

Postgres 9.5最新特性

李元佳 2015/04

DTCC

2015中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2015

大数据技术探索和价值发现



Postgres 9.5最新特性

李元佳

galylee@gmail.com



目录

- **概览**
- 性能特性
- 管理特性
- SQL特性
- 待提交的特性

自我介绍

面向高性能

基础设施
(并行执行、逻辑复制)

多样性

JSON
外部表

企业级特性

流复制
同步复制

- Logical Decoding for Scalability
- JSONB Data Type
- JSONB

- Indexing
- Expanded JSON functions
 - Delayed Application of Replication

- 3x Faster GIN indexes
- Support for Linux Huge Pages

v9.4

- **pg_prewarm**
- **ALTER SYSTEM**
- **Concurrently updatable Materialized Views**
- **Mongo FDW & MySQL FDW**

- 64 bit LOBs up to 4TB in size

- Custom background workers

- Writable Foreign Data Wrappers

v9.3

- **Materialized Views**

- Cascaded streaming replication

- JSON support, Range Types

v9.2

- **MySQL Foreign Data Wrappers for SQL/MED**

- Synchronous replication
- Serializable Snapshot Isolation

- In-memory (unlogged) tables
- Writeable Common Table Expressions (WITH)

v9.1

- **Index-only scans (covering indexes)**
- **Linear read scalability to 64 cores**

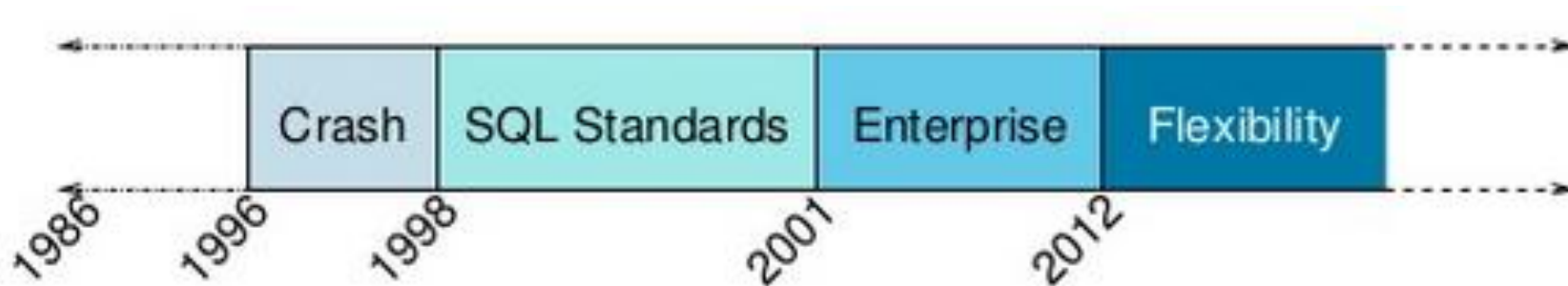
- Deferrable unique constraints and Exclusion constraints
- Streaming replication

- Windows 64 bit Support
- Hot standby

v9.0

- **No restore In-place version upgrades**

Postgres的演进脉络





- 高性能：多核扩展、大数据处理
- 高扩展：逻辑多主复制、PG-XC集群

驱动力1 : Paralle Query

Specific Opportunities

Parallel opportunities include:

- [Sorting](#) 
- Tablespaces
- Partitions
- Foreign tables
- Multi-table access
- Joins (e.g. nested loop), CTEs
- Sequential scans on 1GB segment files
- Per-Page visibility checks and tuple filtering
- [I/O and row processing](#) 
- Aggregates
- Data import/export
- COPY (to reduce the CPU overhead of parsing)
- Index builds
- Constraint checking
- Expensive functions, e.g. PostGIS

Basic Facilities

- Dynamic Background Workers (done in 9.4)
- Dynamic Shared Memory (done in 9.4)

