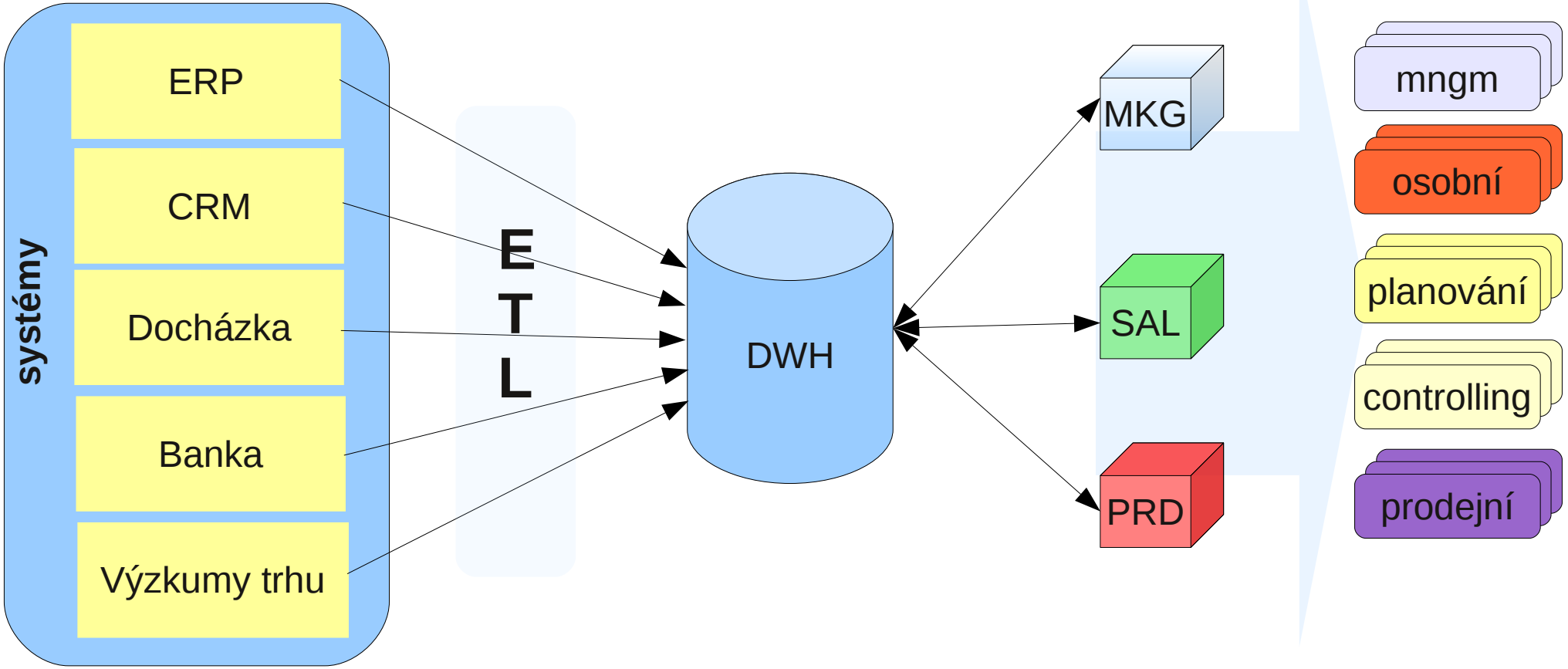
PostgreSQL jako platforma pro datové sklady

Vratislav Beneš
benes@optisolutions.cz

1. **Co to jsou datové sklady?**
2. **Požadavky na datový sklady**
3. **Technické řešení datového skladu**
4. **PostgreSQL a datové sklady**
5. **Paralelní data**
6. **GridSQL vs MSSQL 2005**
7. **Funkce v PostgreSQL**
8. **PostgreSQL a Business**
9. **PostgreSQL vs MSSQL vs MySQL**

Co to jsou datové sklady?

