



PostgreSQL: Presente y Futuro

Álvaro Herrera
PostgreSQL developer – 2ndQuadrant Inc.
pgDay Santiago 2019

PostgreSQL histórico



PostgreSQL Anniversary Summit, Toronto, Canadá, 2006

PostgreSQL moderno

Increasing regression test coverage based on GCOV analysis	Committed	stable	movead li (movead.li)		michael-kun	1	2019-09-04 04:11	2019-09-04 04:36	<input type="checkbox"/> Author <input type="checkbox"/> Reviewer
Performance									
Evaluate immutable functions during planning (in FROM clause)	Committed		Alexander Kuzmenkov (akuzmenkov), Aleksandr Parfenov (a.parfenov)	Antonin Houska (a.houska), Anastasia Lubennikova (lubennikovaav), Anthony Bykov (anthonyb)	tgl	7	2019-08-01 22:51	2019-09-24 15:17	<input type="checkbox"/> Author <input type="checkbox"/> Reviewer
Optimization of some jsonb functions	Committed	13	Nikita Glukhov (n.glukhov)	Andrew Dunstan (adunstan)	alvherre	3	2019-09-21 00:13	2019-09-21 00:09	<input type="checkbox"/> Author <input type="checkbox"/> Reviewer
Reduce overhead of NOTIFY with many databases	Committed		Martijn van Oosterhout (martijnvoosterhout)	Tom Lane (tgl)	tgl	2	2019-09-22 15:46	2019-09-22 15:48	<input type="checkbox"/> Author <input type="checkbox"/> Reviewer
Speedup truncates of relation forks	Committed		Kirk Jamison (kirkjamison)	Fujii Masao (masaofujii)	masaofujii	2	2019-09-24 08:42	2019-09-24 23:57	<input type="checkbox"/> Author <input type="checkbox"/> Reviewer
Use PageIndexTupleOverwrite() within nbtsort.c	Committed		Peter Geoghegan (pgeoghegan)	Anastasia Lubennikova (lubennikovaav)	pgeoghegan	1	2019-08-13 19:09	2019-08-13 19:00	<input type="checkbox"/> Author <input type="checkbox"/> Reviewer
Allow simplehash to use already-calculated hash values	Committed		Jeff Davis (jdavis)		jdavis	1	2019-08-08 22:48	2019-07-17 19:59	<input type="checkbox"/> Author <input type="checkbox"/> Reviewer

84 min ago	Michael Paquier	Fix handling of pg_class.relispartition at swap phase...	master	github/master	commit commitdiff tree snapshot
8 hours ago	Tom Lane	Allow extracting fields from a ROW() expression in...			commit commitdiff tree snapshot
9 hours ago	Tom Lane	On Windows, use COMSPEC to find the location of cmd...			commit commitdiff tree snapshot
11 hours ago	Tom Lane	Handle empty-string edge cases correctly in strpos().			commit commitdiff tree snapshot
22 hours ago	Michael Paquier	Doc: Add missing step for pg_stat_progress_cluster			commit commitdiff tree snapshot
24 hours ago	Michael Paquier	Fix dependency handling at swap phase of REINDEX CONCUR...			commit commitdiff tree snapshot
46 hours ago	Michael Paquier	Fix initialization of fake LSN for unlogged relations			commit commitdiff tree snapshot
2 days ago	Noah Misch	Fix copy-paste defect in comment.			commit commitdiff tree snapshot
2 days ago	Noah Misch	Update comment about __sync_lock_test_and_set() bug.			commit commitdiff tree snapshot
2 days ago	Tom Lane	Doc: improve documentation of configuration settings...			commit commitdiff tree snapshot

pgsql-hackers@lists.postgresql.org

Nuevas bases de datos

- El término "BD relacional" suena obsoleto
- Las BDs modernas han avanzado mucho desde SQL:89
- Cada año,
 - PostgreSQL agrega nuevas tecnologías, nuevos conceptos
 - Se abren nuevos potenciales casos de uso

Líneas de desarrollo

- Particionamiento
- Extensibilidad de almacenamiento
 - zheap
 - zedstore
- Nuevas características del estándar SQL
 - SQL/JSON
 - Temporal query
- *Foreign data*
- *Sharding*
- Revolución en el ejecutor
 - JIT
 - ejecución en “batch”

