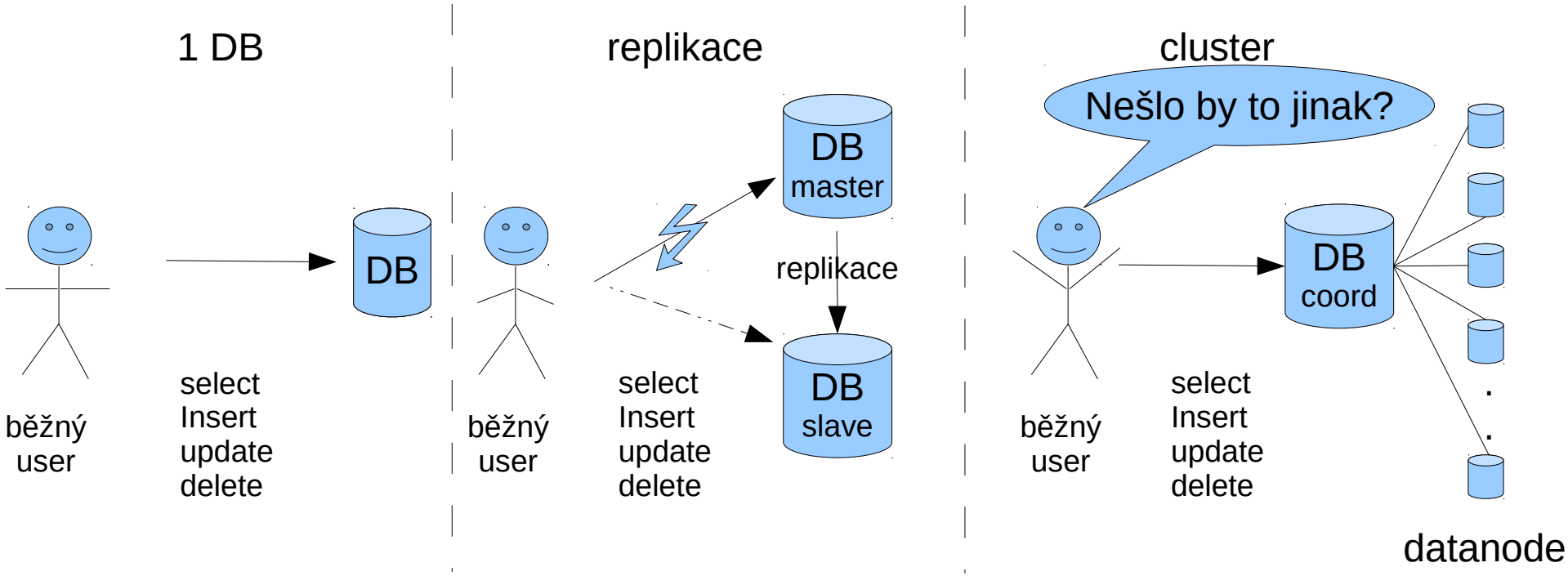
Nestrukturovaná a senzorová data

Vratislav Beneš
benes@optisolutions.cz

1. **Distribuovaný systém jako DB pro analytickou aplikaci**
2. **Nestrukturovaná data**
3. **Senzorová data**

Proč vymýšlet kolo?

- potřeba mít PG 9.4
- umět řídit distribuci dat
- umět řídit způsob, kde a jak je spuštěný dotaz
- zbytnost transakcí napříč datovými nody
- paralelní load dat



Možnosti

- Postgres-XC, Postgres-XL
- MongoDB
- Něco si vyrobit
  - PostgreSQL
  - Python
  - Linux

