Zotavenie zložitého systému pomocou programovej redundancie procesov

Liberios Vokorokos [1]

Complex system recovery by process programming redundancy

This paper presents the recovery of a control system resistant against faults. We come out from parallel computer system with distributed memory and communication based upon exchange of messages. This system consists of processor elements, communication lines and switches. At least one application process is running on each of the processor of parallel system. Processes are executed parallely and sequently, communicating with each other through the communication lines executing one task. Several tasks can be run on the parallel system. Processes are mapped to the processor elements.

This applied method of system endurance against fault is ensured on the level of processor elements, communication lines, switches and processes using software and hardware redundancy. The purpose of the recovery in fault tolerant parallel system is to create and insure system supporting against fault after its appearing. Resistance against faults is ensured by the applied method of a fault tolerant system.

The paper describes the function of the system after system fault. Faults in different parts of parallel system have different importance. Let's think about a fault processor, line or switch. The most important is fault on processor. In this case the processes allocated on this processor have to be moved to other processor, recovered and initialled one more time. Usually we can think about that processor memory content is lost after fault appearing, or unaccessing. It is necessary to remove and to redirect all communications lines going through this process.

The process of system recovery is known. But there is a question how and who controls recovery of kernel of processor. Control can be either centralised or decentralised. There is a question how many copies of processes are enough for sufficient resistance against faults. In case of active and passive processes it depends on requested security. One passive copy of the process is sufficient if we assume, that fault doesn't appear on two processors occupied by the same process at the same time or in time of recovery of the system.

Key words*: redundancy, recovery, process, locked process, tolerant, fault.*

Úvod

Paralelný počítačový systém sa skladá z procesorových elementov PE, komunikačných liniek a prepínačov. Procesorové elementy a prepínače sú prepojené komunikačnými linkami a tvoria systémovú topológiu alebo prepojovaciu sieť.

Na každom procesore paralelného systému beží aspoň jeden aplikačný proces. Procesy sú vykonávané paralelne a sekvenčne, komunikujú medzi sebou cez komunikačné kanály a tvoria úlohu. Na paralelnom systéme môže bežať súčasne niekoľko úloh. Procesy sú mapované do procesorových elementov. Komunikačné kanály reprezentujú výmenu údajov medzi procesmi a sú mapované do komunikačných liniek, prepínačov a procesorových elementov (Hudec et al., 1996; Vokorokos, 2000 a).

Odolnosť proti poruchám sa v takomto systéme zabezpečuje pomocou redundancie, ktorá môže byť klasifikovaná ako hardvérová, softvérová, informačná a časová. Metódy softvérovej redundancie najčastejšie využívajú existenciu niekoľkých kópií procesov. Tieto kópie môžu byť aktívne (vykonávajú sa ako bežiaci kód), alebo pasívne (sú iba odložené v pamäti ako kód). K tomu, aby pasívne procesy mohli byť zotavené od posledného aktuálneho stavu hlavného procesu, je potrebné, aby mali k dispozícii aktuálny záznam, informáciu o stave hlavného procesu. Tieto počítačové konfigurácie sa výhodne môžu aplikovať pre riadenie systémov s vysokou náročnosťou na bezpečnosť a spoľahlivosť prevádzky, ako sú napríklad procesy kontinuálnej výroby (vysoké pece), systémy zvislej banskej dopravy a podobne.

Vzniká otázka, koľko kópií procesov je postačujúcich na zabezpečenie odolnosti proti poruchám. V prípade aktívnych aj pasívnych procesov je to otázka vyžadovaného zabezpečenia. Postačujúca je napr. jedna pasívna kópia procesu, ak predpokladáme, že nevznikne porucha na dvoch procesoroch obsadených tým istým procesom súčasne, alebo v intervale v ktorom nie je ukončené zotavenie z poruchy.

