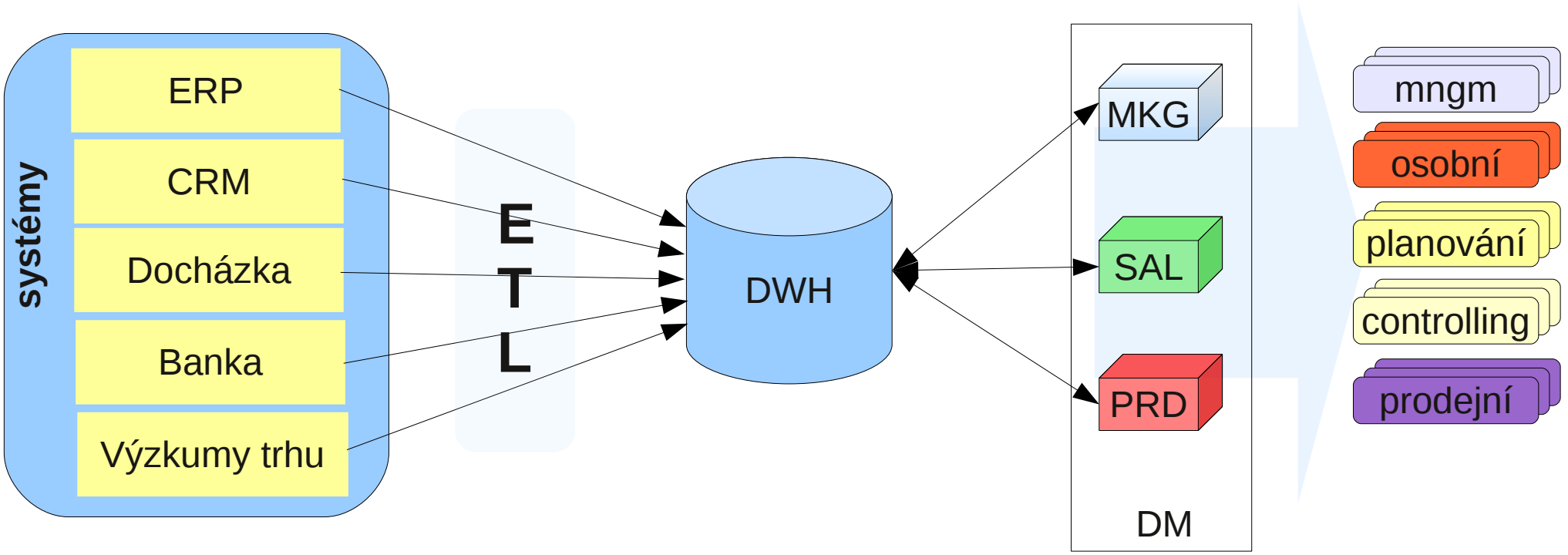
GridSQL a pg-pool II

Vratislav Beneš
benes@optisolutions.cz

Agenda

1. Datové sklady a datová tržiště
2. pg-pool II
    1. Infrastrukutra
    2. Využití pro datové sklady
3. GridSQL
    1. Infrastuktura
    2. Vytvoření DB a tabulek
    3. Nahrávání dat
    4. Test výkonosti
    5. Přednosti a slabiny
4. Závěrečné shrnutí

Datové sklady a datová tržiště



Požadavky na systém pro Data Warehouse

- Dostupnost
- Stabilita
- Výkonost
- Velké objemy dat
- Vysoký počet uživatelů
- Zpracování časově náročných úloh
- Speciální funkce (statistika, výpočty apd)

Požadavky na systém pro Data Marts (DM)

- Vysoká výkonost – odezvy v sekundách
- Stabilita
- Menší objemy dat než DWH
- Méně uživatelů než DWH

pg-pool II

pg-pool II

- Middleware
- Open Source – BSD License
- dblink
- PostgreSQL jako backend
- http://pgpool.projects.postgresql.org/