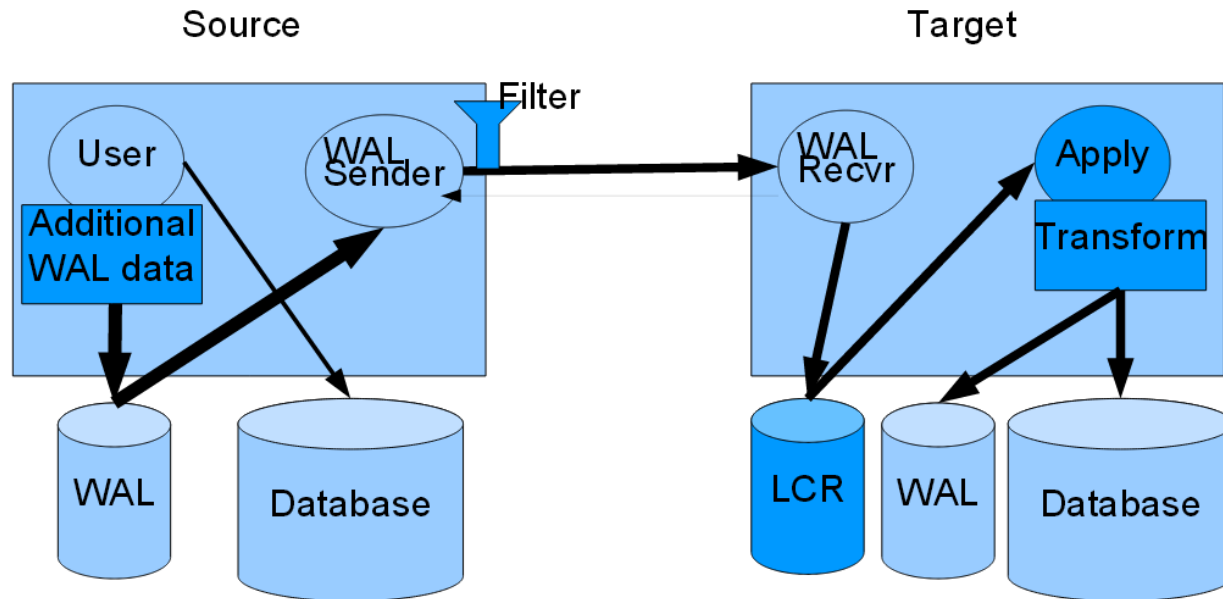
Plumbing

- DSM Table of Contents (done in 9.4)
 - I just mapped this dynamic shared memory segment; how do I figure out what it contains?
- Message Queueing (done in 9.4)
 - How does a background worker send tuples, errors, notices, etc. to a user backend?
- Error Propagation (working on it)
 - Common infrastructure to make using message queueing easy.
- Shared Memory Allocator (early draft posted)
- Shared Hash Table (someday)






驱动力2 : Logical Replication



Logical Streaming Replication



Logical Replication里程碑

	Postgres Core	Extension	Variant Distro
9.4 – now!	+Logical Decoding +Dynamic Shm +Replication Slots +REPLICA IDENTITY	UDR Extension	Full BDR – up to 48 nodes
9.5	+Commit Timestamps +Sequence AM API +Replication Identifiers +DDL Event Triggers	UDR Extension (Faster!) 	Slim BDR
9.6	+Cluster Metadata +UDR Plugin +Replication Sets +Global Sequences 	(Gone!) 	Multi-master BDR
9.7	+Multi-Master +New conflict models 		(Gone!) 

9.5 版本的开发历程

- June 10, 2014 – branch 9.4
- June 2014 – CF1 - Completed
- August 2014 – CF2 - Completed
- October 2014 – CF3 - Completed
- December 2014 – CF4 - Completed
- February 2015 – CF5 – In Progress
- First Beta for April
- GA for September(?)

代码的统计

- Statistics
 - 2397 files changed
 - 222383 insertions (+)
 - 135826 deletions(-)
- Almost double that of 9.4!

