DJ2 - Dotazování nad proudy dat

Jaroslav Pokorný

Obsah

„Staronové“ problémy

Některé opakující se výzvy:

- správa nejistých informací,
- důvěrnost a bezpečnost dat,
- proudy dat a síťová data,
- samoladící se a adaptivní systémy,

„Staronové“ problémy

Některé opakující se výzvy:

- správa nejistých informací,
- důvěrnost a bezpečnost dat,
- proudy dat a síťová data,
- samoladící se a adaptivní systémy.

Proudy dat

Proud dat je uspořádaná dvojice (s , Δ), kde s je posloupnost n-tic a Δ je posloupnost pozitivních intervalů reálného času

Proudy dat se vyskytují v mnoha moderních aplikacích jako:

- monitorování sítí,
- senzorové sítě (provoz na dálnicích, turisté v krajině, geografická data, medicínská data),
- aplikace využívající značky RFID,
- záznamy telefonních rozhovorů,
- lékařské záznamy,
- finanční aplikace (např. vývoj cen na burze),

Proudy dat

- sledy záznamů o kliknutí (click-streams),
- zpracování záznamů transakcí (záznamy telefonních hovorů a použití kreditních karet),
- provoz Internetu: monitorování chování uživatelů v rozsáhlých webech (portály a vyhledávače) pomocí web logů,
- data systémů monitorujících provoz počítačových sítí či vyhodnocující digitální videosignál z bezpečnostních kamer,
- data vznikající při výrobě, v nepřetržitých provozech,
- monitorování bojových akcí v moderních armádách (smart uniform“).

Proudy dat

Př.: proudy dat obsahující RFID značky generované z událostí získaných RFID čtečkami.

- Proudy RFID dat jsou filtrovány, agregovány, transformovány a harmonizovány tak, že události, které s nimi souvisí, mohou být monitorovány v reálném čase.
- Zpracování událostí je datacetrické, vyžaduje zpracování ve vnitřní paměti na základě sady pravidel a zpracování dotazů.
- Typický příklad - nákladní vozy vezoucí náhradní díly označené značkami RFID, které projíždějí kontrolovaným pásmem.

Proudy dat

Nové databázové problémy:

- senzory mohou produkovat kontinuální (možná nekonečné) proudy dat,
- podobné distribuovaným databázím, ale rozšířené o vlastnosti související s reálným časem,
- stupeň vyhodnocování produkovaných dat je mnohem vyšší než v distribuovaných DBMS,
- neexistuje žádný praktický způsob pro extrakci a natahování dat do společné databáze ke každému jejich výskytu,
- strategie zpracování a optimalizace a dotazů musí být redefinovány.

Senzorová data a senzorové sítě

senzory: levná, bezdrátová zařízení s vlastním zdrojem,

senzorové sítě (SS): poskytují důležité zdroje dat a vytvářejí nové požadavky na řízení dat

2 základní kategorie senzorových dat:

- proudy transakčních dat
- proudy naměřených dat

Senzorová data a senzorové sítě

Senzory mohou být instruovány dělat periodická měření a provádět předzpracování dat.

Typické operace:

- pošli data nebo kombinaci dat (průměr, součet apod.) do nějakého vybraného uzlu,
- ulož data (která mohou být zpracována později),
- pošli zprávu jako reakci na nějakou pozorovanou událost.

Zpracování proudů dat

Vlastnosti dat ze senzorů:

- přicházejí neustále (on-line),
- nelze předpokládat nic o pořadí dat vstupujících do zpracování ať už v jednom nebo více proudu dat,
- proudy dat mají obecně neomezenou velikost,
- jakmile jsou data z proudu zpracována, jsou smazána nebo archivována – nelze je jednoduchým způsobem opětovně získat (srovnej s klasickými DB),
- může se měnit jejich struktura (topologie),
- velký objem a rychlost proudících dat.

Zpracování proudů dat

- Tradiční DBMS nejsou vhodné pro zpracování proudů dat:
  - senzorové uzly produkují a dodávají data nepřetržitě bez přijímaní požadavků na tato data,
  - dotazy nad sebranými daty jsou méně časté než vkládání dat,
  - vytvořená data musí být často zpracovávána v reálném čase protože mohou reprezentovat události požadující rychlou odpověď,
  - dotazy běží nepřetržitě (kontinuální dotazy), protože proudy dat nikdy nekončí, takže mohou „vidět“, jak se mění podmínky systému během jejich volání,
  - kvůli omezením na paměť nemůže být celý proud dat umístěn na disk.