Function/Mode	Raw Mode (*3)	Replication Mode	Master/Slave Mode	Parallel Query Mode
Connection Pool	X	O	O	O
Replication	X	O	X	(*1)
Load Balance	X	O	O	(*1)
Failover	O	O	O	X
Online recovery	X	0	(*2)	X
Parallel Query	X	X	X	O
Required # of Servers	1 or higher	2 or higher	2 or higher	2 or higher
System DB required?	no	no	no	yes

zdroj: pgpool.projects.postgresql.org

pg-pool II – využití pro datové sklady

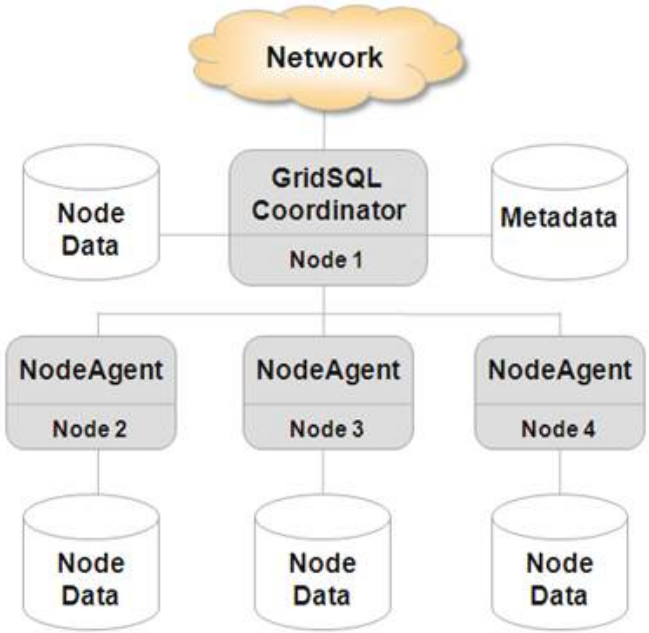
Load balance
+ rozdělení zátěže při dotazování DWH
+ replikace => zvýšená dostupnost
+ odolnost proti výpadků *n-1* nodů
+ nekomplikovaná správa (nevyžaduje servisní DB)
- *n* násobný objem dat
- nikterak neurychlý dotaz 1 uživatele (u menších oddělení/firem ne zcelan neobvyklá situace)

Parallel query
+ rozdělení zátěže při dotazování DWH
+ urychlý dotaz i 1 uživatele
- není odolný vůči výpadku nodu/nodů
- komplikovaná správa (vyžaduje servisní DB s pravidly pro dělení dat)
- problém u JDBC klientů (verze 2.x byla nepoužitelná)
    - dotaz vrátil data pouze z jednoho nodu :-(
    - stejný dotaz zadaný v psql vrátil správný výsledek

GridSQL

GridSQL

- EnterpriseDB
- Open Source – GNU GPL 2
- Nic nesdílející cluster
- Paralelní zpracování dotazu
- Velké objemy dat
- Vysoký výkon
- PostgreSQL jako backend
- JDBC only
- http://sourceforge.net/projects/gridsql/



Požadavky

- Java SE
- PostgreSQL pro servisní DB
- 1 a více strojů s PostgreSQL pro nody
- Vícejádrové CPU nebo SMP server
- Dostatek RAM
- Rychlé diskové úložiště

GridSQL - vytvoření DB a tabulek

Konfigurace

- config/gridsql.config
- Port gridu
- Servisní DB
- Počet nodů gridu
- Servery s nody (PostgreSQL)
- Vytvoření DB s metadaty: *bin/gs-createmddb.sh –u admin –p password*
- Start koordinátoru: *bin/gs-server.sh*

Vytvoření DB

- Název DB
- Výběr nodů pro DB: *gs-createdb.sh -d cspug -h 127.0.0.1 -s 6453 -u admin -p admin -n 1,2,3,4*
- Vlastník
- Strat DB: *gs-dbstart.sh -u admin -p admin -d cspug*

Vytvoření tabulek

- PARTITIONED - faktové tabulky rozřezané do nodů podle klíče – *partitioning key*
- REPLICATED – obsah tabulky je na všech nodech stejný
- Stejná konstrukce jako pro PostgreSQL