9.5 -> 更加容易管理维护

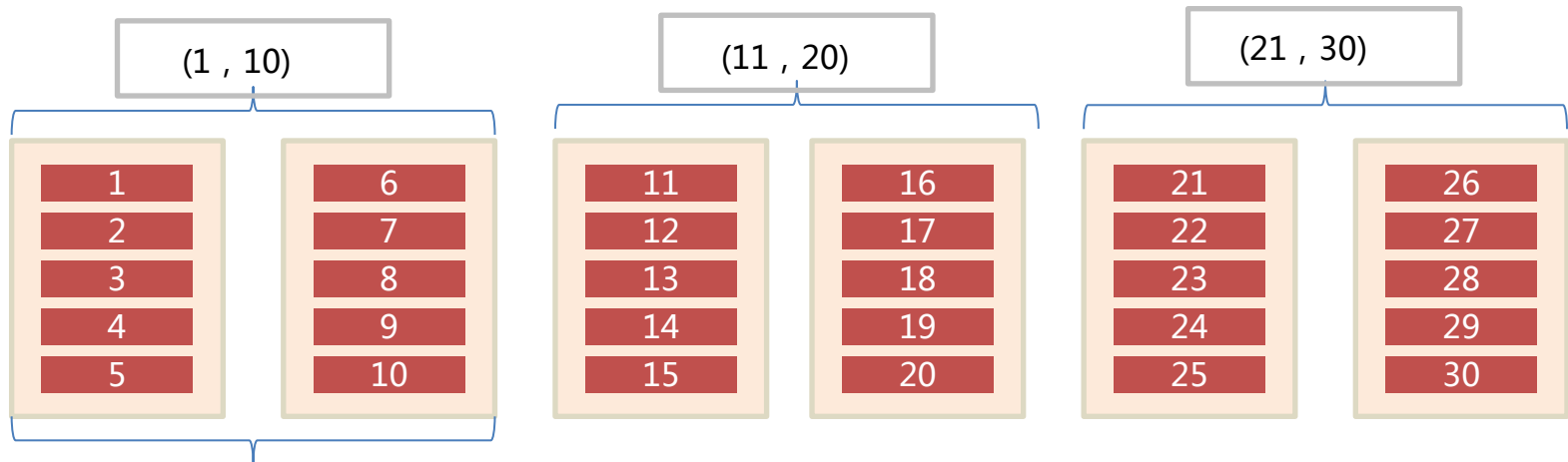
- BRIN Indexes
- pg_rewind
- Row level security
- Import Foreign Schema
- Min and Max wal size
- Set UNLOGGED

目录

- 概览
- **性能特性**
- 管理特性
- SQL特性
- 待提交的特性

BRIN Indexes块范围索引

- BRIN(Block Range Index)：保存数据块的值的摘要信息，如存储某一组块里面所有记录中的最大最小值，与Exadata的Storage Index相似
- 全表扫描之前，先从范围索引过滤掉不满足条件的数据块，可大大提高全表扫描的性能
- 对于按顺序排列的表效果尤为明显



Block Range 块的数量可以自己定义

BRIN 的用法

```
=# CREATE TABLE brin_example AS SELECT generate_series(1,100000000) AS id;
SELECT 100000000
=# CREATE INDEX btree_index ON brin_example(id);
CREATE INDEX
Time: 239033.974 ms
=# CREATE INDEX brin_index ON brin_example USING brin(id);
CREATE INDEX
Time: 42538.188 ms
=# \d brin_example
Table "public.brin_example"
  Column |  Type  | Modifiers
-----+-----+-----
   id    | integer |
Indexes:
    "brin_index" brin (id)
    "btree_index" btree (id)
```

BRIN 的大小

```
=# CREATE INDEX brin_index_64 ON brin_example USING brin(id)
    WITH (pages_per_range = 64);
CREATE INDEX
=# CREATE INDEX brin_index_256 ON brin_example USING brin(id)
    WITH (pages_per_range = 256);
CREATE INDEX
=# CREATE INDEX brin_index_512 ON brin_example USING brin(id)
    WITH (pages_per_range = 512);
CREATE INDEX
```

Having a look at the relation sizes, BRIN indexes are largely smaller in size.