Zpracování proudů dat

Nové databázové architektury:

- systém řízení proudů dat (Stream Data Management System - SDMS)
- stroj pro zpracování proudů dat (Stream Processing Engine - SPE)

Příklad: Streambase vyvinutá Stonebrakerem
benchmark:finanční služby

relační DBMS: 11000 zpráv/s

Streambase: 300000 zpráv/s

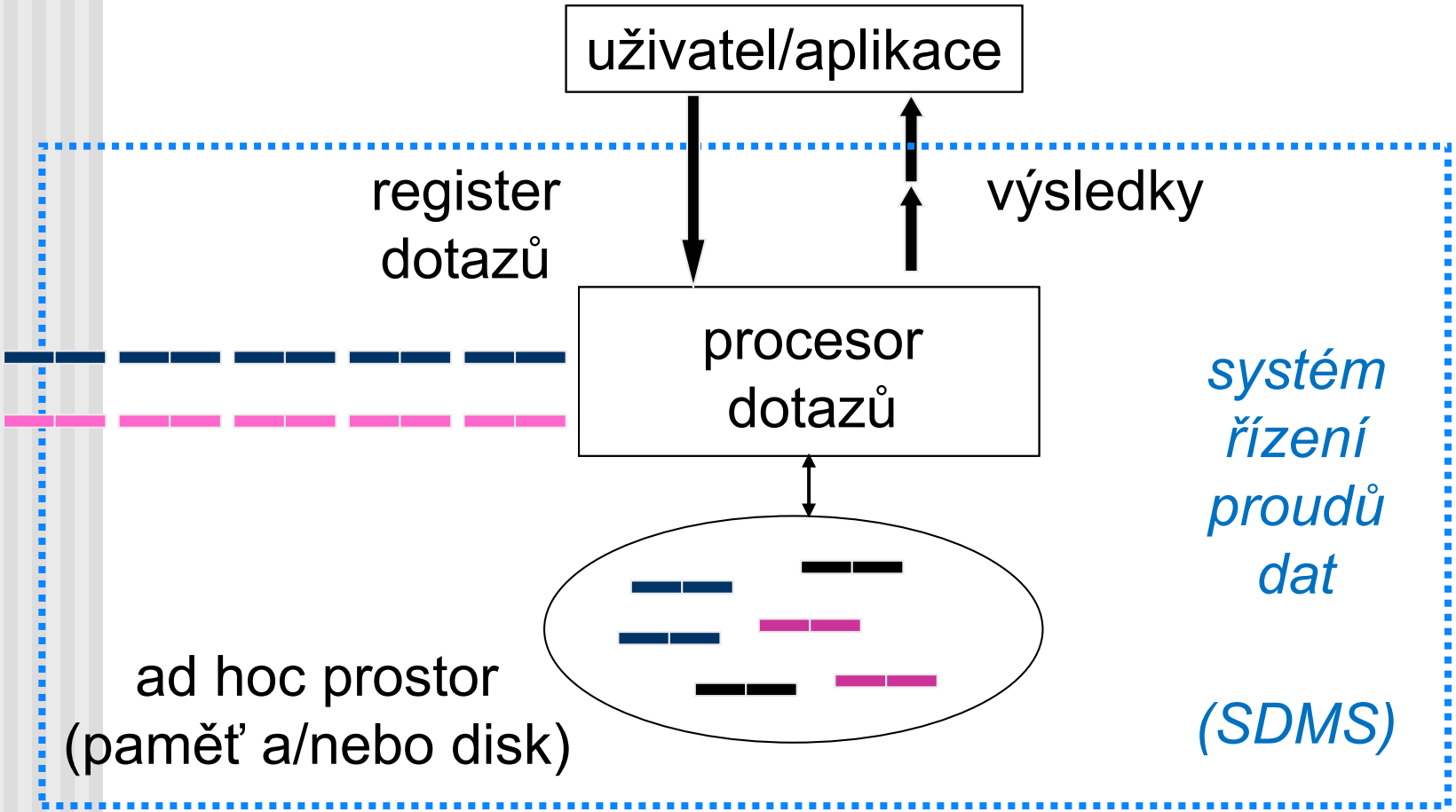
Obecněji: Kombinace SDMS (SPE) a relačního DBMS