- jsou kritickou součástí Business Intelligence
- integrace veškerých data nutných k efektivnímu řízení firmy
- veškerá data umístěná na jednom místě
- snadno dostupná i pro rozdílné úrovně požadavků na detail



Požadavky na datový sklad

Bill Inmon – datové sklady jsou:

- subjektově orientované
- integrované
- zpětně neměnné
- v čase se vyvíjející

Subjektově orientované – cílem je vytvořit takovou strukturu, která je snadno srozumitelná pro koncového uživatele

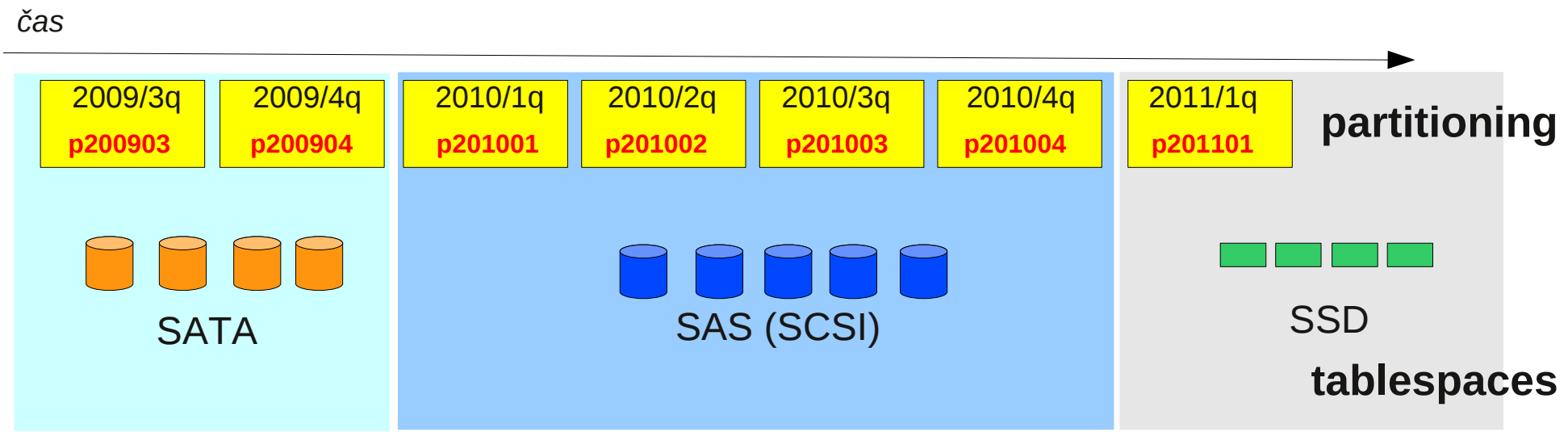
Integrované – do datového skladu směřují různorodá data, která je třeba sjednotit a vytvořit z nich logický celek

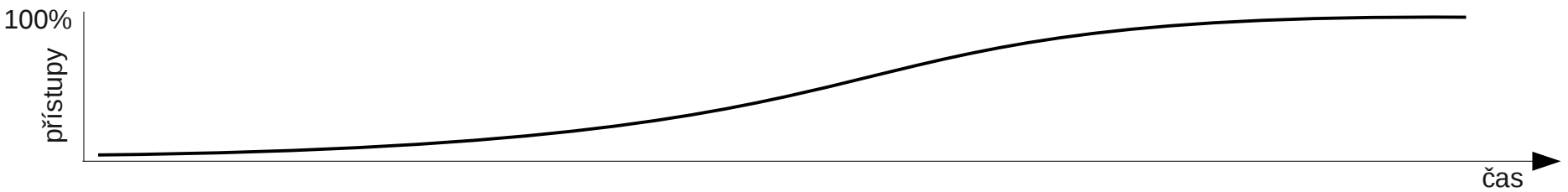
Zpětně neměnné – data nahraná do datového skladu by se měla prohlásit za spolehlivá a neměnná

V čase se vyvíjející – data do *DWH* se nahrávají po přesně definovaných časových intervalech (denně, týdenně, měsíčně ...)

