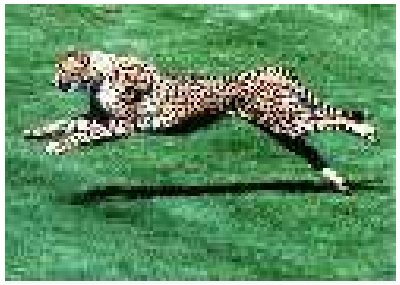
IBM – Informix Dynamic Server



Podrobně o výkonnosti v IDS 11

Jan Musil
IT Specialist SWG IBM

Přehled prezentace

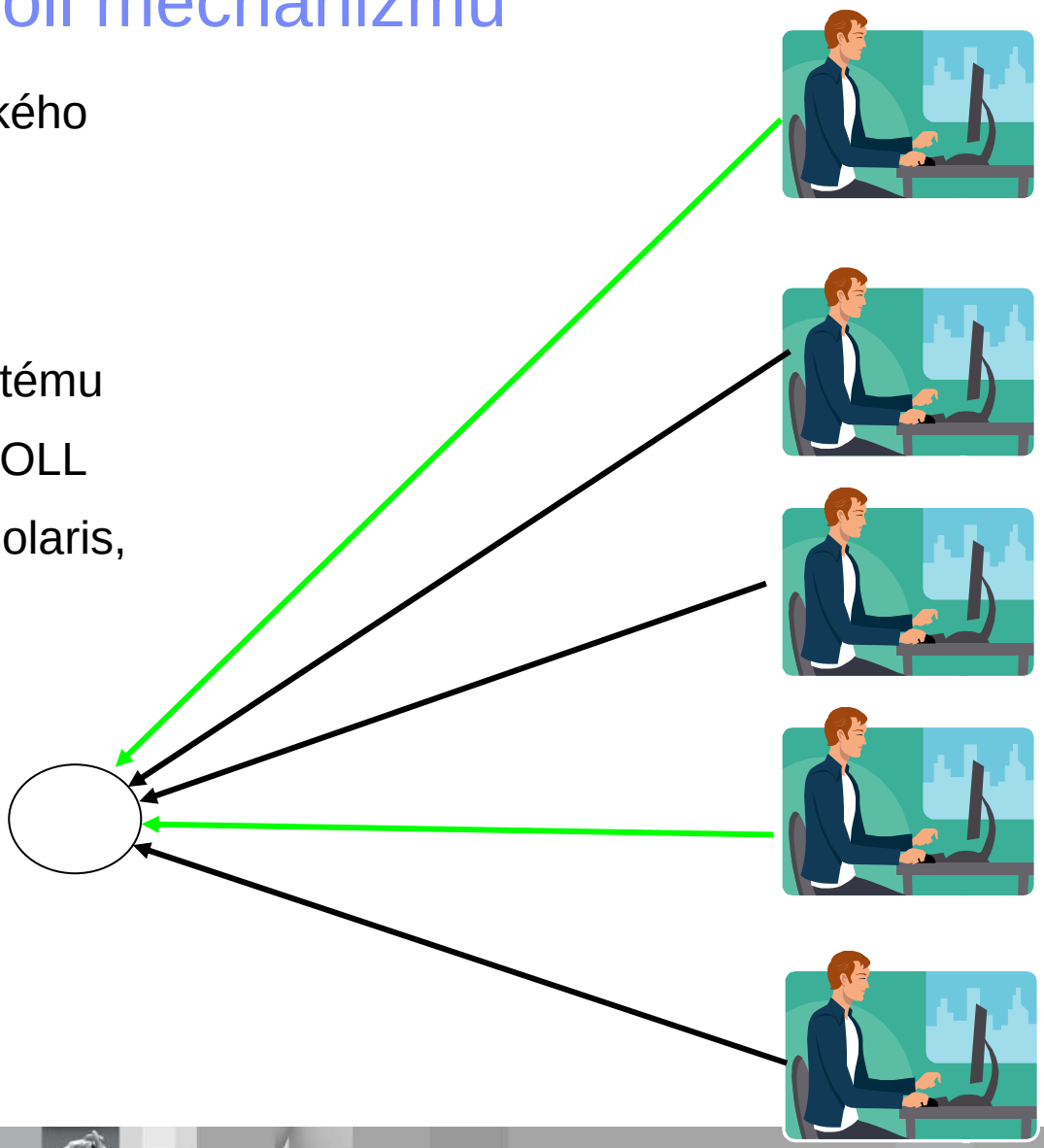
- Implementace Fast Poll
- Vyhrazená paměť pro CPU VP
- Implementace Direct I/O
- Optimalizace PREPARE fáze SQL příkazů
- Lineární automatické čištění indexů
- Kontrolní body a RTO