Zpracování proudů dat

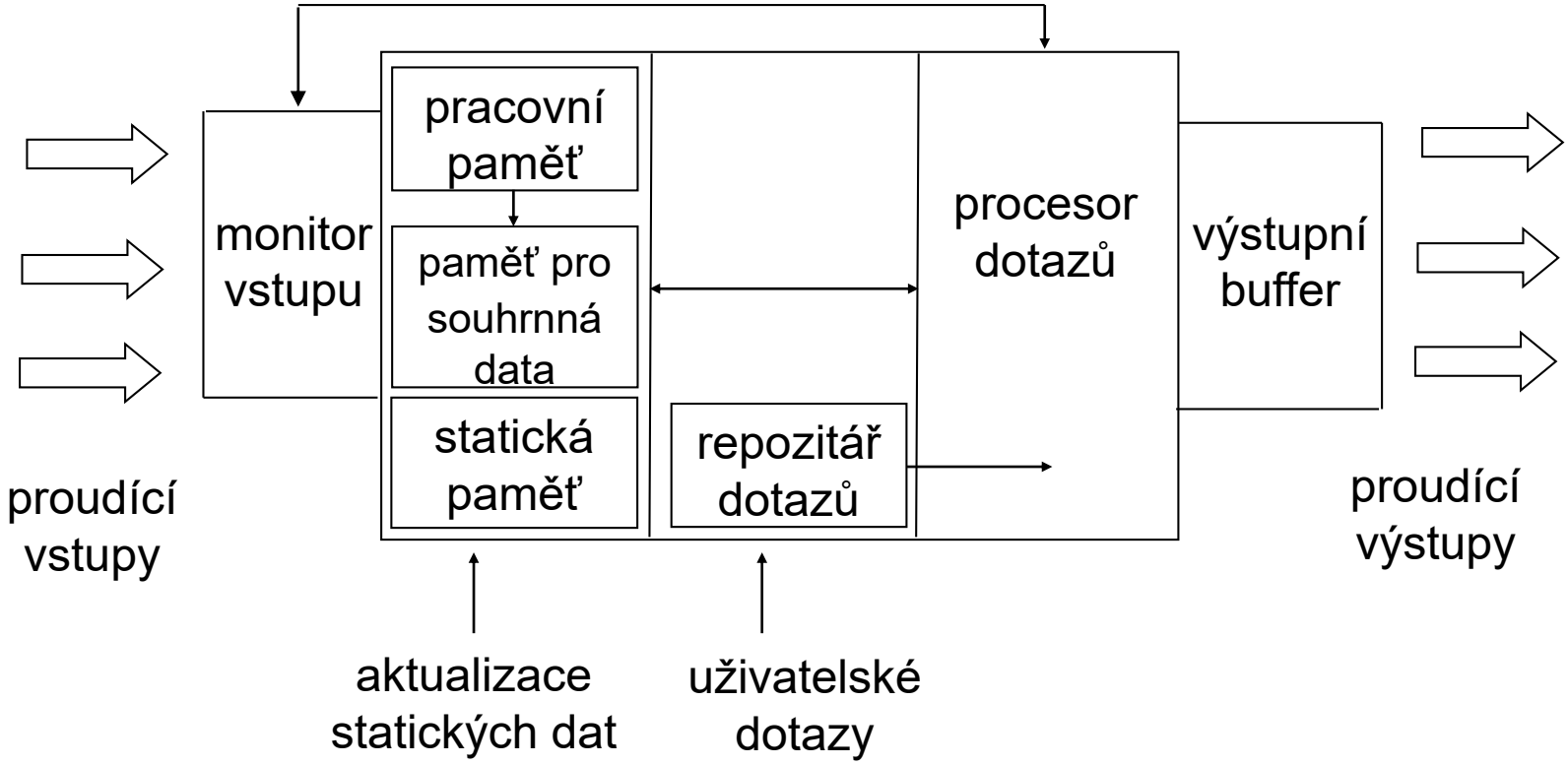
Vlastnosti SDMS:

- Místo trvalých relací pracuje SDMS s daty, která mohou do systému přitékat a zase odtékat.
- Výsledek dotazu není statický.
- Nemusíme nutně ukládat celý proud, ale stačí data potřebná pro okno dotazu.
- Real time vlastnosti důležitější než přesnost (I za cenu zahození dat, která nestíháme zpracovat)
- Některé SDMS integrují DBMS a umožnují například spojení tabulky a proudu.