Technické řešení datového skladu

- Relační databáze
- Velký objem dat *(CZ: jednotky až stovky GB, mobilní operátoři/banky TB, USA: PB)*
- ~~3. NF~~
- Redundance je povolená a přínosná





PostgreSQL a datové sklady

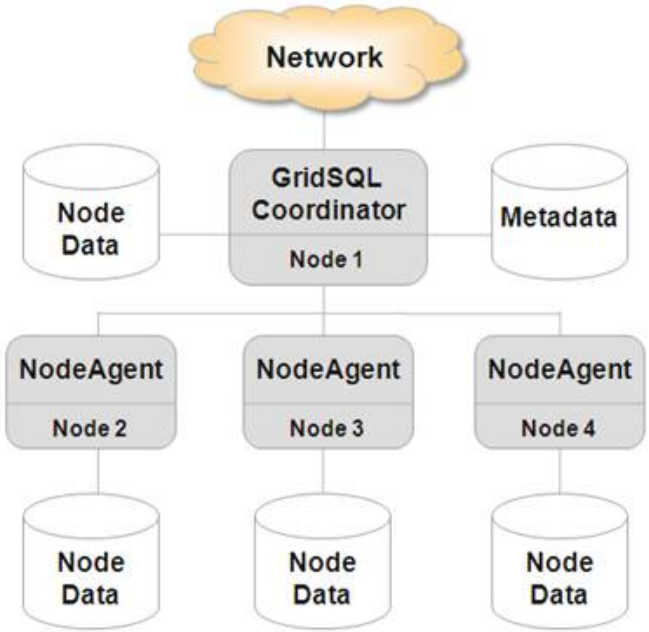
Požadavky na DBS

PostgreSQL 8.4 +

- tablespaces ✓
- partitioning ✓
- bulk insert ✓
- škálovatelnost **pg-pool**
- paralelní zpracování **GridSQL**

GridSQL

- EnterpriseDB
- Open source
- nic nesdílející cluster
- paralelní zpracování dotazu
- velké objemy dat
- vysoký výkon
- PostgreSQL jako backend
- **-** JDBC only
- http://sourceforge.net/projects/gridsql/

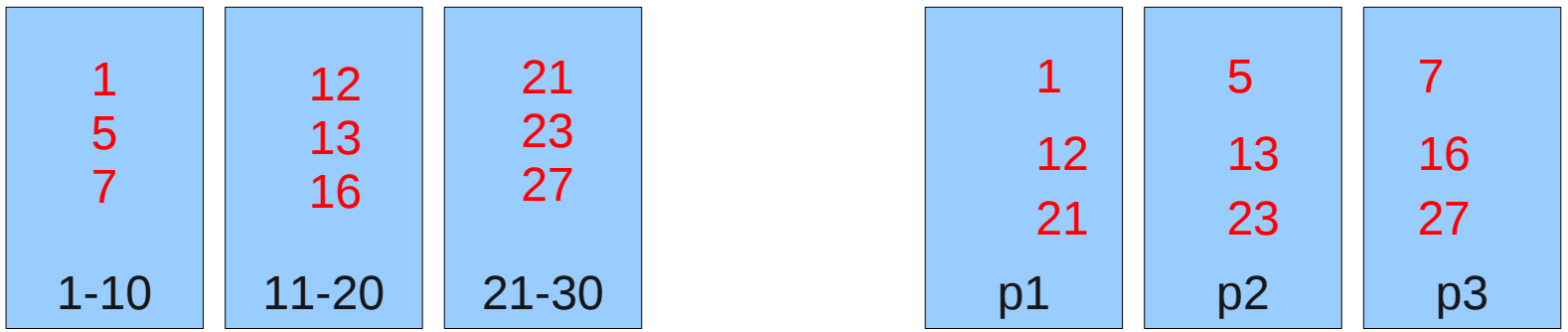


Paralelní data

Table a:
time_id
sales

partitions podle time_id

time_id



select sum(sales) from a

(1+5+7+12+13+16+21+23+27)|CPU0

select sum(sales) from p1 & select sum(sales) from p2 & select sum(sales) from p3

(1+12+21)|CPU0 & (5+13+23)|CPU1 & (7+16+27)|CPU2

GridSQL vs MSSQL 2005

Generování XLS reportů pro marketing

- 1 strana: 26 dotazů
- report: 5 stran
- reportů: 13
- celkem: 1690 dotazů

ROLAP

- faktová tabulka 32 046 794 záznamů
- tabulky dimenzí celkem 5 266 záznamů
- ~10GB dat