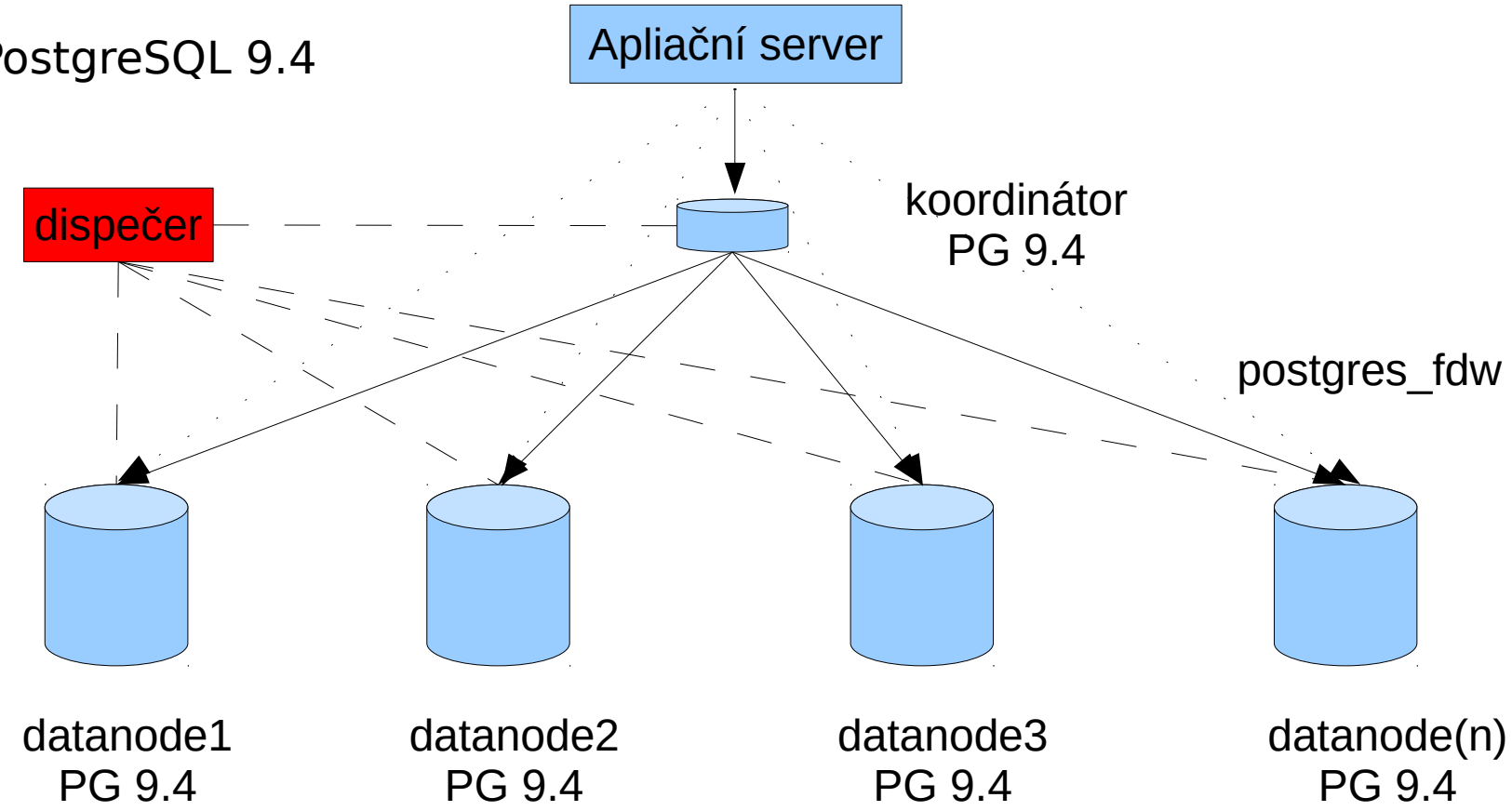
	PostgreSQL	MongoDB
strukturovaná data	TBL/COL	collections / doc
nestrukturovaná data	TBL/COL (Hstore, JSON/JSONB)	collections / doc
jazyk	SQL	JS | aplikace
paralelní dotaz	cluster	přes chunks | aplikace pomocí frameworku

moje vlastní pozorování			
	úložiště	1	~ 2x vyšší než PG
	rychlost agregace		
	DB <RAM – první hledání	1	1
	DB <RAM – opakované hledání	1	~1.5x
	DB >RAM	1	~0.85

Distribuovaný systém

Proč?

- rozložení zátěže
- paralelní data
- zvýšení kapacity
- std. PostgreSQL 9.4



Nestrukturovaná data

Kde se berou?

- aktivní síťové prvky
- aktivní prvky průmyslových/domácích automatizací
- textová komunikace (e-mail, chat, sociální sítě...)
- mobilní zařízení – internet věcí
- kamery

Kam je ukládat?

- noSQL DB
- distribuované systémy
- relační DB

Nestrukturovaná data – doménové znalosti

Co musíme vědět o zkoumaných datech?