Zpracování proudů dat



Zpracování proudů dat



Referenční architektura stroje pro zpracování proudů dat

Zpracování proudů dat

	SDMS	DBMS
zdroje	omezené (paměť, počítaní po n-ticích)	bohaté (paměť, disk, počítaní po n-ticích)
zpracování dotazu	rozumně složité, blízké zpracování v reálném čase	velmi sofistikované, analýzy
použití	zjištění, jaká data ukládat do databáze DBMS	audit výsledků dotazu z SDMS

SDMS vs. DBMS: globální rozdíly

Zpracování proudů dat

SPE v reálném čase: 8 pravidel (Stonebraker et al, 2005)

- zpracování bez zpoždění (bez požadavku na jejich uložení),
- podpora neprocedurálního dotazovacího jazyka vyšší úrovně, jakéhosi „StreamSQL“,
- umět ošetřit situace poruch v proudu dat (např. chybějící prvky),
- garantovat predikovatelné a opakovatelné výsledky,
- umět efektivně ukládat, přistupovat a modifikovat informace o stavu zpracování, když je to nutné,
- vysoká dostupnost v případě chyb systému (rychlé zotavení),
- možnost distribuce zpracování a tím škálovatelnosti systému, distribuce by měla být automatická a transparentní,
- odezva systému v reálném čase i pro objemné proudy dat.

Dotazování nad proudícími daty

Pro dotazování nad proudy dat je podstatný pojem modelu proudů dat.

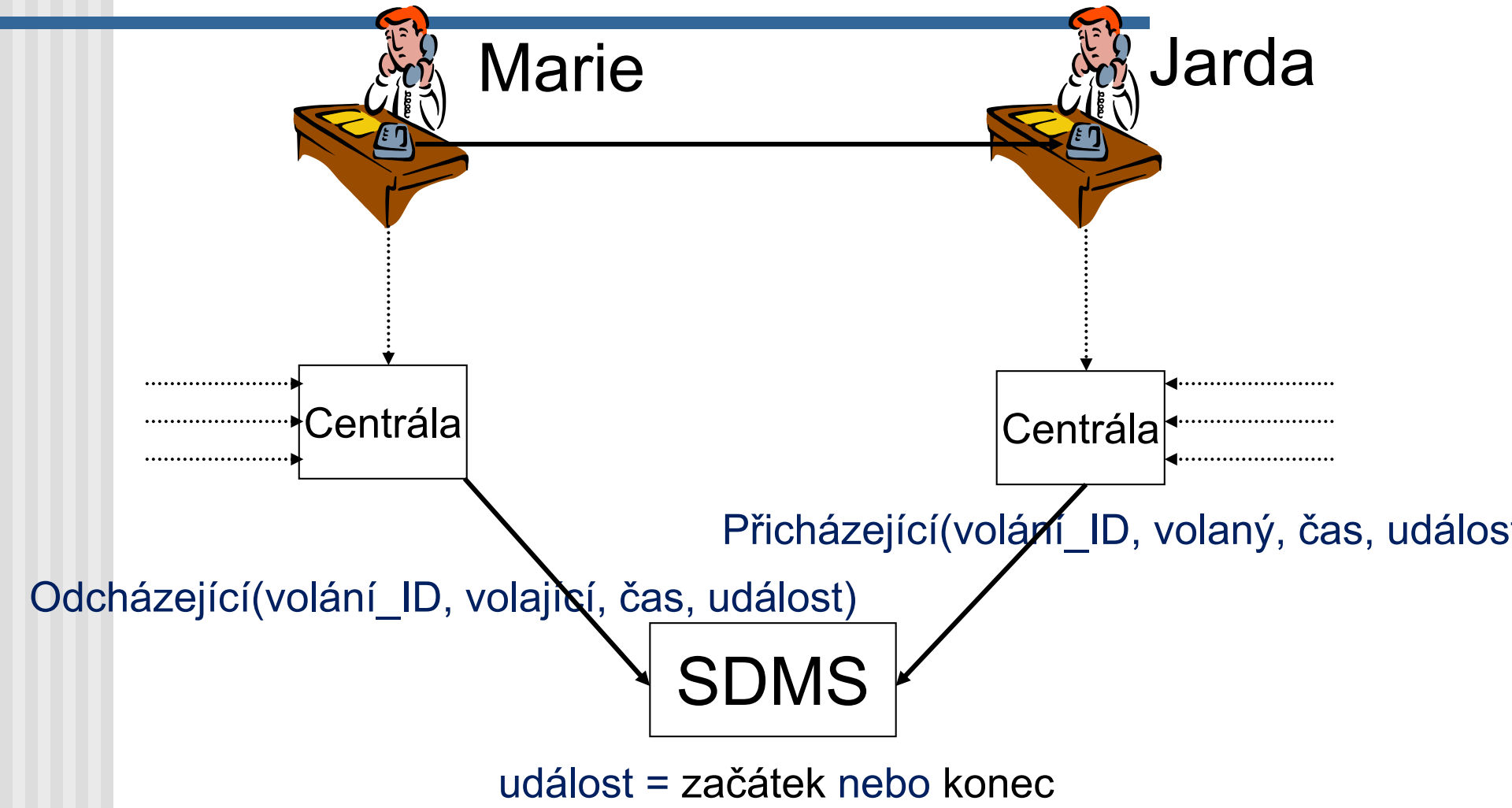
⇒ typy zpracování v SDMS :