Particionamiento

- Particionamiento declarativo agregado en PostgreSQL 10
 - CREATE TABLE PARTITION BY; estrategias range y list; sin PKs, sin FKs
- PostgreSQL 11: mucha funcionalidad extra
 - Significativas mejoras de rendimiento, *pruning*
 - Estrategia *hash*; partición default
 - DDL adicional: CREATE INDEX; PKs y FKs (con restricciones)
- PostgreSQL 12: mejoras de rendimiento
 - mejor escalabilidad a grandes cantidades de particiones
 - FKs
- PostgreSQL 13: mejoras de rendimiento
 - mejor *pruning* en UPDATE/DELETE
 - rehacer parte del optimizador

Extensibilidad de almacenamiento

- “table access method”
- actualmente: heapam → VACUUM
- profundamente ligado al código del servidor
- Agregar abstracción requirió dos años de trabajo
- PostgreSQL 12: CREATE ACCESS METHOD .. TYPE TABLE

zheap

- No+VACUUM
- basado en UNDO
- UPDATE/rollback no generan tuplas muertas
 - UNDO las limpia
- *undo workers*

zedstore

- A partir de “table AM”
- Column stores!
 - Citus: cstore_fdw
 - Greenplum: AOCO (append-optimized column-oriented)
 - PostgresPro: VOPS

<https://www.postgresql.eu/events/pgconfeu2019/schedule/session/2738-zedstore-column-store-for-postgresql/>

Nuevas características en estándar SQL

- PostgreSQL tiene un miembro en el comité de estandarización!

SQL/JSON: JSONPATH

- PostgreSQL 12: JSONPATH

```
select * from jsonb_path_query('{"a": [1,2,3,4,5]}',  
    '$.a[*] ? (@ >= $min && @ <= $max)',  
    '{"min":2, "max":4}');
```

jsonb_path_query

2

3

4

(3 filas)

SQL/JSON: JSON_TABLE

- JSON_TABLE

```
CREATE TEMP TABLE jsonb_table_test (js jsonb);
INSERT INTO jsonb_table_test
VALUES (
    '[
        {"a": 1, "b": [], "c": []},
        {"a": 2, "b": [1, 2, 3], "c": [10, null, 20]},
        {"a": 3, "b": [1, 2], "c": []},
        {"x": "4", "b": [1, 2], "c": 123}
    ]'
);
```

SQL/JSON: JSON_TABLE (2)

```
select
  jt.*
from
  jsonb_table_test jtt,
  json_table (
    jtt.js, 'strict $[*]' as p
    columns (
      n for ordinality,
      a int path 'lax $.a' default -1 on empty,
      nested path 'strict $.b[*]' as pb columns ( b int path '$' ),
      nested path 'strict $.c[*]' as pc columns ( c int path '$' )
    )
  ) jt;
```

n	a	b	c
1	1		
2	2	1	
2	2	2	
2	2	3	
2	2		10

Temporal query

products				
id	name	price	valid_from	valid_til
1	shoe	\$5	Jan 2018	Jan 2019
1	shoe	\$7	Jan 2019	
2	snow	\$2	Jan 2016	

```
ALTER TABLE variants
```