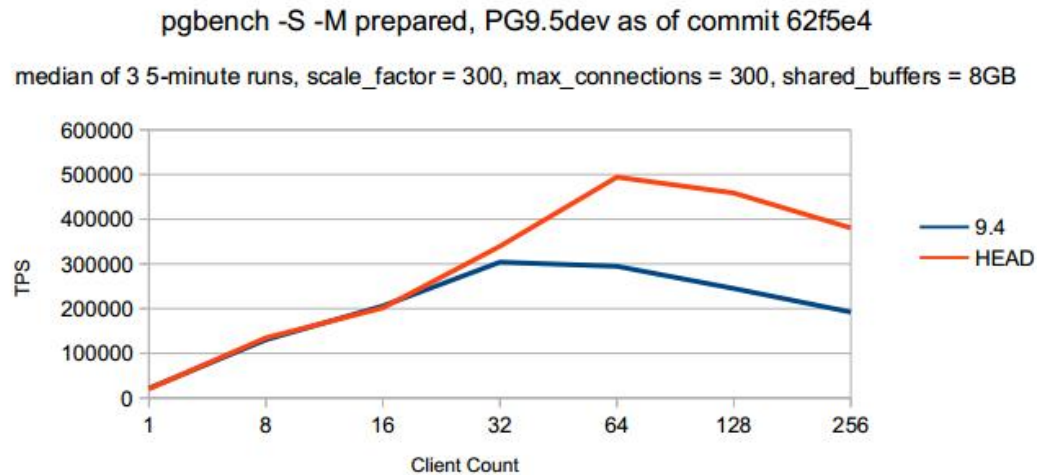
```
=# SELECT relname, pg_size_pretty(pg_relation_size(oid))
    FROM pg_class WHERE relname LIKE 'brin_%' OR
        relname = 'btree_index' ORDER BY relname;
```

relname	pg_size_pretty
brin_example	3457 MB
brin_index	104 kB
brin_index_256	64 kB
brin_index_512	40 kB
brin_index_64	192 kB
btree_index	2142 MB

(6 rows)

读的多核扩展性提高

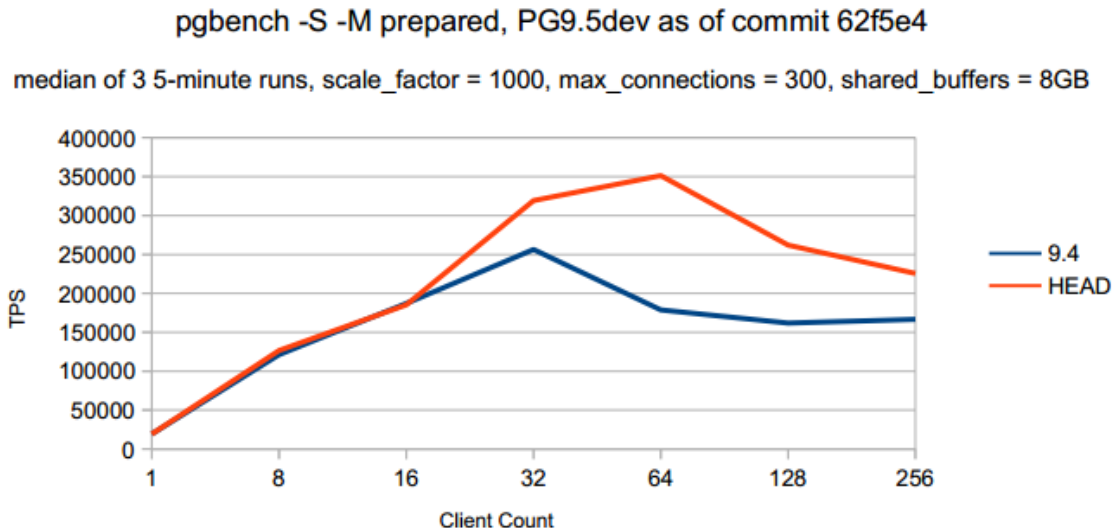
Read Scalability – Data fits in shared_buffers