- stream-to-stream (např. v systémech Aurora, Gigascope, STREAM),
- stream-to-relation (v jazyku Hancock),
- kombinace obojího.

⇒ typy dotazů v SDMS:

- one-time dotazy,
- kontinuální dotazy

Motivační příklad



Motivační příklad

D1: Najdi všechny odcházející hovory delší než 2 minuty.

```
SELECT   O1.volání_ID, O1.volající
FROM     Odcházející O1, Odcházející O2
WHERE    (O2.čas – O1.čas > 2
           AND   O1.volání_ID = O2.volání_ID
           AND   O1.událost = začátek
             AND   O2.událost = konec)
```

- výsledek vyžaduje neomezenou paměť
- může poskytovat výsledek jako proud dat
- výstup může být po 2 min., aniž by bylo vidět konec

Motivační příklad

D2: Spáruj volající a volané.

```
SELECT   O.volající, P.volaný
FROM     Odcházející O, Přicházející P
WHERE    O.volání_ID = P.volání_ID
```

- může se stále poskytovat výsledek jako proud dat
- vyžaduje neohraničenou temporární paměť
- … pokud proudy nejsou skoro-synchronizovány

Motivační příklad

D3: Spočítej celkový provolaný čas pro každého volaného.

```
SELECT  P1.volaný, sum(P2.čas – P1.čas)
FROM  Přicházející P1, Přicházející P2
WHERE  (P1.volání_ID = P2.volání_ID
     AND P1.událost = začátek  AND  P2.událost = konec)
GROUP BY  P1.volaný
```

- Výsledek nemůže být poskytnut pouze z proudu typu append-only
  - Jak? Aktualizace nepřetržitého výstupu?
  - Poskytovat běžnou hodnotu na požádání?
  - Co paměť?