Jadro systému pre odolnosť proti poruchám

Úlohou systémovej diagnostiky je detekovať a lokalizovať vznik poruchy a informovať o nej ostatné časti operačného systému. Aplikovaná metóda systémovej odolnosti proti poruchám zabezpečuje odolnosť proti poruchám na úrovni procesorových elementov, komunikačných liniek, prepínačov a procesov formou softvérovej a hardvérovej redundancie. Systémové zotavenie z poruchy realizuje zotavenie procesov, ktoré boli ovplyvnené poruchou alebo tie procesy, ktoré bežali na poruchovom procesore (Vokorokos, 2000 c).

[1] *Ing. Liberios Vokorokos*, Department of Computers and Informatics Faculty of Electrical Engineering and Informatics, Letná 9, 042 00 Košice, Slovakia, vokoroko@tuke.sk
(Doručené 20.10.2000, revidovaná verzia dodaná 25.1.2001)

Časťou operačného systému paralelného počítačového systému odolného proti poruchám je jadro pre odolnosť proti poruchám (Janík, 1996). Je to množina procesov vysokej priority, distribuovaných na každom procesore. Zabezpečuje úlohy a funkcie zotavovania po chybách v systéme.

Úlohami jadra je vykonávať niektoré funkcie počas zotavovania:

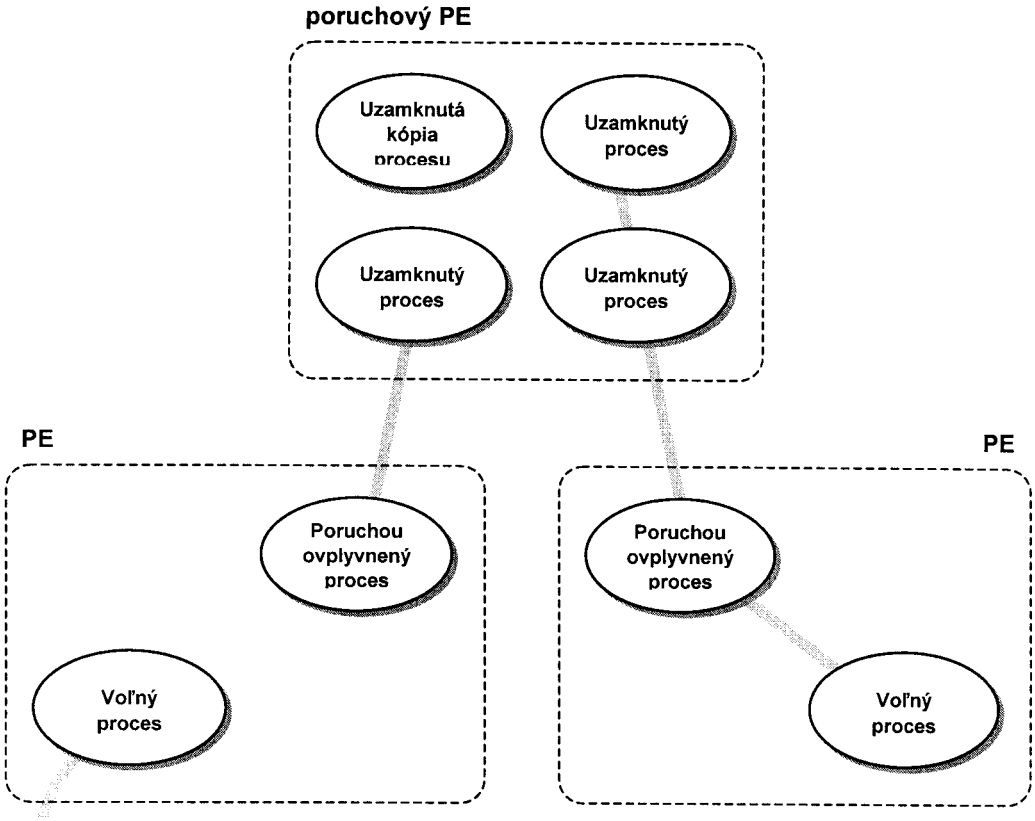
- zastaviť ľubovoľný proces,
- zaznamenať stav procesu,
- určiť vhodnú alokáciu procesov, ktoré treba premiestniť v dôsledku vzniknutej poruchy,
- presunúť ľubovoľný proces aj s jeho údajmi z jedného procesorového elementu na niektorý iný,
- naštartovať, naplánovať, zrušiť plán a zmeniť plán ľubovoľného procesu vo vhodnom stave,
- zastaviť, naštartovať, naplánovať a preplánovať komunikáciu vybraných procesov,
- zabezpečovať správu a aktualizovať údaje o parametroch procesov a procesorov, potrebné pre rozhodovanie o alokácii procesov počas zotavovania systému.

