从Uber切换Potgres说起

李元佳
云徙科技联合创始人及CTO

主办单位：杭州 Alibaba Group 阿里巴巴集团 战略合作伙伴：intel

事件的背景

ARCHITECTURE

WHY UBER ENGINEERING SWITCHED FROM POSTGRES TO MYSQL

JULY 26, 2016

BY EVAN KLITZKE

事件的背景

The Architecture of Postgres

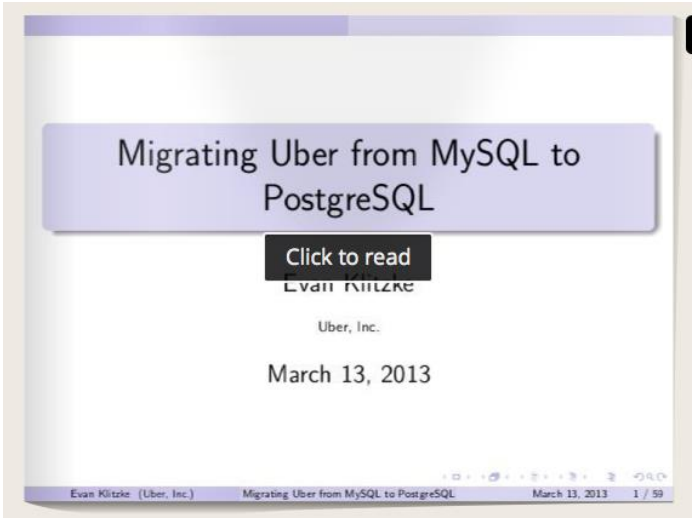
We encountered many Postgres limitations:

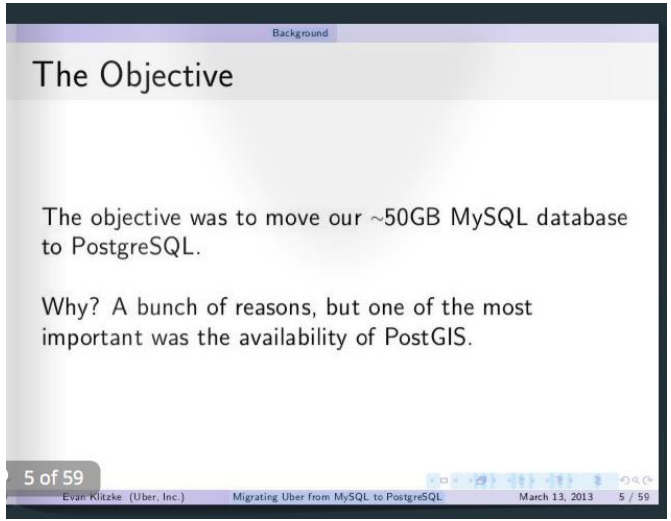
- Inefficient architecture for writes
- Inefficient data replication
- Issues with table corruption
- Poor replica MVCC support
- Difficulty upgrading to newer releases

事件的背景

Postgres served us well in the early days of Uber, but we ran into significant problems scaling Postgres with our growth. Today, we have some legacy Postgres instances, but the bulk of our databases are either **built on top of MySQL (typically using our Schemaless layer)** or, in some specialized cases, NoSQL databases like Cassandra.