Dotazování nad proudícími daty

Modelování a dotazování	SDMS	DBMS
model	tranzientní proudy	perzistentní relace
relace	posloupnost n-tic	množina/multimnožina n-tic
uspořádání n-tic	podle vstupu	neuspořádané
aktualizace dat	přidávání	modifikace
dotaz	perzistentní	tranzientní
dotazování v čase	kontinuální	jeden v čase
odpověď na dotaz	aproximovaná	přesná
vyhodnocení dotazu	jeden průchod	libovolné
plán dotazu	adaptivní	pevný

SDMS vs. DBMS: modelování a dotazování

Dotazování nad proudícími daty

- blokující operátor dotazu: není schopen produkovat první n-tici výsledku, aniž by „viděl“ celý vstup (např. agregační funkce, ORDER BY v SQL)
- neblokující operátor dotazu produkuje všechny n-tice výstupu před tím, než detekuje konec vstupu.
- funkce F na konečných posloupnostech může být spočítána neblokujícím operátorem právě když F je monotónní vzhledem k částečnému uspořádání na posloupnostech, kde $S_1 \sqsubseteq S_2$, právě když S_1 je počáteční podposloupností S_2.
- F je monotónní, jestliže $S_1 \sqsubseteq S_2 \Rightarrow F(S_1) \sqsubseteq F(S_2)$
- Proudy jsou obecně nekonečné posloupnosti: $\Rightarrow$ použití pouze neblokujících operátorů.

Dotazování nad proudícími daty

- Neblokující (monotónní) operátory na nekonečných posloupnostech: selekce, projekce, některé agregační funkce, např. COUNT a SUM
- NOT EXISTS, EXCEPT, NOT IN (vedou k univerzální kvantifikaci) nemohou být využity.
- Problémy:
  - monotónní spojení může mít blokující i neblokující implementaci (srovnej setřídění-slévání, hnízděné cykly)
  - nemonotónní operace je rozdíl
  - AVARAGE někdy může mít neblokující implementaci
- Obecněji: nutnost tzv. synopsí (aproximativní agregační funkce, přehledů)

Dotazování nad proudícími daty

Konjunktivní dotaz nad více proudy dat vyžaduje hlubší analýzu pro zajištění jeho vyčíslitelnosti v omezené paměti.

Př. Proudy S(A, B, C) a T(C, D) s celočíselnými doménami pro všechny atributy a dotazy:

D1: S(A > 10)[A]

D2: (S*T)(S.C > 10 AND T.C < 20)[C]

D3: (S[B<D]T)(A > 1 AND A< 10)[A]

D1 lze zpracovat v konečné paměti pouze při zachování duplicit ve výsledku.

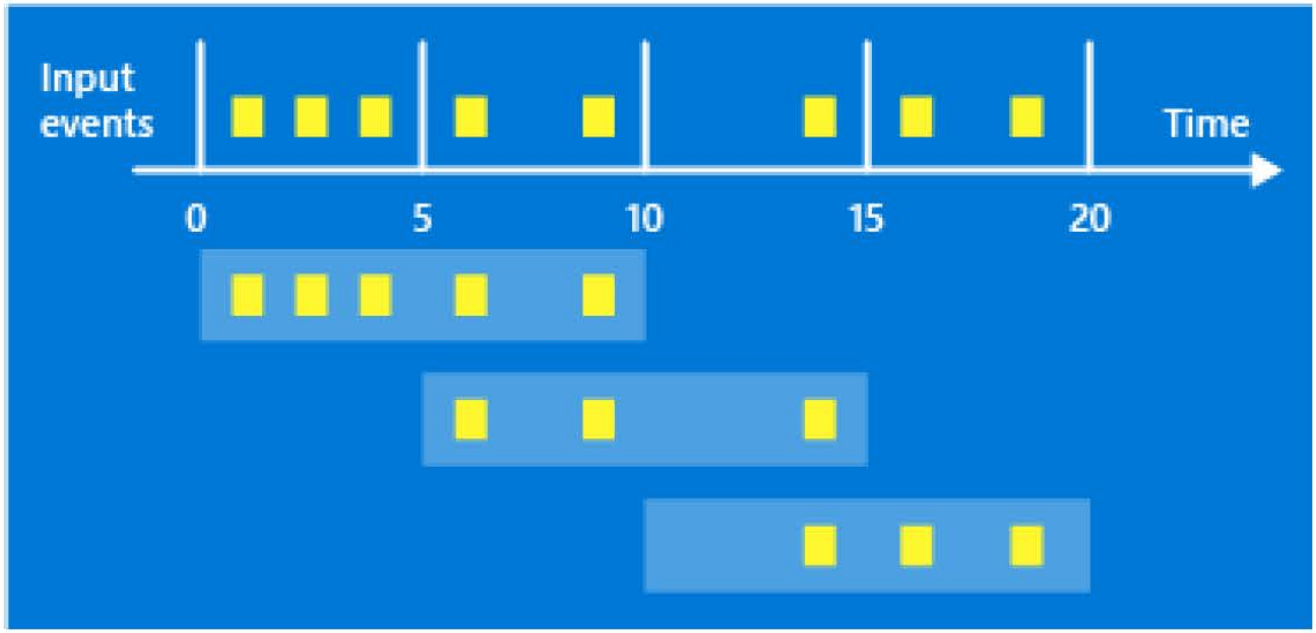
D2 též, dokonce s odstraněním duplicit. Stačí udržovat jistou informaci pro hodnoty atributu C ϵ (10, 20).