Implementace Fast Poll

Implementace Fast Poll mechanizmu

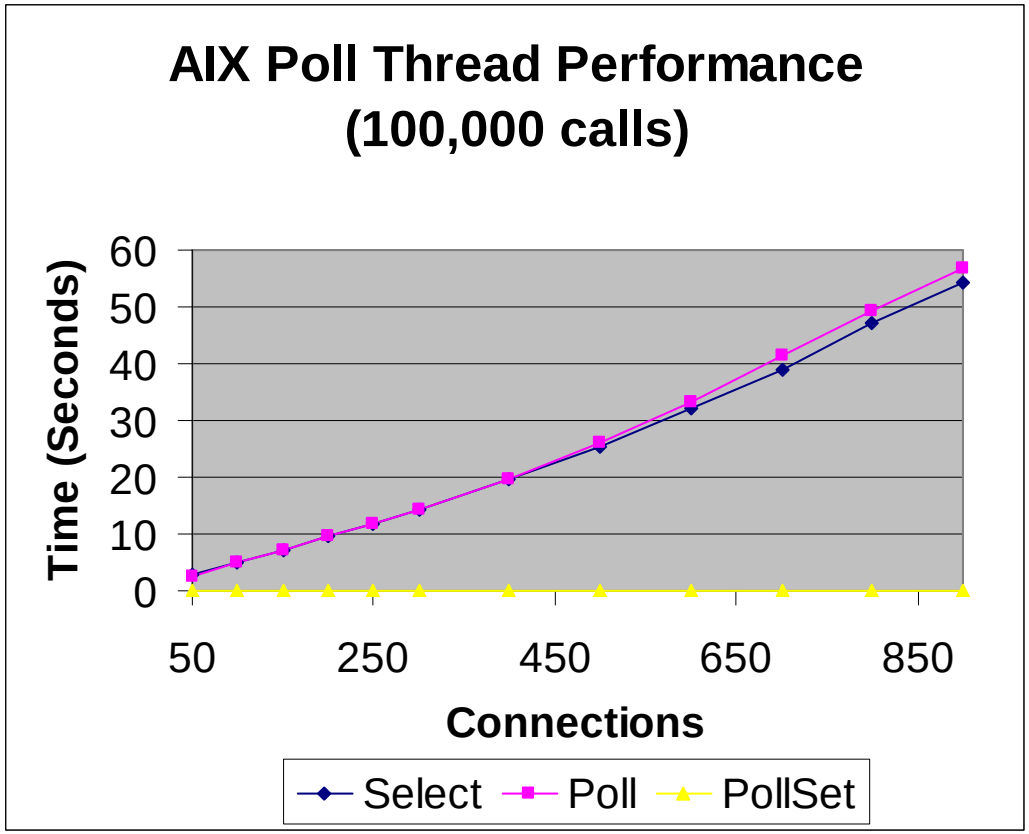
- Optimalizace monitorování velkého množství připojení
- Na straně IDS zajištěno prostřednictvím poll vlákna
- Nutná podpora operačního systému
- Konfigurační parametr: FASTPOLL
- Podporované platformy: AIX, Solaris, HP/UX, Linux

Poll List
- Line 1
- Line 4



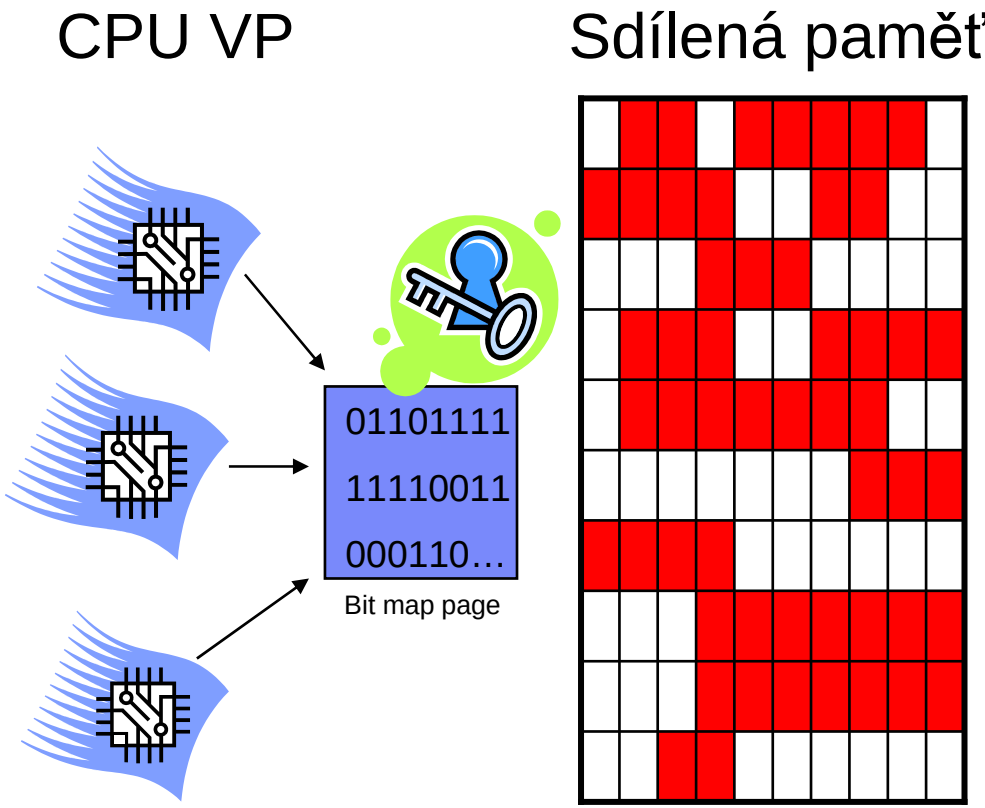
Porovnání „klasického“ přístupu a FastPoll u AIXu

- Použití klasického přístupu (select() a poll()) vede k exponenciální době odezvy při monitorování požadavku na připojení
- PollSet – AIX 5.3 implementace Fast Poll mechanizmu
- Použitím PollSet volání je odezva lineární