IBM POWER-8 having 24 cores, 192 hardware threads, 492GB RAM

读的多核扩展性提高

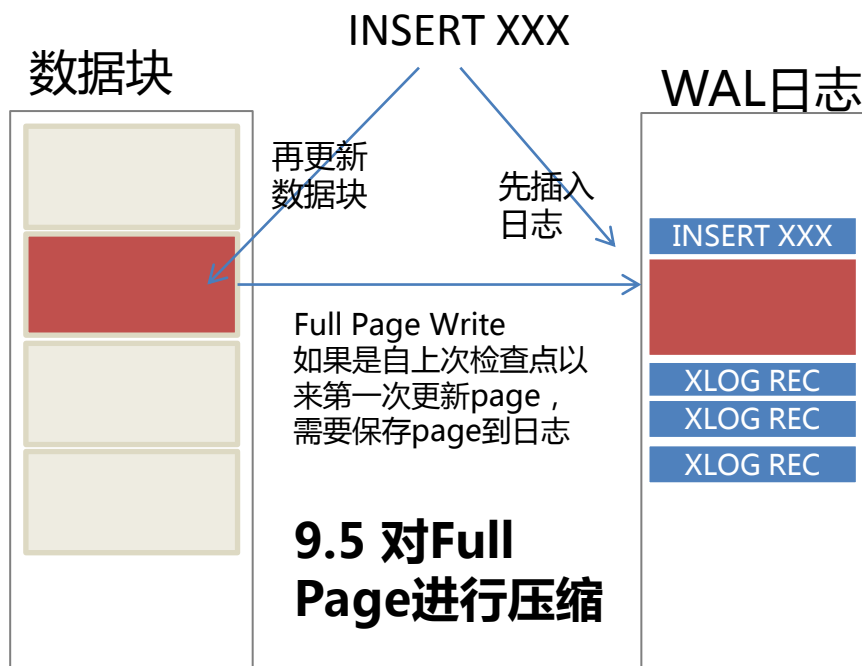
Read Scalability – Data fits in RAM



IBM POWER-8 having 24 cores, 192 hardware threads, 492GB RAM

WAL日志压缩

- 写日志时候对数据块进行压缩
- 往磁盘写的少，复制的传输的量也会少，更新的性能提高
- CPU的消耗量会上升



`wal_compression (boolean)`

When this parameter is on, the PostgreSQL server will decompress the compressed page image

Turning this parameter on can reduce the WAL the compression during WAL logging and on the

Case	WAL generated	User CPU	Syst
UUID tab, compressed	633 MB	30.64	1.89
UUID tab, not compressed	727 MB	17.05	0.51
int tab, compressed	545 MB	20.90	0.68
int tab, not compressed	727 MB	14.54	0.84

文字排序的性能改善

- Improve sort of text Datum
 - Use first bytes of strxfrm
 - Fallback to old method if 8-first bytes identical
- Performance
 - CREATE INDEX time 3x faster
 - Slower for 8-first bytes identical

- ```
create table stuff as select random()::text as a, 'filler filler filler'::text
as b, g as c from generate_series(1, 1000000) g;
SELECT 1000000
```
- ```
create index on stuff (a);  
CREATE INDEX
```

并行查询 - 9.6

Parallel Seq scan

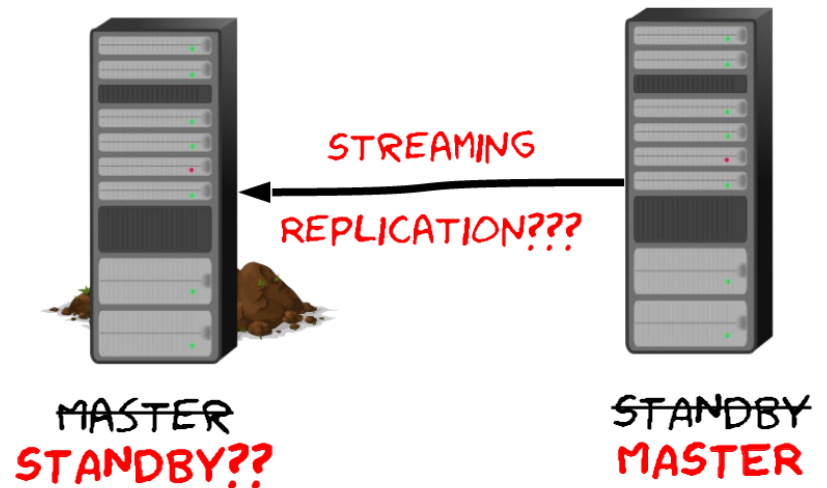
<div>Edit</div> <div>Comment ▾</div> <div>Status ▾</div>	
Title	Parallel Seq scan
Topic	Server Features
Created	2015-01-12 12:13:35
Last modified	2015-01-16 13:36:46 (2 months, 3 weeks ago)
Latest email	2015-04-09 10:07:10 (5 days ago)
Status	2015-02: <div>Needs review</div> 2014-12: <div>Returned with Feedback</div>
Authors	Amit Kapila (amitkapila)
Reviewers	