MDX dotaz

```
with member [periods.monthly].[MAT-1] as '[periods.monthly].[MATval-1]'
  member [periods.monthly].[MAT] as '[periods.monthly].[MATval]'
  member [periods.monthly].[MAT Changes] as '(([periods.monthly].[MAT] / [periods.monthly].[MAT-1]) - 1.0)'
  member [periods.monthly].[YTD] as '[periods.monthly].[YTDval]'
  member [periods.monthly].[YTD-1] as '[periods.monthly].[YTDval-1]'
  member [periods.monthly].[YTD Changes] as '(([periods.monthly].[YTD] / [periods.monthly].[YTD-1]) - 1.0)'

select NON EMPTY Union(Crossjoin({LastPeriods(13.0, ClosingPeriod([periods.monthly].[month]))}, {[Measures].[sales value]}), Crossjoin({[periods.monthly].
[MAT-1], [periods.monthly].[MAT], [periods.monthly].[MAT Changes], [periods.monthly].[YTD-1], [periods.monthly].[YTD], [periods.monthly].[YTD Changes]},
{[Measures].[sales value]})) ON COLUMNS,
  NON EMPTY Order(Hierarchize(Union({[segment.segment].[All segment.segments]{segment_param}}, [segment.segment].[All segment.segments]
{segment_param}.Children)), ([periods.monthly].[MAT], [Measures].[sales value]), DESC) ON ROWS
from [fbel]
where  [outlet type.outlet type].[All outlet type.outlet types].[Hypermarkets]
```

MSSQL 2005 - IBM 3500 – 2 x DC Xeon, 4GB RAM - Win 2003, 6 x SAS 15k RAID 10, MOLAP
jeden dotaz: ~90s => **celkem: ~42.25h**

Pentaho BI + GridSQL 2 - IBM 3650, QC Xeon, 16GB RAM, Fedora 13, PostreSQL 8.4.4, SSD, ROLAP
jeden dotaz: ~12s => **celkem: ~5.41h**

Funkce v PostgreSQL

Statistika pomocí R-project

- rozšíření základních statistických funkcí SQL
- PL/R (http://www.joeconway.com/plr/)

select time_id, outlet_id, **maxfreq(price)** from pricing.price_data group by time_id, outlet_id

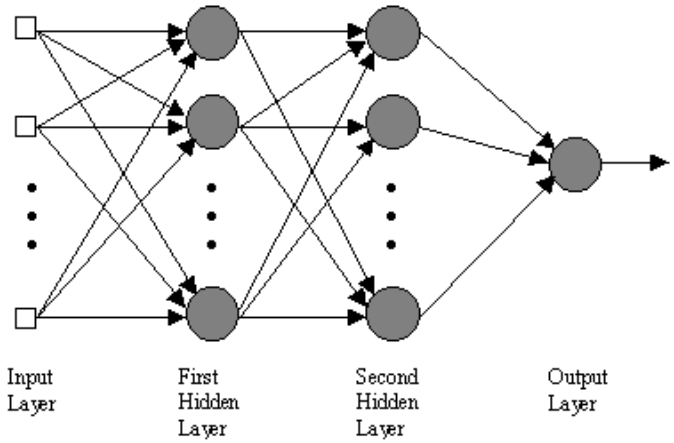
```
CREATE OR REPLACE FUNCTION r_maxfreq(double precision[])
  RETURNS double precision AS
$BODY$
  names(which.max(table(arg1)))
$BODY$
  LANGUAGE 'plr' VOLATILE
  COST 100;
ALTER FUNCTION r_maxfreq(double precision[]) OWNER TO postgres;


CREATE AGGREGATE maxfreq (
  sfunc = plr_array_accum,
  basetype = float8,
  stype = _float8,
  finalfunc = r_maxfreq
);
```

Funkce v PostgreSQL

Neuronové sítě jako modelovací nástroj

- predikce, klasifikace ...
- BPNN (*Back Propagation Neural Network*)
- nelineární závislosti
- vybavování sítě jako funkce v C volaná z SQL