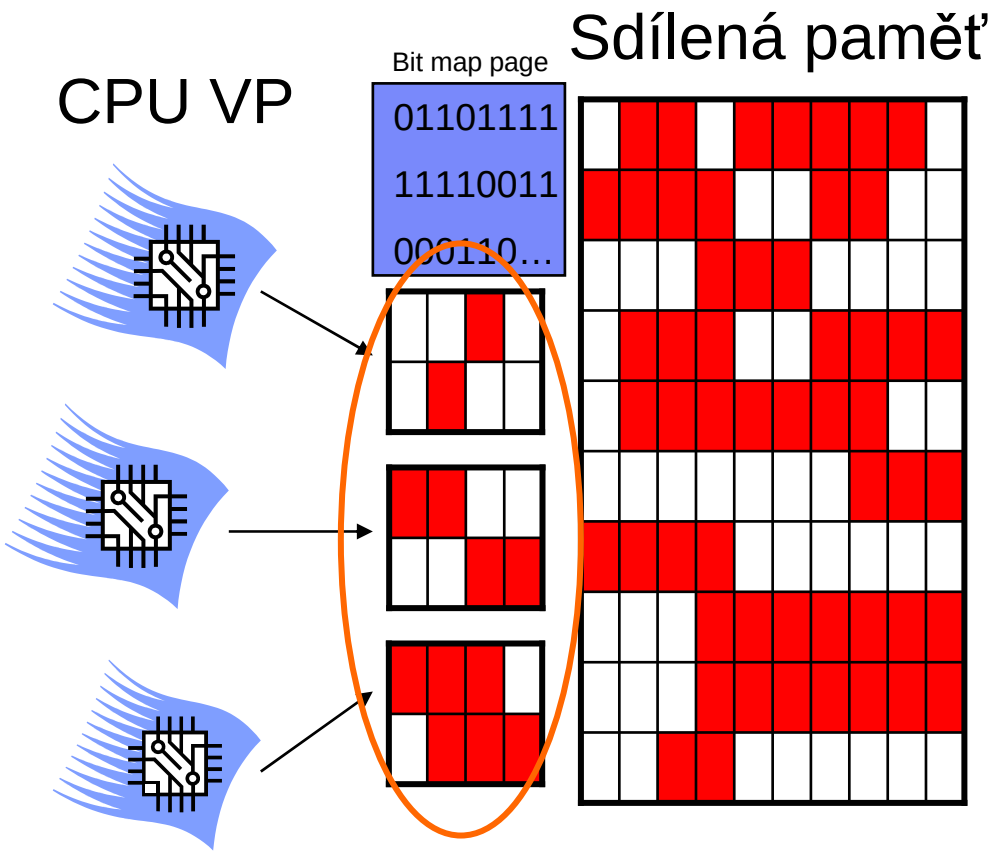


Vyhrazená paměť pro CPU VP

Přiřazení vyhrazené paměti pro CPU VP



Přiřazení vyhrazené paměti pro CPU VP



- VP_MEMORY_CACHE_KB
- onmode –wm|-wf
- Minimum pro každý CPU VP je 800 kB
- Nesmí být alokováno více než 40% ze SHMTOTAL
- onstat –g vpcache

Implementace Direct I/O

Problém s *cooked* zařízeními

- Výkonnost IO operací výrazně nižší než u *raw* zařízení
- Ztráta informací z vyrovnávací paměti OS v případě pádu systému
- Nemožnost použití Kernel AIO
- I/O operace lze provádět pouze prostřednictvím IDS implementace AIO

DIRECT I/O

- Vlastnost operačních systémů, která umožňuje provádět zápisy do souborů přímo bez použití vyrovnávací paměti souborového systému
- Jedná se o funkcionalitu implementovanou s ohledem na databázové systémy
- Implementovaná pro Linux, AIX a Solaris
- Umožňuje používat Kernel AIO

Co je to Direct I/O

- Direct I/O (DIO) je vlastnost operačního systému, která zajišťuje, že aplikace provádí čtení a zápisy přímo na diskové zařízení bez použití vyrovnávací paměti OS pro čtení a zápisy
- Výhodou DIO je snížení CPU operací a omezení režie s dvojnásobným kopírováním dat (nejprve z disku do vyrovnávací paměti OS a pak z vyrovnávací paměti OS do paměti aplikace)
- DIO je možné použít takovými aplikacemi, které spravují svojí vlastní vyrovnávací paměť, například databázové systémy
- Soubor musí být otevřen s příznakem O_DIRECT

Porovnání I/O přístupů používaných IDS

- LINUX:
  - *raw* devices jsou stále nejlepší
  - Výkonnost direct I/O se přibližuje *raw* zařízením
  - Použití kaio může být nepatrně pomalejší (jak pro *raw* zařízení, tak pro soubory)
- AIX: nebyl pozorován žádný rozdíl pro zápisy (ale nižší cpu využití při použití direct I/O).
- Solaris
  - U souborů čtyřnásobné zlepšení IO při použití Direct IO
  - Výkonnost direct I/O do souboru a raw zařízení stejná

I/O metoda	Podporované platformy
KAIO	Většina platforem – kromě Z-Series Linux, Unixware, Tandem, DG
Asynchronous I/O	Všechny platformy
Direct I/O	AIX jfs2, RHAT4 ext2/ext3, RHAT5 gpfs, Solaris ufs

Konfigurace DIO

- Konfigurační parametr: DIRECT_IO
  - 0 = nepoužívat direct I/O (default)
  - 1 = použít direct I/O pokud je podporované (default pro Linux)
- Není podporované pro dočasné databázové prostory

Optimalizace PREPARE fáze SQL příkazů

Problém s PREPARE fází

- Při provádění PREPARE fáze, se ze systémového katalogu sysuser zjišťuje, zda je uživatel DBA nebo nikoliv
- sysuser systémový katalog se prochází sekvenčně
- Při velkém počtu uživatelů tento proces zpomaluje PREPARE fázi

