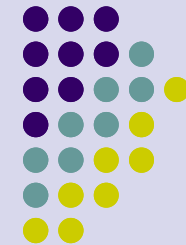
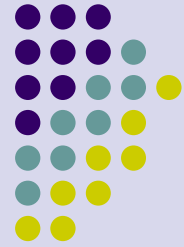


Distribuovaný deadlock - přehled



Přednášky z Distribuovaných systémů
Ing. Jiří Ledvina, CSc.



Distribuovaný deadlock

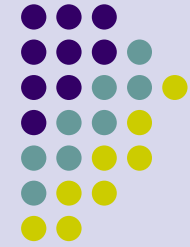
- Soubor procesů, blokováných čekáním na podmínky, které nemohou nastat
- Ilustrace grafem WFG – Wait For Graph
 - uzly jsou procesy v systému
 - orientované hrany – zachycují vztahy blokování
- Starvation – procesu je bráněno v pokračování (vyhladovění)
 - není deadlock



Podmínky vzniku deadlocku

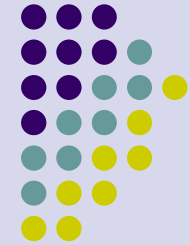
- stálé zdroje
 - vzájemné vyloučení
 - postupné předělování
 - nepreemptivní plánování
 - neomezené čekání
- dočasné zdroje
 - nehierarchické volání
 - neomezená doba odezvy

Zpracování deadlocku – stálé zdroje



- prevence (prevention) – principiálně nepřipustím deadlock
- zamezení (avoidance) – vyhnu se deadlocku včasným testováním
- detekce a odstranění (detection and resolution)
 - centralizovaný algoritmus
 - decentralizovaný algoritmus
 - hierarchický algoritmus

Zobrazení situace vedoucí k deadlocku



- Graf přidělených zdrojů a čekajících procesů
- resource allocation graph (RAG)
 - dva typy uzlů
 - procesy (kroužky)
 - zdroje (čteverečky)
 - dva typy hran
 - hrana požadavků
 - hrana přidělení



Prevence deadlocku

- eliminace jedné ze čtyř nutných podmínek pro vznik deadlocku
- nemusí být efektivní
- např. hierarchické přidělování zdrojů
 - každý zdroj má přiděleno ohodnocení
 - o zdroje se žádá vzestupně
 - uvolňují se sestupně
 - při znovu přidělení zdroje je třeba všechny zdroje ze skupiny uvolnit a pak znovu přidělit



Zamezení deadlocku

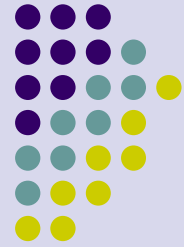
- uspořádání požadavků, nemusí být efektivní
- všechny požadavky se musí znát dopředu
- zdroje musí být předem dány, požadavky na ně také
- pro každý požadavek se musí provést analýza stavu
- např. bankéřův algoritmus
 - je dána jistina bankéře (kolik může max. půjčit)
 - jsou dány max. požadavky klientů
 - podmínky úlohy – jinak nemá ceny řešit
 - žádný z požadavků nesmí být větší než jistina
 - součet požadavků musí být větší než jistina
 - bankéř je ve stavu jistý, testuje vznik stavu nejistý
 - při požadavku na další zdroje "dopočítává" vývoj půjčování pro nejhorší případ



Detekce a odstranění deadlocku

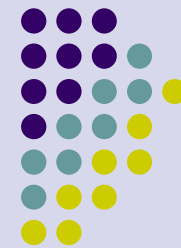
- detekce deadlocku
 - udržování grafu dostupnosti zdrojů, hledání deadlocku
 - požadavky na algoritmus
 - schopnost detekovat jakýkoliv deadlock
 - nedetekovat zdánlivé deadlocky (phantom deadlock)
- odstranění deadlocku
 - přerušování zacyklení grafu
 - přerušování jednoho nebo více procesů a vrácení jejich zdrojů
 - procesy se mohou vracet k synchronizačnímu bodu