Predikce vývoje kurzu CZK/EURO na den dopředu

```
select getBpnKurzy(ARRAY[0.04973,1.2583,6.7817]::float[]);
```

```
CREATE OR REPLACE FUNCTION getbpnkurzy(arr double precision[])
  RETURNS double precision AS
'$libdir/bpn_kurzy.so', 'bpn_kurzy_tanh'
  LANGUAGE c STABLE STRICT
  COST 1;
ALTER FUNCTION getbpnkurzy(double precision[]) OWNER TO postgres;
```

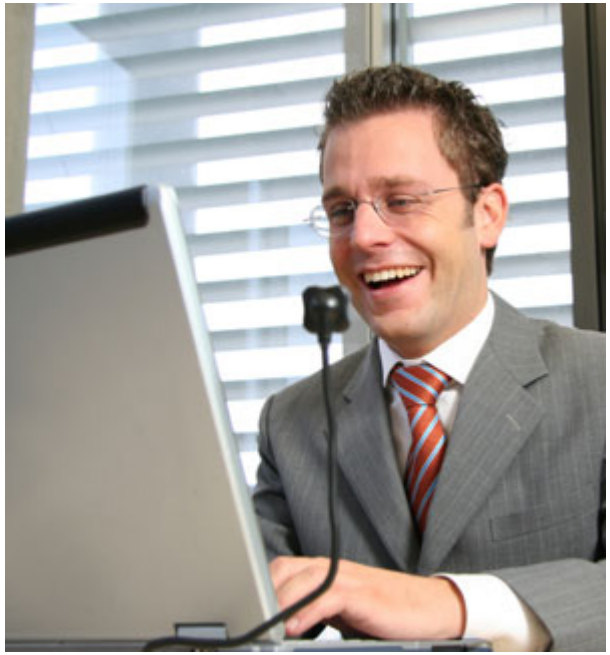
PostgreSQL a business

Náš zákazník požaduje

- řešení
- podporu
- funkční systém, který řeší jeho problémy
- minimalizovat dodatečné požadavky na jeho IT
- adekvátní cenu k výslednému efektu

Náš zákazník nepožaduje

- silnou značku
- technické složitosti
- **Nechce nic mít společného s IT**



Kde vidíme prostor pro PostgreSQL

- střední firmy
- oddělení velkých firem
- specifické zařízení
- vysoký výkon vs nízké rozpočty
- clustery
- SaaS

PostgreSQL vs MSSQL vs MySQL

MSSQL

- vynikající DB
- propracované nástroje
- velká obliba firem
- silná značka
- odstraněný limit na RAM
- Business Intelligence v ceně
- licence na CPU
- partitions až od verze Enterprise
- MS Windows only

VS

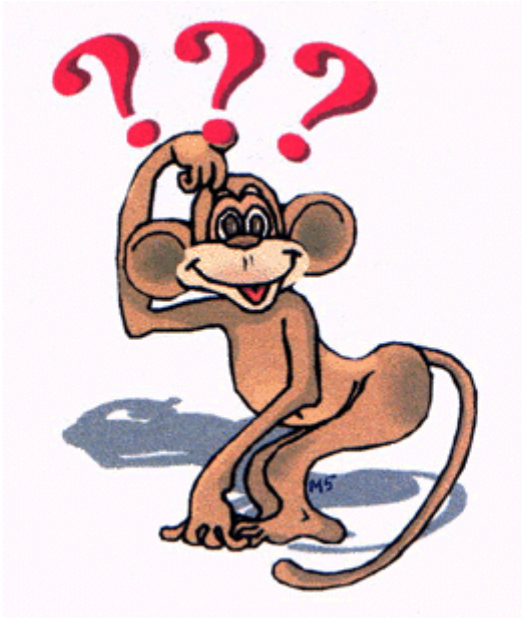
MySQL

- masivně rošířená DB
- “kde kdo ji umí”
- Oracle ?

PostgreSQL

- vynikající DB
- GridSQL, pg-pool
- EnterpriseDB support
- ohebná a rozšiřitelná (podpora GPU apd)
- platformě nezávislá

Otázky



Děkuji za pozornost

Vratislav Beneš
benes@optisolutions.cz