultatelor parțiale ale calculelor de până în acel moment. Astfel încât, să se poată relua calculul pornind de la acestea, dacă în viitor, înaintea unei alte salvări periodice reapare un defect tranzitoriu.

Salvarea la un moment dat a informațiilor cheie astfel încât să se poată relua calculul pornind de la acele informații, este un *checkpoint*.

Mediul de salvare al informației trebuie să fie nu numai fiabil dar să poate găzdui, eventual, un volum suficient de mare de informație cu costuri reduse. Nu sunt rare situațiile în care informația salvată periodic are volume de ordinul sutelor de megaocteți.

În aceste condiții sunt preferate discurile clasice ca mediu de salvare. Chiar și dacă se întrerupe sursa de alimentare a acestora, nu se pierde informația stocată, atâta timp cât nu este afectat mediul de stocare (suprafața, în cazul discurilor clasice).

Sunt și situații când memoria RAM, asigurată eventual și printr-o sursă suplimentară foarte stabilă de alimentare electrică (cum ar fi bateriile, spre exemplu), poate găzdui astfel de *checkpoint*-uri.

O frecventă salvare a informațiilor necesare și suficiente pentru reluare procesului de calcul poate induce o creștere a timpului de execuție, indiferent dacă apar sau nu apar defecte tranzitorii.

Această creștere a timpului de calcul, impusă de procesul de salvare definește *suprasarcina* indusă de procesul de *checkpoint*-are.

Legat de procesul de *checkpoint*-are este de luat în considerație și timpul de salvare al informațiilor. Acest timp este numit în literatură *latența checkpoint*-ării.

În cele mai multe dintre aplicațiile foarte simple *suprasarcina* și *latența checkpoint*-ării sunt aproape identice.

Dar, în cazul sistemelor mari care permit o suprapunere a operațiilor de salvare cu execuția aplicației, *latența* poate fi substanțial mai mare decât *suprasarcina*.

Latența checkpoint-ării depinde mult de volumul salvărilor.