Proces zotavovania systému

V predchádzajúcej časti popísané vlastnosti jadra pre odolnosť voči poruchám determinujú jeho činnosť po vzniku poruchy.

Poruchy rôznych častí paralelného systému majú rôznu závažnosť. Uvažujme napr. poruchu procesora, linky alebo prepínača.

Najviac závažná je porucha procesora. V tomto prípade procesy alokované na ňom musia byť premiestnené na iné procesory a tam zotavené a inicializované. No spravidla treba počítať s tým, že obsah pamäti procesora je po vzniku poruchy navždy stratený, resp. nedostupný, že je potrebné premiestniť a presmerovať všetky komunikačné kanály, ktoré prechádzali týmto procesorom (Vokorokos, 2000 b).



Obr.1. Vlastnosti procesov po vzniku poruchy procesorového elementu.
Fig.1. Properties of processes after fault appears in processor element.

Porucha procesorového elementu má za následok stratu údajov, aktuálneho stavu procesov, ich kódu a navyše degradáciu výkonnosti systému, ak nie je k dispozícii rezervný procesor (Kollár, 1995).

Každý proces paralelného systému od okamihu vzniku poruchy až po koniec zotavenia systému dostáva nový atribút (obr. 1).

Ak sa procesorový element PE stal poruchovým, potom:

- každý proces alokovaný na procesorovom elemente PE sa nazýva uzamknutý proces (hlavný aj kópia),
- každý proces, s výnimkou uzamknutých, komunikujúci s uzamknutým procesom, sa nazýva poruchou ovplyvnený proces,
- každý proces, s výnimkou uzamknutých, nekomunikujúci s uzamknutým procesom, sa nazýva voľný proces.

Ak sa dostala do poruchy linka alebo prepínač, potom:

- proces, ktorý práve uskutočňoval komunikáciu cez linku, ktorá sa dostala do poruchy, sa nazýva poruchou ovplyvnený proces,
- každý iný proces sa nazýva voľný proces.

Proces komunikujúci s uzamknutým procesom je proces, ktorý práve komunikuje alebo čaká na komunikáciu alebo bude komunikovať v budúcnosti.

Komunikácii dvoch procesov možno priradiť jeden z dvoch atribútov. Komunikácia dvoch procesov, alokovaných na tom istom procesore, sa nazýva vnútroprocesorová komunikácia. Komunikácia dvoch procesov, ktoré nie sú alokované na tom istom procesore, sa nazýva medziprocesorová komunikácia. Vnútroprocesorová komunikácia je rýchla a zaberá minimálny čas.

Tab.1. Proces zotavovania. Tab.1. Recovery process.