Detekce distribuovaného deadlocku



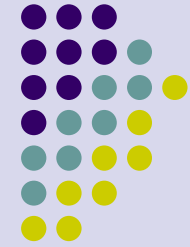
- Centralizovaný algoritmus
 - koordinátor určuje globální WFG a hledá cykly
 - je jednoduchý
 - Ho a Ramamoorthy, jedno a dvoufázový algoritmus
- distribuovaný algoritmus
 - globální WFG se schopností detekce rozšířené na více uzlů
 - Obermark algoritmus path-pushing (strkáním cesty)
 - Candy, Mistra, Haas – edge-chasing (vytepávání hran)
 - difuzní
- hierarchický algoritmus – hierarchická organizace, strany detekují deadlock pouze zahrnutím svých potomků
 - Menasce a Muntz algoritmus
 - Ho a Ramamoorthy algoritmus

Jednoduchá a centralizovaná detekce deadlocku



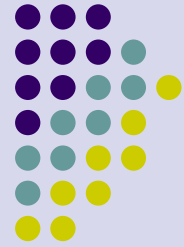
- centrální koordinátor určuje WFG systému
- všichni přidělují a uvolňují zdroje posíláním zpráv koordinátorovi
- koordinátor udržuje graf a detekuje deadlock
- problémy
 - zatížení komunikačních linek
 - úzké místo systému
 - může také detekovat "zdánlivý deadlock"
 - P1 čeká na P2
 - P3 čeká na P1
 - zpráva 1: P1 uvolní zdroj, na který čeká P3
 - zpráva 2: P2 požaduje zdroj, který drží P3
 - pokud se zpráva 1 a 2 při příjmu zamění, nastane zdánlivý deadlock

Ho a Ramamoorthy dvoufázový centralizovaný algoritmus detekce deadlocku



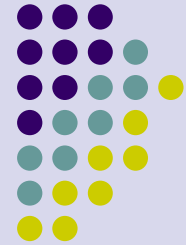
- každá strana si udržuje 2 tabulky
 - tabulku přidělených zdrojů (zdroj přidělen procesu)
 - tabulku čekajících procesů (proces čeká na zdroj)
- koordinátor periodicky požaduje zaslání všech stavových tabulek, vytváří WFG a testuje deadlock
 - nejsou-li cykly, není deadlock
 - najde-li cyklus, vyžádá si opět tabulky a znovu buduje WFG
 - používá ale pouze ty hrany, společné oběma souborům stavových tabulek
- důvodem je to, že pokud použije informaci ze dvou po sobě jdoucích záznamů, získá koordinátor konzistentní pohled na stav.
 - algoritmus redukuje možnost indikace falešných deadlocků, ale úplně je neodstraňuje
- nalezení cyklu v tabulce neznamená existenci deadlocku

Ho a Ramamoorthy jednofázový centralizovaný algoritmus detekce deadlocku



- každá strana si udržuje dvě tabulky
 - všechny lokální procesy a zdroje, které jsou uzamčené
 - zdroje uzamčené v této straně od lokálních i nelokálních procesů
- strana periodicky požaduje obě tabulky, konstruuje graf WFG
- WFG obsahuje pouze informaci o nelokálních procesech, jestliže tato informace je shodná pro procesy strany i zdroje strany
 - detekuje-li se cyklus, jde o deadlock,
 - nedetekuje-li se, nejde o deadlock
- detekuje korektně deadlocky eliminujících nekonzistentnost podávání zpráv vlivem zpoždění
- vyžaduje více prostoru než 2-fázové HR

Distribuovaná a hierarchická detekce a odstranění deadlocku



- detekce
 - distribuovaný algoritmus
 - Obermarckův path-pushing algoritmus
 - Chandy, Misra a Haas edge-chasing
 - hierarchický algoritmus
 - Menasce a Muntz algoritmus
 - Ho a Ramamoorthy algoritmus
 - odstranění

Distribuovaná detekce deadlocku



- path-pushing
 - WFG je rozdělen na cesty – posloupnosti hran
 - deadlock nastane, když proces detekuje lokální cyklus
- edge-pushing
 - cirkulují zkušební zprávy
 - blokový proces posílá zkušební zprávu procesům držícím požadované zdroje
 - deadlock nastává pokud iniciátor obdrží vlastní zprávu

Distribuovaná detekce deadlocku



- Obermark path-pushing
- Chandy, mitra a Haas Edge-chasing
- Menasce a Muntz Hierarchická detekce deadlocku
- Ho a Ramamoorthy hierarchická detekce deadlocku
- Odstranění deadlocku