目录

- 概览
- 性能特性
- **管理特性**
- SQL特性
- 待提交的特性

pg_rewind

- Rebuild from scratch
 - Erase old master, take new base backup from new master, and copy it
- rsync
 - Reads all data from disk
- pg_rewind
 - Only reads and copies data that was changed



min and max wal size

- checkpoint_segments removed!
- Instead, control min and max size
 - min_wal_size (default 80MB)
 - max_wal_size (default 1GB)
- Checkpoints auto-tuned to happen in between
- Space only consumed when actually needed

cluster_name

- Set custom string in process title
- Identify instances on same host

```
35839 ?? Ss 0:00.04 postgres: pg_5432: checkpointer process
35873 ?? Ss 0:00.02 postgres: pg_5433: checkpointer process
35909 ?? Ss 0:00.02 postgres: pg_5434: checkpointer process
35956 ?? Ss 0:00.03 postgres: pg_5435: checkpointer process
36001 ?? Ss 0:00.02 postgres: pg_5436: checkpointer process
36044 ?? Ss 0:00.02 postgres: pg_5437: checkpointer process
36095 ?? Ss 0:00.03 postgres: pg_5438: checkpointer process
36223 ?? Ss 0:00.00 postgres: pg_5439: checkpointer process
36353 ?? Ss 0:00.00 postgres: pg_5440: checkpointer process
```

Parallel vacuumdb

- vacuumdb can use concurrent connections
- Add -j<n> to command line
- Speed up important VACUUM or ANALYZE
- Mind the load!

Logging of replication commands

- Control with GUC `log_replication_commands`
- Default = off
 - Log messages as `DEBUG1`
 - Same as prior releases
- When on, written as `LOG`

```
LOG: received replication command: IDENTIFY_SYSTEM  
LOG: received replication command: START_REPLICATION  
0/3000000 TIMELINE 1
```

目录

- 概览
- 性能特性
- 管理特性
- **SQL特性**
- 待提交的特性

Row level security

- 可以控制记录级别的访问权限
- 需要对表进行策略定义

(CREATE | ALTER | DROP Policy)

- 需要对每张表启用行安全策略

ALTER TABLE .. ENABLE/DISABLE ROW
SECURITY.

Row level security例子-1

```
$ select * from clients;
```

```
id | account_name | account_manager
```

```
----+-----+-----
```

```
1 | initrode    | peter
```

```
2 | initech     | bill
```

```
3 | chotchkie's | joanna
```

```
(3 rows)
```

Row level security例子-2

create policy just_own_clients on clients

for all

to public

using (account_manager = current_user);

CREATE POLICY

alter table clients ENABLE ROW LEVEL SECURITY;

ALTER TABLE

Row level security例子-3

```
$ select * from clients;
```

```
id | account_name | account_manager
```

```
----+-----+-----
```

```
1 | initrode    | peter
```

```
(1 row)
```

```
$ \c - joanna
```

```
$ select * from clients;
```

```
id | account_name | account_manager
```

```
----+-----+-----
```

```
3 | chotchkie's | joanna
```

```
(1 row)
```


Reindex Schema

- Reindex all relations on a single schema
- Including toast relations
- Good for maintenance of multi-schema environments

```
=# REINDEX SCHEMA pg_catalog;  
NOTICE: 00000: table "pg_catalog.pg_class" was reindexed  
LOCATION: ReindexObject, indexcmds.c:1942  
[...]  
NOTICE: 00000: table "pg_catalog.pg_depend" was reindexed  
LOCATION: ReindexObject, indexcmds.c:1942  
REINDEX
```

Import Foreign Schema

```
postgres=# CREATE SCHEMA remoteschema;  
CREATE SCHEMA  
postgres=# IMPORT FOREIGN SCHEMA testschema FROM SERVER otherserver INTO remoteschema;  
IMPORT FOREIGN SCHEMA  
postgres=# \det remoteschema.*  
      List of foreign tables  
 Schema | Table | Server  
-----+-----+-----  
 remoteschema | test2 | otherserver  
 remoteschema | test3 | otherserver  
(1 row)
```

Foreign Table Inheritance

- 外部表支持继承特性

```
=# CREATE TABLE log_entries(log_time timestamp, entry text);  
CREATE TABLE  
=# CREATE FOREIGN TABLE log_entry_y2014_f (log_time timestamp,  
                                             entry text)  
    INHERITS (log_entries) SERVER server_5433 OPTIONS (table_name 'log_entry_y2014');  
CREATE FOREIGN TABLE  
=# CREATE FOREIGN TABLE log_entry_y2015_f (log_time timestamp,  
                                             entry text)  
    INHERITS (log_entries) SERVER server_5434 OPTIONS (table_name 'log_entry_y2015');  
CREATE FOREIGN TABLE
```