Nová DBA vyrovnávací paměť

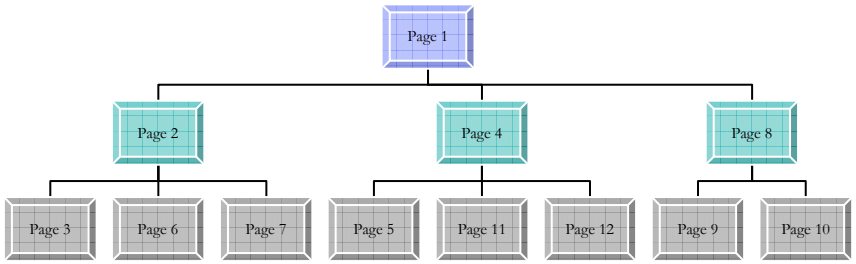
- Nová vyrovnávací paměť vytvořená na úrovni IDS instance
- Obsahuje seznam všech databází a uživatelů s DBA právy pro každou databázi
- Při PREPARE fázi se prohledává tato DBA vyrovnávací paměť
- Systémový katalog sysuser se již sekvenčně neprohledává, PREPARE fáze je tím rychlejší
- Pokud je nějakému uživateli přiděleno resp. odejmuto DBA právo, nový záznam je vytvořen resp. vymazán z DBA vyrovnávací paměti
- Totéž platí při vytvoření resp. smazání databáze

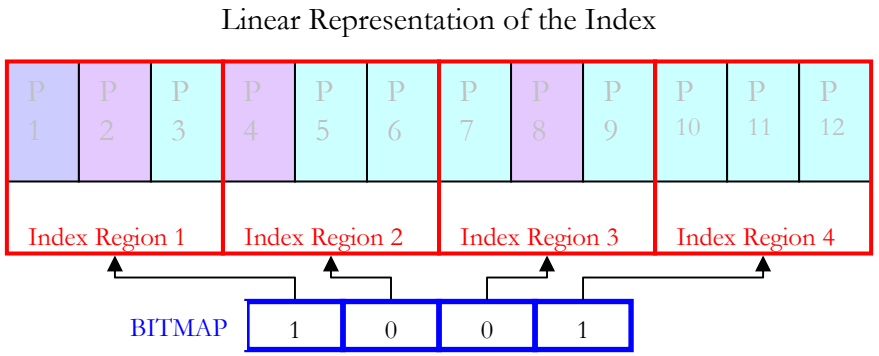
Lineární automatické čištění indexů

ALICE (Autonomic Linear Index Cleaning)

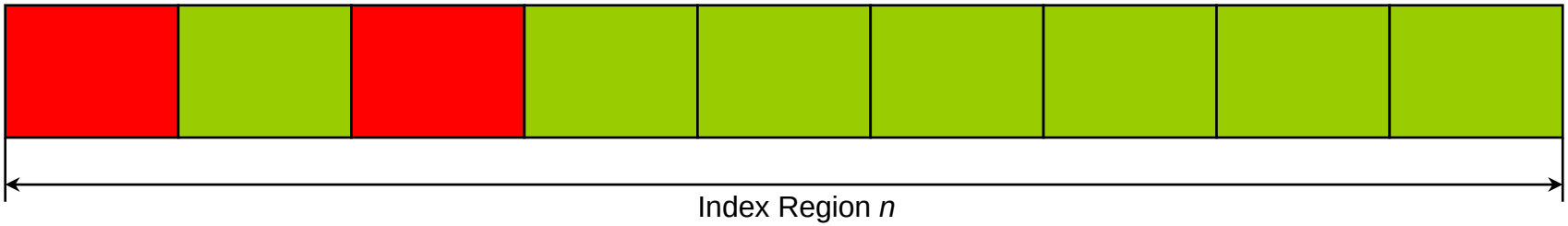
- Optimalizuje ladění b-tree scannerů na systémové úrovni
- Automaticky určuje neefektivní indexy
  - Alokuje další zdroje (paměť) pro indexy náročné na IO
  - Redukuje množství diskových IO operací
- Měřítkem efektivity je “I/O miss” poměr
  - „I/O Miss“ poměr = počet I/O čtení porovnaný s počtem I/O čtení, která nenalezla žádnou stránku se záznamem ukončené transakce určeným pro vymazání (commited deleted item).