<table>
<tr><th colspan="2">Porucha procesorového elementu:</th><th>Porucha komunikačnej linky alebo prepínača:</th></tr>
<tr><td>1</td><td colspan="2">Priebežná alebo periodická systémová diagnostika detekuje poruchu niektorého komponentu a vysiela správu do jadra systému.</td></tr>
<tr><td>2</td><td colspan="2">Systémová diagnostika vysiela správu o lokácii poruchy.</td></tr>
<tr><td>3</td><td colspan="2">Jadro systému analyzuje poruchový stav.</td></tr>
<tr><td>4</td><td>Jadro systému zmrazí všetky uzamknuté procesy (ak je to možné) a komunikáciu poruchou ovplyvnených procesov s uzamknutými. Ak je to možné, tak každá dôležitá informácia, uložená na poruchovom procesore, musí byť presunutá na niektorý iný vhodný procesorový element.</td><td>Jadro systému zmrazí komunikáciu všetkých poruchou ovplyvnených procesov a zabráni ďalšiemu aktivovaniu komunikácie na danej linke (linkách)
Zmrazenie znamená zastavenie bežiaceho procesu a jeho komunikácie a odpamätanie jeho stavu.</td></tr>
<tr><td>5</td><td>Jadro systému vyberá cieľové procesory a procesory pre nové kópie uzamknutých procesov.</td><td rowspan="3">Jadro systému informuje ostatné časti operačného systému, hlavne systém smerovania správ.</td></tr>
<tr><td>6</td><td>Jadro systému vytvára nové kópie, presúva kód.</td></tr>
<tr><td>7</td><td>Jadro systému zotavuje, reštartuje a preplánováva uzamknuté procesy, o zmenách informuje ostatné časti operačného systému.</td></tr>
<tr><td>8</td><td colspan="2">Jadro systému rozmrazuje poruchou ovplyvnené procesy.</td></tr>
</table>

V procese zotavovania systému po vzniku poruchy niektorého procesora, určuje jadro systému pre každý uzamknutý proces procesor, kde bude tento umiestnený, zotavený a reštartovaný. Tento procesor sa nazýva cieľový procesor PE.

V tabuľke 1 je znázornený proces zotavovania systému pri poruche procesora, linky a prepínača. V kroku 5 v prípade poruchy procesora sa vyberá cieľový procesor. Postupnosť krokov v procese zotavovania určuje možnú stratégiu.

Nasledujúce tri stratégie sa líšia poradím, časom a spôsobom vykonávania krokov 5 a 6.

Následná stratégia zotavovania systému. Táto stratégia je najjednoduchšia. Určenie cieľových procesorov pre uzamknuté procesy sa určuje v čase zotavovania, po vzniku poruchy. Je vhodná pre aplikácie, v ktorých sa mení záťaž procesorov a parametre procesov a premiestňovanie procesov nie je vhodné.

Statická dopredná stratégia redukuje čas potrebný na určenie cieľových procesorov a premiestnenie na ne kópií uzamknutých procesov, čím urýchľuje zotavovanie systému. Určenie cieľových procesorov sa uskutočňuje pred výskytom poruchy a kópie procesov sú umiestnené na procesoroch, kde v prípade poruchy procesora budú aj zotavené a naštartované. Je vhodná pre systémy, v ktorých sa záťaž procesorov a parametre procesov nemenia a sú známe dopredu.

Dynamická dopredná stratégia dopĺňa predchádzajúcu v tom, že umiestnenie kópií procesov sa mení podľa zmien v systéme. Kópie sú umiestňované na tie procesory, ktoré by sa stali cieľovými v prípade vzniku poruchy

niektorého procesora, podľa aktuálneho stavu systému. Je vhodná pre systémy, v ktorých sa mení záťaž procesorov a v ktorých je implementovaný vyrovnávač záťaže.

Riadenie zotavenia procesu

Proces zotavovania systému je známy. No vzniká otázka, ako a kto riadi zotavovanie, jadro ktorého procesora. Riadenie môže byť:

- centralizované,
- decentralizované.

V centralizovanom riadení je jeden procesor určený ako riadiaci uzol. Určí ho buď systémová diagnostika alebo sa môže určiť podľa princípu prvý informovaný - ktoré jadro systému je skôr informované o vzniku poruchy. Vtedy proces zotavovania rýchlo začne, no je potrebné zabezpečiť informovanosť ostatných jadier o tom, že riadiaci uzol je už určený.