- Co v datech hledáme?
- Jak nalézt požadovanou informaci?
- Jak informace měřit

syslog

- standard pro logování systémových událostí
- facelity, priority, messages

2015-01-01 00:01:02	9	5	casa01syslog	finished 0anacron
2015-01-01 00:01:02	9	5	casa01syslog	starting mcelog.cron
2015-01-01 00:01:02	9	5	casa01syslog	finished mcelog.cron
2015-01-01 00:01:35	21	6	static-84-242-	access-list outside_access_in permitted tcp out
2015-01-01 00:01:35	21	6	static-84-242-	Built inbound TCP connection 18338139 for outsi
2015-01-01 00:01:36	21	6	static-84-242-	Teardown TCP connection 18338139 for outside:12
2015-01-01 00:01:56	21	6	static-84-242-	Teardown UDP connection 18338128 for outside:50
2015-01-01 00:01:59	21	6	static-84-242-	access-list inside_access_in permitted udp insi
2015-01-01 00:01:59	21	6	static-84-242-	Built dynamic UDP translation from inside:OptiV
2015-01-01 00:01:59	21	6	static-84-242-	Built outbound UDP connection 18338140 for outs
2015-01-01 00:02:00	21	6	static-84-242-	Teardown UDP connection 18338140 for outside:8.

Nestrukturovaná data – dobytí informace

Jak z nich dostat informace/znalosti?

- otrocká práce
- strojové učení
  - fulltext, regulární výrazy … analýza přirozeného jazyka (NL), počítačové vidění (CV)

```
select id, facility, devicereportedtime,message, syslogtag
from systemevents
where facility < 90 -- and fromhost = 'casa02'
 and devicereportedtime between '2015-01-29 00:00:00' and '2015-02-01 01:00:00'
and to_tsvector(message) @@ to_tsquery('english', 'error');
```

```
--IP

(regexp_matches(message, '[[:digit:]]+.[[:digit:]]+.[[:digit:]]+.[[:digit:]]+')::text[])[1] as ip
```

Nestrukturovaná data – skladování nalezených informací

Konverze text to JSON vhodná pro analýzu

- atribut
- hodnota atributu
- metrika

Nestrukturovaná data – skladování nalezených informací

```
CREATE OR REPLACE FUNCTION getjson_facility10(text)
  RETURNS jsonb AS
$BODY$
DECLARE
  regex text;
  json_key text[5];
  json_val text[5];
begin

  json_key[0] := 'port';
  json_val[0] := split_part((regexp_matches($1, '(?:port)[\s]\w+')::text[])[1],' ',2);

  json_key[1] := 'direction';
  json_val[1] := (regexp_matches($1, 'by|from')::text[])[1];

  json_key[2] := 'ip_address';
  json_val[2] := (regexp_matches($1, '[[:digit:]]+.[[:digit:]]+.[[:digit:]]+.[[:digit:]]+')::text[])[1];

  json_key[3] := 'incident';
  json_val[3] := (regexp_matches($1, '(?i)failed password')::text[])[1];

  json_key[4] := 'user';
  json_val[4] := split_part((replace((regexp_matches('password check failed for user (root)',
                                     '(?:user)[\s]\(\w+')::text[])[1],'(',''')),' ',2);

  return json_object(json_key,json_val);
end;
$BODY$
  LANGUAGE plpgsql IMMUTABLE STRICT
  COST 100;
```

Nestrukturovaná data – dotazování

JSON pro analýzu

- atribut - key
- hodnota atributu - value
- metrika – výskyt/suma hodnot jevu

```
select id, message,events_json, events_json->'port', events_json->'ip_address'
from systemevents
where facility = 10


select events_json->>'ip_address', events_json->>'direction', count(id)
from systemevents
where facility = 10 and events_json->>'direction'='from'
--and devicereportedtime between '2015-02-01 00:00:00' and '2015-02-01 01:00:00'
group by events_json->>'ip_address',events_json->>'direction'



select id, facility, devicereportedtime,message, syslogtag
from systemevents
where facility < 90 -- and fromhost = 'casa02'
 and devicereportedtime between '2015-01-21 00:00:00' and '2015-02-01 01:00:00'
and to_tsvector(message) @@ to_tsquery('english', 'error');
```

Senzorová data

Co to je?

- fyzikální veličiny
  - automatizaca (průmyslová, domácí)
  - vědecké experimenty (např CERN)
  - vesmír
- finanční ukazatele
- *stavové hodnoty*
  - *kurzy sázek, CRM – příležitosti*

Jak na ně?