ALICE – linearizace indexových stránek



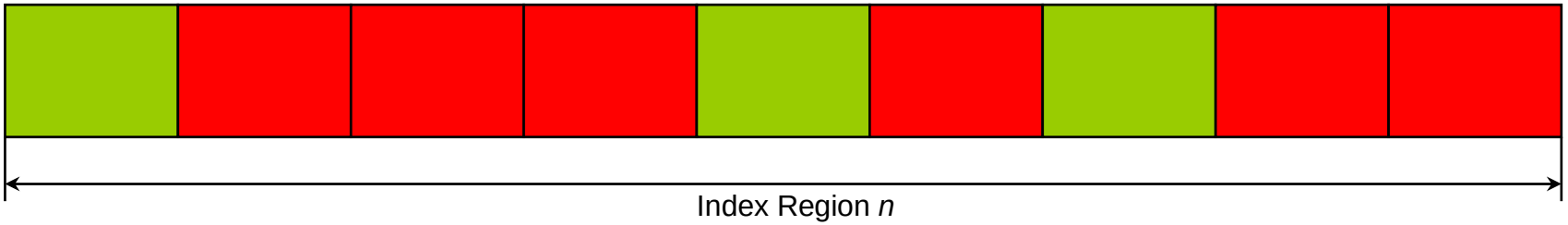


Dynamické nastavení indexového regionu



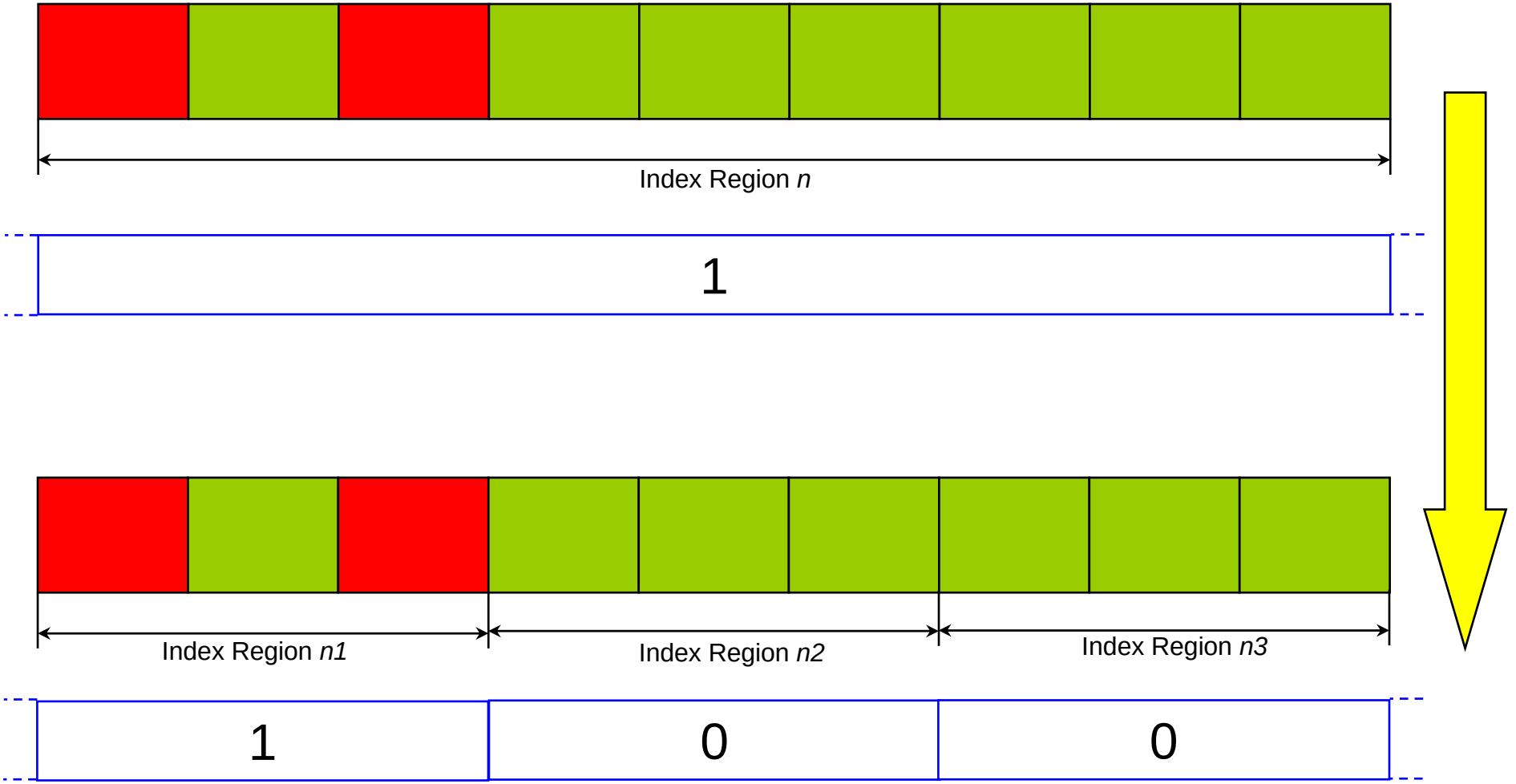
- Region s vysokým „IO Miss“ poměrem

- Zvětšení počtu regionů
- Zmenšení velikost regionu



- Region s nízkým „IO Miss“ poměrem

Dynamická rekonfigurace regionů



Konfigurace

- Konfigurační parametr: BTSCANNER ….. alice=#
  - 0 , pokud je funkcionalita ALICE vypnuta
  - default hodnota je 6, maximum je 12.
- Číslo udává množství alokovaných zdrojů
- Malé a střední systémy s žádným nebo malým počtem indexů na 1 GB
  - alice=6-7
- Pro velké systémy
  - alice>7
- onmode –C alice #
- onstat –C alice

Kontrolní body a RTO

Problémy se současnou implementací kontrolních bodů

- Ladění délky kontrolních bodů je v rozporu s laděním OLTP nastavení
  - Zkrácení MLRU (agresivní ladění LRU) vede k neustálému vyprazdňování modifikovaných bufferů
  - Zmenšení vyrovnávací paměti pro zápisy
  - Velmi náročné na CPU
  - Vyšší míra „soupeření“ o buffery
- Obtížné dosáhnout optimálního naladění
- Mnoho zákazníků vyžadovalo stanovit dobu trvání a frekvenci kontrolních bodů podle požadavku na dobu zotavení systému
  - Nyní je doba trvání stanovena pevně a vyvolání kontrolního bodu je určeno dosažením určitého omezení
- Pro splnění tohoto požadavku má IDS nedostatek informací