Set LOGGED/UNLOGGED

- Unlogged table (9.1~)
 - Permanent table storage
 - Not WAL'ed, not persistent on crash
- Change persistence of table
 - LOGGED / UNLOGGED switch
 - Complete table rewrite
 - Unlogged -> logged faster than logged -> unlogged

```
postgres=# ALTER TABLE a SET UNLOGGED;
```

```
ALTER TABLE
```

```
postgres=# ALTER TABLE a SET LOGGED;
```

```
ALTER TABLE
```

Skip Locked for row-level lock

- New layer for FOR [SHARE | UPDATE | ...]
 - Default, wait for lock
 - NOWAIT, error instead of waiting
 - SKIP LOCKED, skip locked tuples
- View of data not consistent
- Reduction of lock contention
- For queue-like tables

```
postgres=# SELECT * FROM a FOR UPDATE NOWAIT;  
ERROR:  could not obtain lock on row in relation "a"  
postgres=# SELECT * FROM a FOR UPDATE SKIP LOCKED;  
 a  | b  | c  
----+---+---  
  2 |  2 |  2  
  3 |  3 |  3
```

Multi columns update

- Nice with functions!
- Simplify UPDATE ... SET ... FROM ()

```
CREATE TABLE table1 (a int, b int, c int);  
CREATE TABLE table2 (a int, b int, c int);  
UPDATE table1 AS t1  
SET b = t2.b, c = t2.c  
  FROM (SELECT a, b, c FROM table2) AS  
t2 WHERE t1.a = t2.a;
```



```
UPDATE table1  
SET (b, c) = (SELECT b, c FROM table2  
              WHERE table2.a = table1.a);
```

generate_series (numeric)

- Now for numeric
- Already here for:
 - Timestamps
 - Integers

```
=# SELECT generate_series(-4.9, 5.5, 2.2);  
generate_series  
-----  
      -4.9  
      -2.7  
      -0.5  
       1.7  
       3.9  
(5 rows)
```

目录

- 概览
- 性能特性
- 管理特性
- SQL特性
- **待提交的特性**

pgaudit

read	Commands that read database objects (SELECT)
write	DML commands that modify database objects (e.g. INSERT)
privilege	DCL commands that are related to access privileges (e.g. GRANT/REVOKE)
user	DDL commands that are related to database users (e.g. CREATE/DROP/ALTER ROLE)
definition	User-level DDL commands (e.g. CREATE TABLE)
config	Administrator-level commands that change the database configuration (e.g. CREATE LANGUAGE, CREATE OPERATOR CLASS)
admin	Administrator-level commands that are not configuration related (e.g. CLUSTER, VACUUM, REINDEX)
function	Function execution

```
LOG:  [AUDIT],2014-04-30 17:13:55.202854+09,auditdb,ianb,ianb,DEFINITION,CREATE TABLE,TAB
LOG:  [AUDIT],2014-04-30 17:14:06.548923+09,auditdb,ianb,ianb,WRITE,INSERT,TABLE,public.>
LOG:  [AUDIT],2014-04-30 17:14:21.221879+09,auditdb,ianb,ianb,READ,SELECT,TABLE,public.x,
LOG:  [AUDIT],2014-04-30 17:15:25.620213+09,auditdb,ianb,ianb,READ,SELECT,VIEW,public.v_>
LOG:  [AUDIT],2014-04-30 17:15:25.620262+09,auditdb,ianb,ianb,READ,SELECT,TABLE,public.x,
LOG:  [AUDIT],2014-04-30 17:16:00.849868+09,auditdb,ianb,ianb,WRITE,UPDATE,TABLE,public.>
LOG:  [AUDIT],2014-04-30 17:16:18.291452+09,auditdb,ianb,ianb,ADMIN,VACUUM,,VACUUM x;
LOG:  [AUDIT],2014-04-30 17:18:01.08291+09,auditdb,ianb,ianb,DEFINITION,CREATE FUNCTION,f
```