D3 je nutné duplicity eliminovat. Pro každou hodnotu A ϵ (1, 10) se udržují stávající hodnoty *min*(S.B) a *max*(T.D) přes všechny dosud zpracované n-tice.

Požadavky na procesor dotazů v SDMS

- Model dat a sémantika dotazů musí dovolovat operátory založené na uspořádání a čase.
- Použití aproximativních agregačních struktur (synopsí): $\Rightarrow$ dotazy využívající agregace nemohou vracet přesné odpovědi.
- Plány dotazu nad proudy dat nemohou používat blokující operátory.
- Paměťová omezení: $\Rightarrow$ není možné používat víceprůchodové algoritmy. On-line algoritmy nad proudy dat jsou omezeny pouze na jeden průchod daty.
- Aplikace, které monitorují proudy dat, musí v reálném čase rychle reagovat na neobvyklé hodnoty dat.
- Dlouhotrvající dotazy mohou v průběhu svého života narazit na změny (např. rychlosti proudu), což může vést ke změně plánu vyhodnocení dotazu.
- Pro sdílené volání více dlouhotrvajících dotazů je třeba zajistit škálovatelnost dotazovacího systému.

Metoda posuvné okno



Metoda posuvné okno

Čas. razítko	n-tice
1000	10,0.1
1002	15,0.14
1004	33,4.4
1006	23,56.33
1008	34,4.4
1010	20,0.2
1012	45,23.44
1014	30,0.3
2000	17,1.3

Dotaz

SELECT * FROM S[rows 3]

výsledek v čase 1003:

(10,0.1),(15,0.14)

výsledek v čase 1007:

(15,0.14), (33,4.4), (23,56.33)

Metoda posuvné okno

Proč posuvná okna?

- aproximační technika pro omezenou paměť,
- přirozené pro aplikace (důraz na současná data),
- dobře specifikovaná a deterministická sémantika,
- délka okna:
  - v čase (např. 10 minut),
  - v počtu příchozích prvků: každých dalších K prvků, evidujeme je a zahodíme K nejstarších prvků
- princip FIFO (fronta – queue),
- rozšíření relační algebry nebo SQL zpracovávat posuvná okna je netriviální.

CQL

Continuous Query Language (z r. 2006)

- Proud v CQL je multimnožina dvojic (*s*, *T*), kde s je element a *T* je časové razítko).
- Dotazy jsou aktualizovány na základě času.

Typy operací:

stream-to-relation (posuvné okno)

relation-to-relation (selekce, projekce a agregace)

relation-to-stream (konvertuje relace do proudu)

Příklad:

Aukce(id_aukce, prodavač, T)

Nakupy(id_aukce, kupující, cena, T)

CQL

Continuous Query Language (z r. 2006)

- Proud v CQL je multimnožina dvojic (*s*, *T*), kde s je element a *T* je časové razítko).
- Dotazy jsou aktualizovány na základě času.