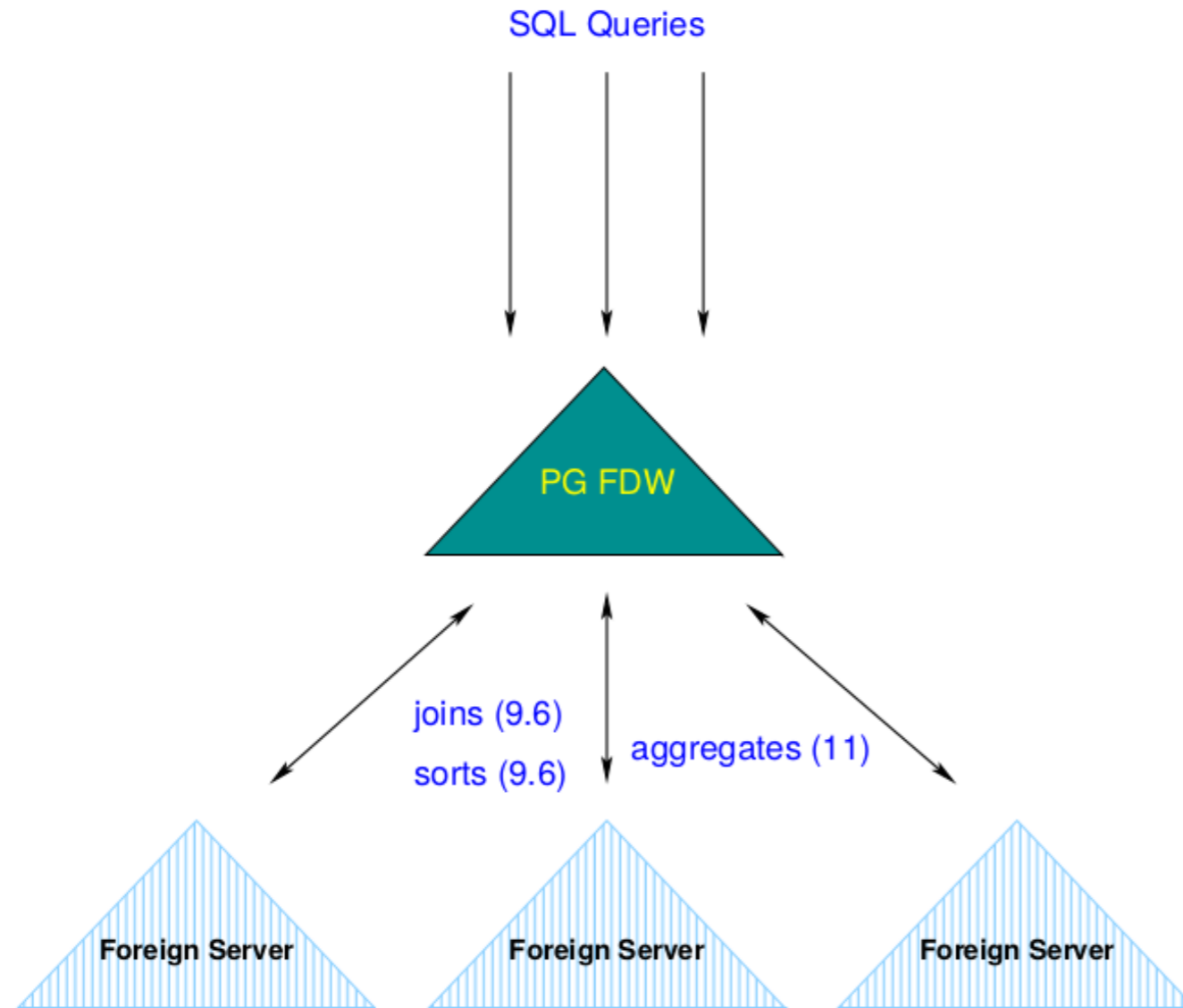
```
ADD CONSTRAINT tfk_variants_product_id
```

```
FOREIGN KEY (product_id, PERIOD valid_at)
```

```
REFERENCES (id, PERIOD valid_at);
```