pgaudit

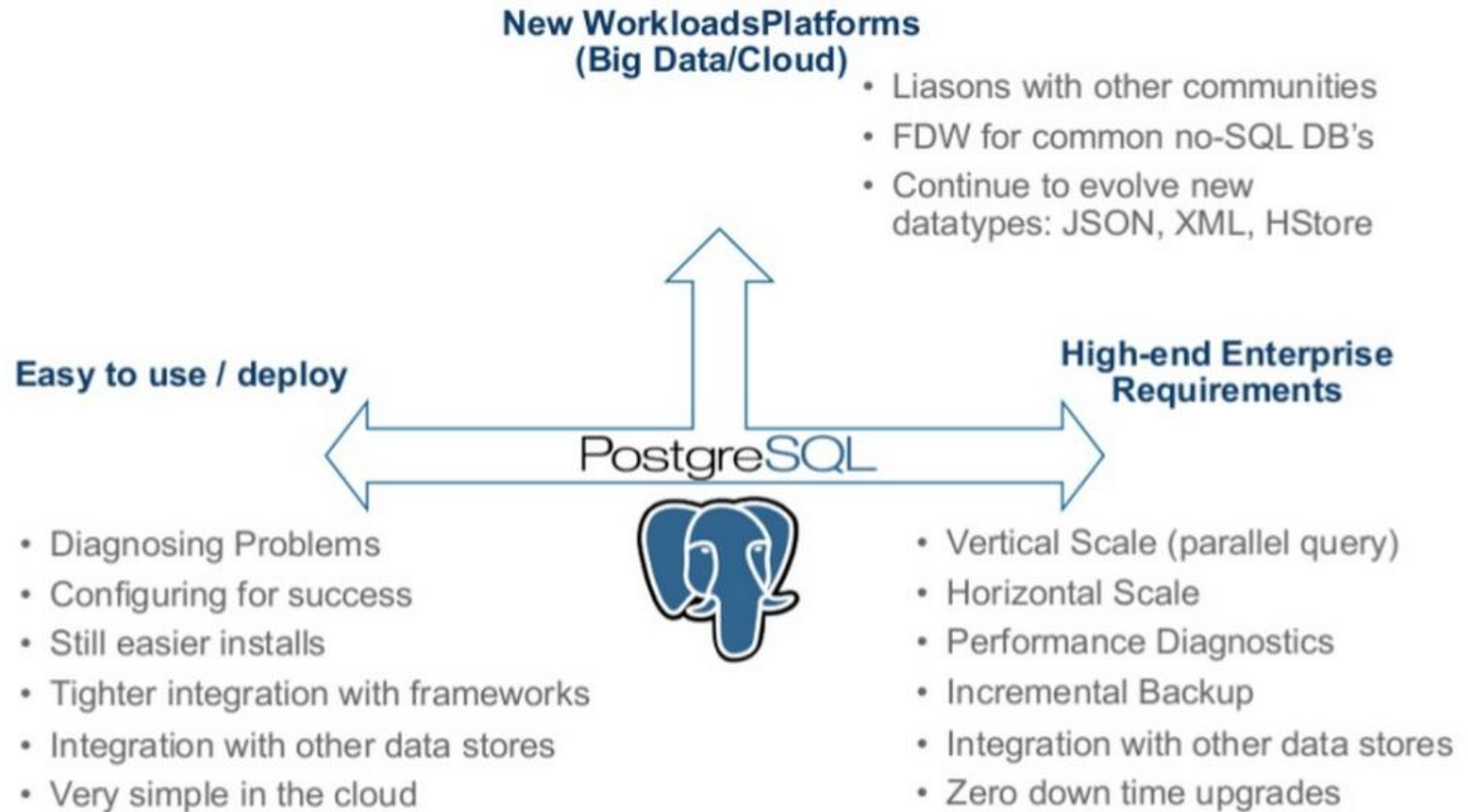
- Audit capabilities:
 - At user level
 - At table level
- Spec still unsure
 - AUDIT in-core clause
 - Simple contrib module (?)
- Feature has been asked for years

UPSERT

- If fully-baked, number #1 for 9.5
- Grammar:
 - Agreement (!)
 - ON CONFLICT [UPDATE | IGNORE]
- Still development work going on...

INSERT ... ON CONFLICT {UPDATE | IGNORE}

未来发展方向



鸣谢 Thanks

- **Michael Paquier**
 - <http://en.pgconf.ru/static/presentations/2015/paquier.pdf>
- **Magnus Hagander**
 - <http://www.hagander.net/talks/postgresql95.pdf>
- **Amit Kapila**
 - Exciting Features In PostgreSQL 9.5



THANKS