Typy operací:

stream-to-relation (posuvné okno)

relation-to-relation (selekce, projekce a agregace)

relation-to-stream (konvertuje relace do proudu)

Příklad:

Aukce(id_aukce, prodavač, T)

Nakupy(id_aukce, kupující, cena, T)

CQL

D4.: Kolik utratil Láďa na aukcích od Jiřího za poslední den?

```
SELECT SUM(N.cena)                    okno založené na čase
FROM Aukce AS A, Nakupy [RANGE 1 DAY] AS N
WHERE A.id_aukce = N.id_aukce
            AND A.prodavač = 'Jiří, AND N.kupující = ‚Láďa'
```

D5: Kolik utratil Láďa za poslednich 10 aukcí od Jiřího?

```
SELECT SUM(N.cena)                    okno založené na n-ticích
FROM Aukce AS A, Nakupy [ROWS 10] AS N
WHERE A.id_aukce = N.id_aukce
            AND A.prodavač = 'Jiří, AND N.kupující = 'Láďa'
```

CQL

D6: Uvažuj 10 posledních aukcí od každého prodavače a vrať prodavače s nejdražším prvkem.

```
SELECT A.prodavač, MAX(N.Cena)
FROM Aukce [PARTITION BY A.prodavač ROWS 10] AS A,
        Nákupy AS N
WHERE A.id_aukce = N.id_aukce
GROUP BY A.prodavač
```

rozdělena okna

- dotazy D4 a D6 jsou stream-to-relation dotazy
- stream-to-stream dotazy nejsou podporovány

Poznámka: CQL byl využit pro implementaci Apache Samza Streaming SQL.

Streaming SQL language

Původně: StreamSQL

- Podobná syntaxe jako CQL
- Na rozdíl od CQL má operace pro stream-to-stream dotazy

Nové operace:

- selekce z proudu
- spojení proud-relace
- sjednocení a merging
- okno nad proudem + agregace
- okna nad dvěma proudy + spojení

Příklady dotazů

Př.: stream-to-stream operace

UNION – slije dva proudy do jednoho na základě času

Příklad:

České_aukce(aukce, T)

Slovenské_aukce(aukce, T)

D6:

SELECT * FROM České_aukce UNION Slovenské_aukce

Výsledek:

(čes.aukce, 2), (slov.aukce, 3),
(slov.aukce, 5), (čes.aukce, 8), (čes.aukce, 9)

n-tice jsou proloženy se zachováním uspořádání

Příklady dotazů

D7: Dotaz s posuvným oknem pro posledních 10 elementů:

```
SELECT AVG(cena)
FROM Proud [SIZE 10 ADVANCE 10 TUPLES] WHERE
     cena > 100.0
```

Proud kontinuálně počítá průměrnou hodnotu ceny z posledních 10 n-tic, uvažuje ale jen ty, jejichž cena je větší než 100.0.

D8: Vytvoř proud nejdražších aukcí.

```
CREATE STREAM
SELECT A.prodavač, MAX(N.Cena)
FROM Aukce [SIZE 10 ADVANCE 1 TUPLES PARTITIONS BY
   A.prodavač] AS A, Nákupy AS N
WHERE A.id_aukce = N.id_aukce
GROUP BY A.prodavač
```

Současný stav

Podpora Streaming SQL v projektech:

- Apache Storm[1],
- Apache Flink[2],
- Apache Kafka[3],
- Apache Samza[4]

[1]https://storm.apache.org
[2]https://flink.apache.org/
[3]https://kafka.apache.org/
[4]https://samza.apache.org/

PipelineDB

PipelineDB je velmi výkonné rozšíření PostgreSQL pro zpracování SQL kontinuálních dotazů na datech časových řad.

Zdroje

1. Data Stream Management Systems from the example of PipelineDB, Andreas Egloff, Hochschule für Technik Rapperswil, Davos 2016
2. Streaming SQL With PipelineDB, Derek Nelson (PipelineDB), Konference Postgres Open 2015, www.youtube.com/watch?v=nubwRF54M2M
3. PipelineDB dokumentace, http://docs.pipelinedb.com/introduction.html

Závěry

- Vztah k Big data: rychlost (tj. Velocity – jedno z charakteristických V), tj. jak rychle jsou data produkována jak rychle musí být zpracována pro analýzu na požádání
- Vztah k BI: analýza proudů událostí z živých zdrojů s cílem nalézt vzory a význačné identifikátory za účelem spuštění nějaké akce

Zdroje

1. Foundations of streaming SQL: stream & table theory, Anton Kedin (Google), DataWorks Summit 2018, https://www.youtube.com/watch?v=lYx-Mg3Jdb0
2. Data stream management system, en.wikipedia.org/wiki/Data_stream_management_system