- plochá tabulka
- diferenciální tabulky
- streamové DB

	s1	s2	s3
t0	30	50	32
t1	30	51	32
t2	30	51	32
t3	31	52	34
t4	31	52	36
t5	30	53	37
t6	30	53	37
t7	29	54	38
t8	28	54	39
t9	26	55	40
t10	27	56	41

s1 – teoplota na vstupu
s2 – teplota uvnitř zařízení
s3 – teplota na výstupu

Senzorová data – primnitivní řešení

```
CREATE TABLE snp_f0
(
  snp_ptr text NOT NULL,
  t_from timestamp without time zone NOT NULL,
  t_to timestamp without time zone NOT NULL,
  val integer,
  CONSTRAINT pk_snp_f0 PRIMARY KEY (snp_ptr, t_from, t_to)
);

CREATE TABLE snp_f1
(
  snp_ptr text NOT NULL,
  t_from timestamp without time zone NOT NULL,
  t_to timestamp without time zone NOT NULL,
  val integer,
  CONSTRAINT pk_snp_f1 PRIMARY KEY (snp_ptr, t_from, t_to)
);

CREATE TABLE snp_f2
(
  snp_ptr text NOT NULL,
  t_from timestamp without time zone NOT NULL,
  t_to timestamp without time zone NOT NULL,
  val integer,
  CONSTRAINT pk_snp_f2 PRIMARY KEY (snp_ptr, t_from, t_to)
);
```

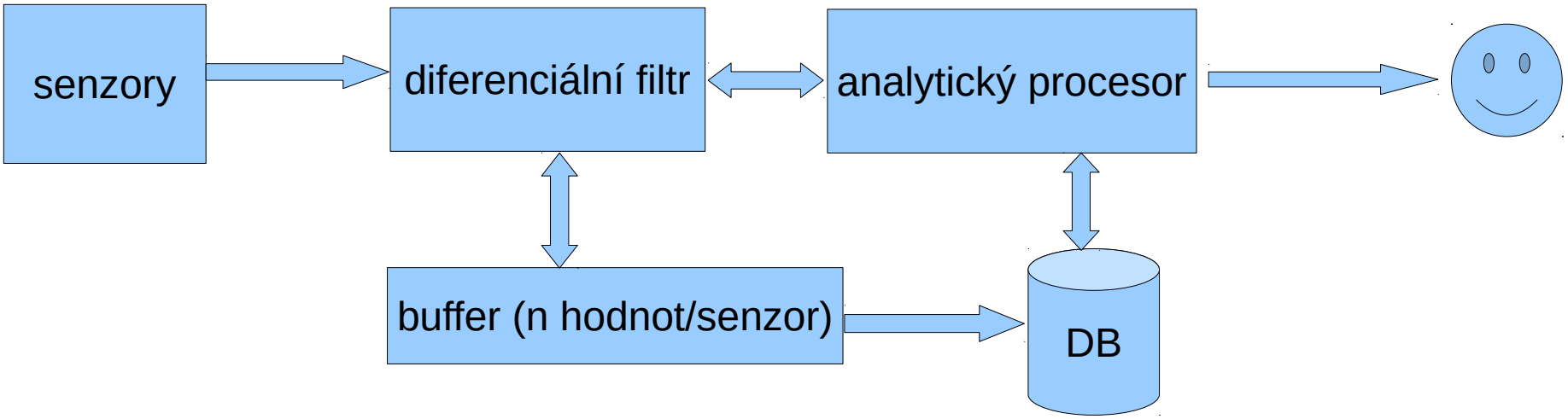
Senzorová data – primnitivní řešení

```
select t, stock_id, product_id, f0.val,f1.val,f2.val
from stock_snp ss
»       inner join snp_f0 f0 on ss.snp_ptr= f0.snp_ptr
»       inner join snp_f1 f1 on ss.snp_ptr= f1.snp_ptr
»       inner join snp_f2 f2 on ss.snp_ptr= f2.snp_ptr
,(select unnest(array['2014-01-01'::timestamp,
                      '2014-01-02'::timestamp,
                      '2014-01-03'::timestamp]) as t) x
where
»       x.t between f0.t_from
»       and f0.t_to and x.t between f1.t_from
»       and f1.t_to and x.t between f2.t_from and f2.t_to
 order by t,stock_id, product_id
```

Senzorová data - zpracování

Co potřebujeme?

- diferenciální filter řízený na základě analýzy
- buffer pro n posledních hodnot
- analytický procesor
- úložiště (DB)



Děkuji za pozornost

Vratislav Beneš
benes@optisolutions.cz