GridSQL – vytvoření DB - scripty

Databáze

gs-createdb.sh -d cspug -h 127.0.0.1 -s 6453 -u admin -p admin -n 1,2,3,4

Faktová tabulka – rozdělená podle času (time_id)

```
CREATE TABLE data
(
  time_id smallint NOT NULL,
  outlet_id integer NOT NULL,
  product_id integer NOT NULL,
  sales_pcs real,
  sales_value real,
  sales_vol real,
  CONSTRAINT pk_data PRIMARY KEY (time_id, outlet_id, product_id)
) PARTITIONING KEY time_id ON ALL;
```

Atributová tabulka – replikovaná na všechny nody

```
CREATE TABLE outlets
(
  outlet_id integer NOT NULL,
  outlet_type_id smallint,
  outlet_type character varying(64),
  chain_id smallint,
  chain_descr character varying(64),
  CONSTRAINT pk_outlets PRIMARY KEY (outlet_id)
) REPLICATED;
```

GridSQL – narávání dat

INSERT

INSERT [INTO] table_name [(column_name,...)]
VALUES ((expression),...)

INSERT [INTO] table_name [(column_name,...)]
SELECT ...

COPY

COPY tablename [(column [, ...])]
FROM { 'filename' | STDIN }
[[WITH]
[DELIMITER [AS] 'delimiter']
[NULL [AS] 'null string']

BULK INSERT

gs-loader.sh –d BIGDB –u myuser –p mypassword –h localhost
-i /home/extendb/mig/order_fact.tbl -t orders -f '|'
-k 100000,20,1 –y /home/extendb/mig/bad

GridSQL – přednosti a slabiny

Přednosti

- Obrovský výkon SELECT dotazů
- Témeř lineární nárůst výkonosti spojený s dodatečným nodem
- Vysoká shoda SQL s PostgreSQL (datové typy, operátory, agr. fce ...)
- PostgreSQL jako back-end
  - Přímočará správa
  - Možnost ladění výkonosti každého nodu
- Kombinace různého HW i platforem

Slabiny – resp. co autor prezentace neodhalil

- Malá výkonost u *UPDATE*, *INSERT* dotazů
- Neumožňuje práci se schématy
- Nemožnost měnit počet nodů bežící DB
- Neumožňuje vytvářet uživatelské funkce
- JDBC only
- Není odolný vůči výpadku nodu/ů

Řešení ala Kutil Tim

- INSERT, UPDATE – spouštět přímo na nodech => řádové zrychlení, ale koledujeme si o malér
- Lze využít uživatelských funkcí, které jsou vytvořeny přímo na nodech

Závěrečné shrnutí z pohledu využití pro DWH a DM

pg-pool II

+ výborný nástroj pro Load Balance

+ vhodný pro Data Warehouse

- ve spojení s JDBC se neosvědčilo Parellel Query (u verze 2.x)

GridSQL

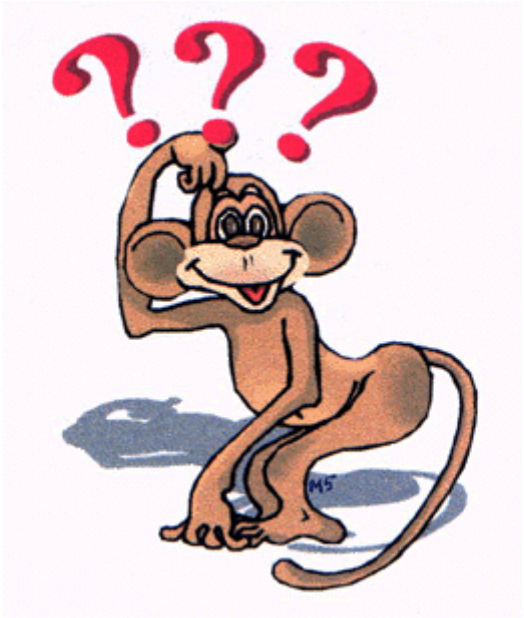
+ velmi výkonný datový zdroj pro reportovací systémy

+ vhodný pro Data Marts

- spolupracuje pouze přes JDBC

- není odolný vůči výpadku nodu/ů – plánováno do dalších verzí

Otázky



Děkuji za pozornost