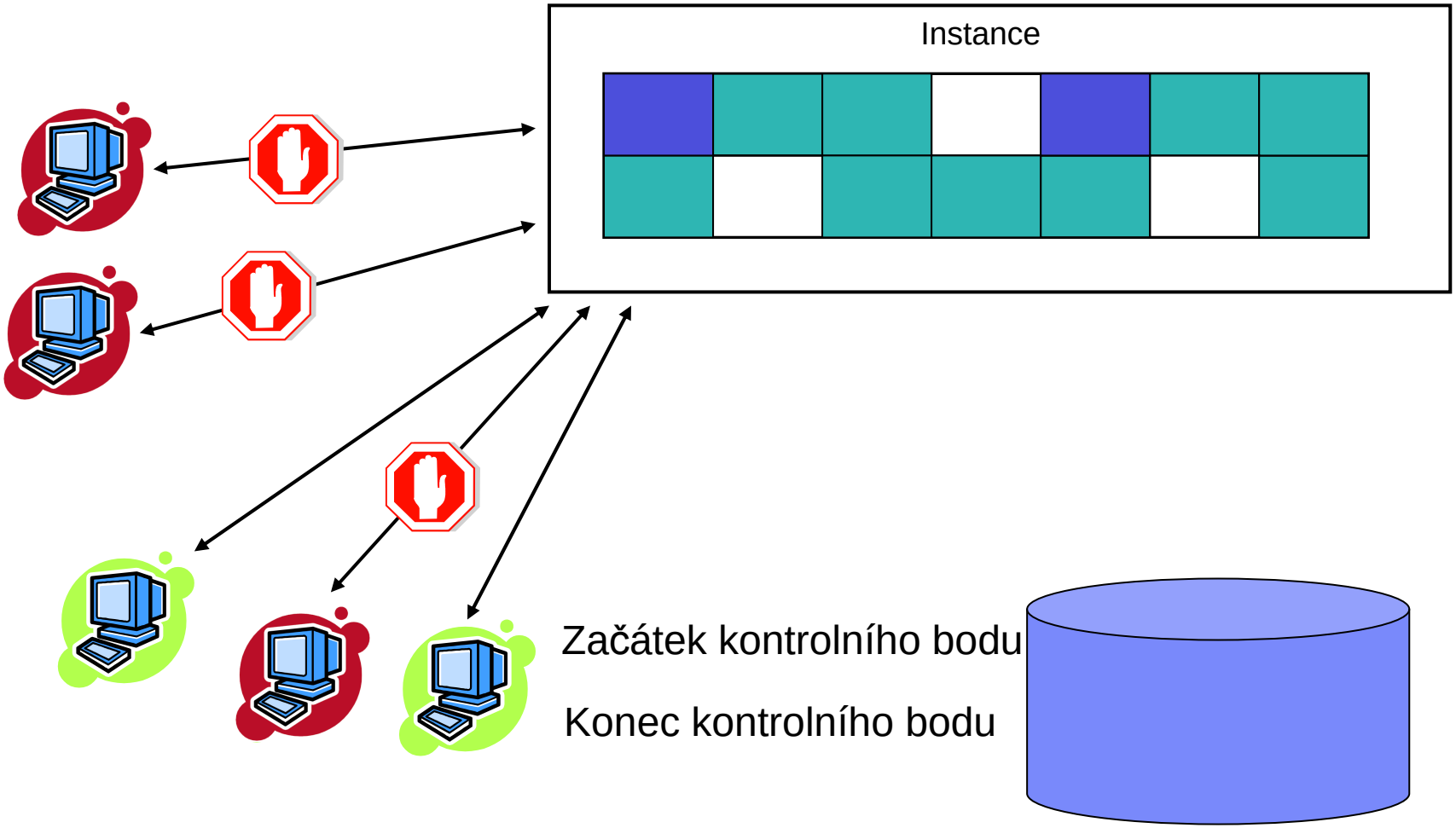
Současná implementace kontrolních bodů způsobuje

- Blokování transakcí (i v případě fuzzy kontrolních bodů)
- Fuzzy kontrolní body jsou nepředvídatelné z pohledu doby, kdy dojde k blokování transakcí
- Fuzzy kontrolní body pracují pouze s datovými stránkami, nikoliv s indexy
- S fuzzy kontrolními body nelze předpovídat, jak dlouho bude trvat zotavení systému po chybě
- Konflikt mezi dobou blokování a dobou zotavení systému
  - Krátký interval mezi KB (rychlejší zotavení) → větší blokování transakcí
  - Dlouhý interval mezi KB (nižší míra blokování TRX) → delší doba zotavení