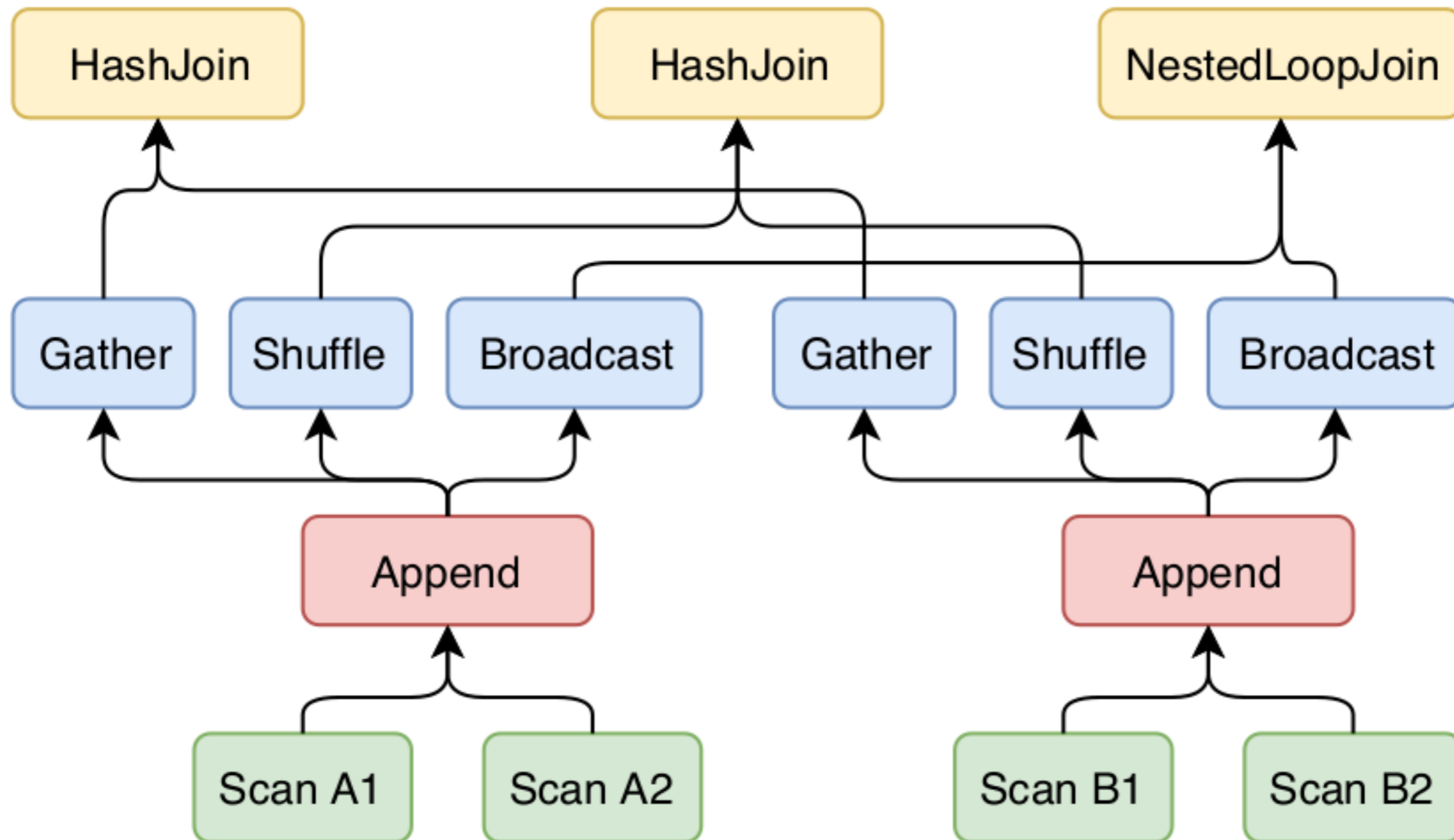
<https://www.pgcon.org/2019/schedule/events/1336.en.html>

Foreign data & sharding



<https://www.postgresql.eu/events/pgconfeu2019/schedule/session/2581-community-roadmap-to-sharding/>

Foreign data & sharding (2)



Revolución en el ejecutor

- El ejecutor estilo “Volcán” es ineficiente
- Para mejorar utilización CPU-caches, paralelismo ...
 - había que rehacerlo con una nueva estructura
 - había que hacer el código más eficiente

Compilación JIT del ejecutor

- Compilación del plan de ejecución a código nativo
- Actualmente:
 - “deformado” de tuplas
 - evaluación de expresiones
 - Mejoras de hasta 30% en ciertas consultas OLAP-BI
 - ... importante, pero no el fin del mundo
- En el futuro:
 - JITizar los nodos del volcán
 - un cache de programas
 - mejorar generación de código

Ejecución en “batch”

- Cambiar estilo del ejecutor
- Permitir que la CPU procese varios registros a la vez
 - Reduce latencia proporcionalmente a cantidad de trabajo paralelo
- crítico con grandes volúmenes de datos
- batch + columnar + JIT: 1000x rendimiento (y más)

¿Preguntas?

¡Gracias por escuchar!