Pri decentralizovanom riadení treba vychádzať z predpokladu, že všetky jadrá majú k dispozícii rovnaké údaje, podľa ktorých určujú cieľové procesory, takže je jedno, ktorý z nich ich určí. Každé jadro určí cieľové procesory pre tie uzamknuté procesy, ktoré majú na jeho procesore alokované kópie procesov. Ak sa kópia procesu nachádza na viacerých procesoroch, tak príslušné procesory vysielajú správu zotavovania systému ostatným procesorom, kde sú kópie ešte umiestnené, o tom, že mali určovať cieľové procesory pre dané kópie procesov, spolu s časovou značkou začiatku zotavovania. Jadro systému po prijatí správ zotavovania systému porovnáva túto časovú značku so svojim začiatkom zotavovania. Neskôr jadro nerealizuje realokáciu kódu príslušných procesov. V prípade rovnosti časových značiek môže rozhodnúť iné kritérium, napríklad identifikačné číslo procesora. Tento mechanizmus si vyžaduje dokonalú synchronizáciu logických systémových hodín.

Určenie procesorov, na ktorých budú umiestnené nové kópie uzamknutých procesov záleží na použitej stratégii. V následnej stratégii nezáleží, kde budú nové kópie umiestnené. V statickej doprednej stratégii sa umiestnenie nových kópií procesov určí podľa parametrov procesov a procesorov tak isto, ako pri inicializácii systému. Pri dynamickej doprednej stratégii nezáleží na umiestnení kópií procesov, samotný jej mechanizmus zabezpečí ich najvhodnejšie umiestnenie.

Záver

V rámci rôznych smerov vývoja počítačov novej generácie s extrémne vysokou výkonnosťou sa v súčasnosti venuje pozornosť osobitnej triede paralelných počítačových systémov.

Cieľom článku bolo nájsť metódu pre určenie optimálneho zotavenia systému, s ohľadom na minimálnu degradáciu výkonnosti procesov po zotavení. Model pozostáva z procesorov, ktoré môžu byť umiestnené do rôznych častí systému, čím sa mení ich vzájomná vzdialenosť. Na procesoroch sú alokované procesy, ktoré tvoria úlohu a komunikujú medzi sebou.

Na procesoroch sú takisto alokované kópie procesov. V istom časovom okamihu sa jeden z procesorov dostane do poruchy. Systémová diagnostika detekuje tento stav a začne sa zotavovanie systému. Uzamknuté procesy sú zotavené z kópií procesov a úloha pokračuje ďalej.

Literatúra

HUDEC, L. and LESKO, J.: Parallel Computing Recovery by Rollback Point Insertion. *In: Proc. Of Scientific Conference with Intern. Participation. Electronic Computers and Informatics*. Košice-Herlany 26-27.9.96, pp. 2-12.

JANÍK, P.: Optimalizácia rekonfigurácie viacprocesorových systémov odolných proti poruchám. *Dizertačná práca.* Bratislava 1996.

KOLLÁR, J.: Funkcionálne programovanie. *ELFA*, Košice, 1995.

VOKOROKOS, L: Decentralizovaná diagnostika rotorov pomocou skupiny pozorovateľov. *Mezinárodní vědecká konference. Výrobní systémy s prúmyslovými roboty*, Ostrava, September, 5-7, 2000 a, pp. 64.1-6.

VOKOROKOS, L: Faults diagnosis of control system using the observer. *4th IEEE Intern. Conference on Intelligent Engineering Systems 2000*, Portorož, Slovenia, September, 17-19, 2000 b, pp. 189-192.

VOKOROKOS, L: Data Flow model design for recovery of parallel systems. *4th intern. scientific conference of electronic computers and informatics*, Košice - Herľany, September, 28-29, 2000 c, pp. 233-236.