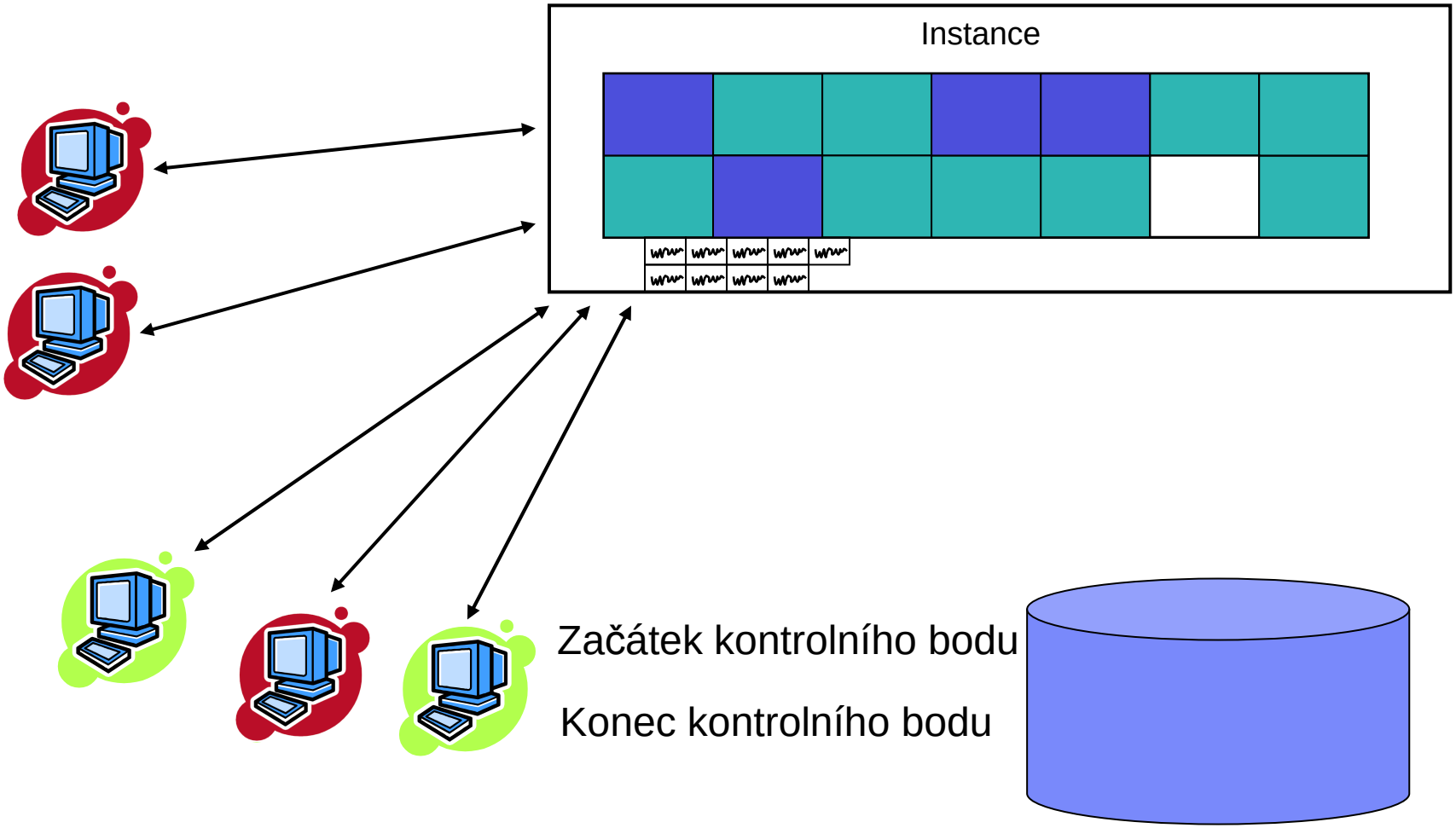
Řešení: neblokující (intervalové) kontrolní body

- Většina těchto kontrolních bodů neblokuje transakce
  - Výjimky tvoří:
    - Administrační kontrolní bod: onmode –c
    - Archivní kontrolní bod
    - Kontrolní bod vyvolaný nedostatkem zdrojů
      - Fyzický žurnál je z 75% plný
      - Stále musí být jeden blokující KB na jeden logický žurnál
- Fuzzy kontrolní body kompletně odstraněny
- LRU ladění může být podstatně zmírněno a nastavení může být podstatně méně agresivní
  - LRU_MAX_DIRTY/LRU_MIN_DIRTY=80/70

Jak pracovaly kontrolní body dříve …



… a jak pracují nyní ?



Monitorování

onstat –g ckp

Auto Checkpoints=On RTO_SERVER_RESTART=60 seconds Estimated recovery time 7 seconds

							Critical Sections						Physical Log		Logical Log	
Interval	Clock Time	Trigger	LSN	Total Time	Flush Time	Block Time	# Waits	Ckpt Time	Wait Time	Long Time	# Dirty Buffers	Dskflu /Sec	Total pages	Avg /Sec	Total Pages	Avg /Sec
1	18:41:36	Startup	1:f8	0.0	0.0	0.0	0	0.0	0.0	0.0	4	4	3	0	1	0
2	18:41:49	Admin	1:11c12cc	0.3	0.2	0.0	1	0,0	0.0	0.0	2884	2884	1966	162	4549	379
3	18:42:21	Llog	8:188	2.3	2.0	2.0	1	0.0	2.0	2.0	14438	7388	318	10	65442	2181
4	18:42:44	*User	10:19c018	0.0	0.0	0.0	1	0.0	0.0	0.0	39	39	536	21	20412	816
5	18:46:21	RTO	12:188	54.8	54.2	0.0	30	0.6	0.4	0.6	68232	1259	210757	1033	150118	735

Max Plog pages/sec	Max Llog pages/sec	Max Dskflush Time	Avg Dskflush pages/sec	Avg Dirty pages/sec	Blocked Time
8796	6581	54	43975	2314	0

- sysmaster:syscheckpoint
  - Historie posledních 20 kontrolních bodů
- sysmaster:sysckptinfo
  - Informace o neblokujících kontrolních bodech

Automatické ladění LRU

- LRU vyprazdňování může být s neblokujícími kontrolními body méně agresivní
- LRU vyprazdňování se automaticky ladí na více agresivní, pokud:
  - Pokud jsou přepisovány často používané stránky, pak se LRU vyprazdňování nastaví o 1% více agresivnější
  - Pokud dojde k FG zápisu, pak se LRU vyprazdňování nastaví o 5% více agresivnější
- Dynamické nastavení pro automatické ladění LRU prostřednictvím onmode –wm nebo –wf:

  AUTO_LRU_TUNING=[1,0] -- zapnuto (1) nebo vypnuto (0).

  AUTO_LRU_TUNING=0,min=50,max=60 -- vypnuto a LRU_MIN_DIRTY/LRU_MAX_DIRTY je nastaveno na 50 a 60

- Konfigurační parametr: AUTO_LRU_TUNING
  - 0 – při startu instance vypnuto
  - 1 - při startu instance zapnuto

Automatické ladění počtu AIO VP

- Pokud databázový server zjistí, že I/O operace nad *cooked* zařízeními nejsou dostatečně rychle obslouženy, dojde automaticky ke zvýšení počtu AIO VP a čističů stránek
- Konfigurace
  - Konfigurační parametr: AUTO_AIOVPS
    - 0 – vypnuto
    - 1 - zapnuto
  - Dynamicky
    - onmode –wm|-wf AUTO_AIOVPS=0|1

Optimalizace práce s fyzickým žurnálem

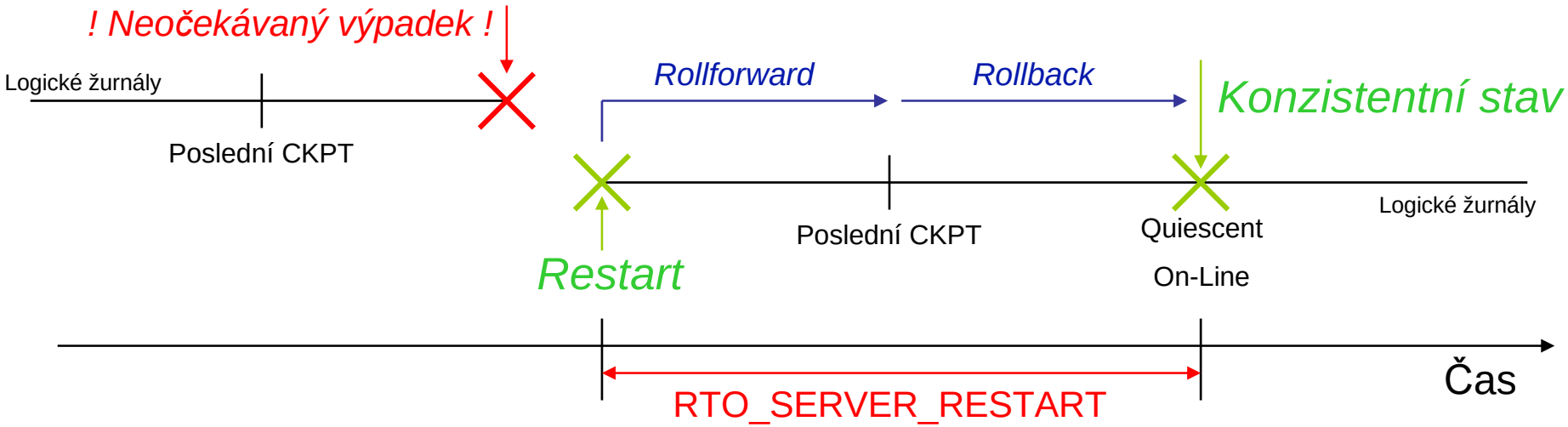
- IDS není již třeba restartovat, pokud dojde k přesunutí fyzického žurnálu do jiného databázového prostoru nebo ke změně jeho velikosti:
- PHYSDBS nelze již modifikovat „za běhu“:
  - Původní nedokumentovanou a nepodporovanou funkcionalitu změny parametru PHYSDBS a/nebo PHYSFILE za účelem změny umístění nebo velikosti fyzického žurnálu již nelze použít
- Musí se použít onparams, onmonitor nebo nové SQL API.

Automatické spouštění kontrolního bodu

- KB se spouští automaticky na základě velikosti fyzického a logických žurnálů tak, aby nedocházelo k blokování
  - Doporučení:
    - Zvětšit velikost fyzického žurnálu
    - Zvětšit velikost logických žurnálů
    - Zvýšit hodnoty LRU
- Defaultně je nastaveno
- onmode –wm|-wf AUTO_CKPTS=0|1
- Konfigurační parametr: AUTO_CKPTS 0|1

Stanovení maximální doby zotavení (RTO)

- RTO = Recovery Time Objective
- RTO_SERVER_RESTART
  - 0 – vypnuto (přednastaveno)
  - Od 60 do 1800 vteřin
  - Dynamicky konfigurovatelné prostřednictvím onmode –wm|-wf
- RTO_SERVER_RESTART = čas, do kdy má databázový server provést fast recovery, tedy uvést se do konzistentního stavu po neočekávaném výpadku
- Na základě nastavení se dynamicky určuje okamžik, kdy má dojít k zápisu bufferů na disk (LRU zápisy) nebo kdy se má provést kontrolní bod
- Nastavení CKPTINTVL se ignoruje



RTO_SERVER_RESTART

- Pokud je nastavený parametr RTO_SERVER_RESTART, pak
  - Při transakčním zpracování dochází k nárůstu fyzického žurnálování:
    - 3%-6% ztráty výkonnosti je bohatě pokryto výhodou neblokování transakcí při kontrolních bodech
- Důvody nárůstu fyzického žurnálování
  - Z důvodu předvídatelnosti doby zotavení je třeba, aby byly k dispozici všechny stránky potřebné pro logické zotavení
    - Všechny potřebné stránky jsou umístěny do fyzického žurnálu
    - V průběhu fyzické fáze zotavení jsou umístěny do vyrovnávací paměti
    - Logická fáze zotavení využívá pouze stránky z vyrovnávací paměti a neprovádí žádná nepředvídatelná čtení z disku
- Velikost fyzického žurnálu by měla být 110% velikosti vyrovnávací paměti při její velikosti do 4 GB
  - Příliš malá vyrovnávací paměť – diskové zápisy a čtení při fyzickém zotavení
  - Příliš malý fyzický žurnál – časté spouštění blokujících KB

Děkuji za pozornost!