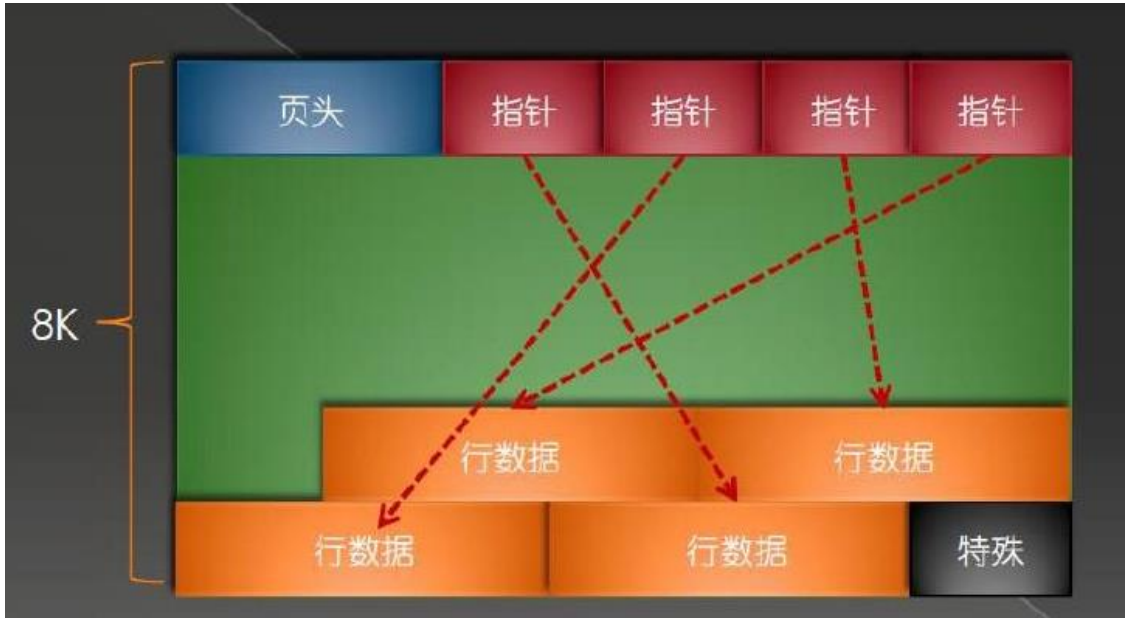
事件的背景



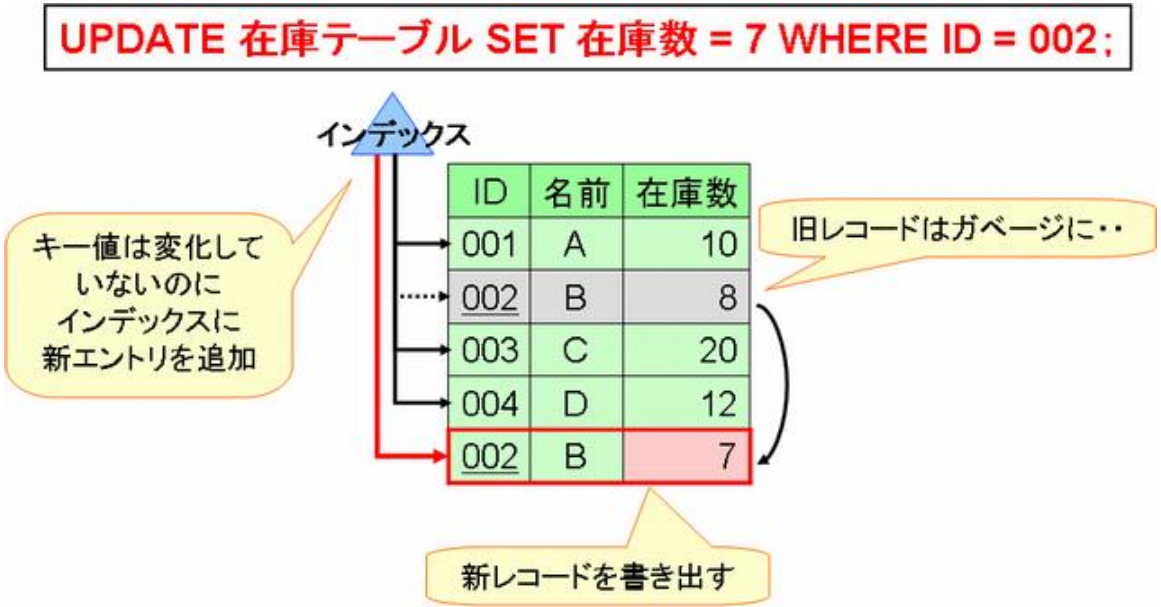


扫码观看大会视频

Postgres数据的存储

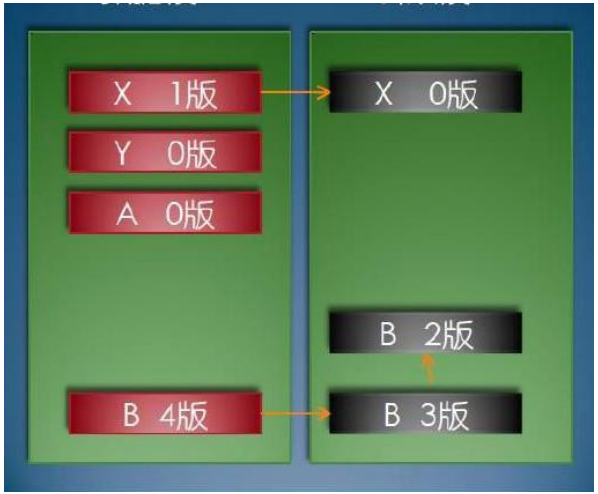


Postgres的数据更新



记录的多版本机制及更新

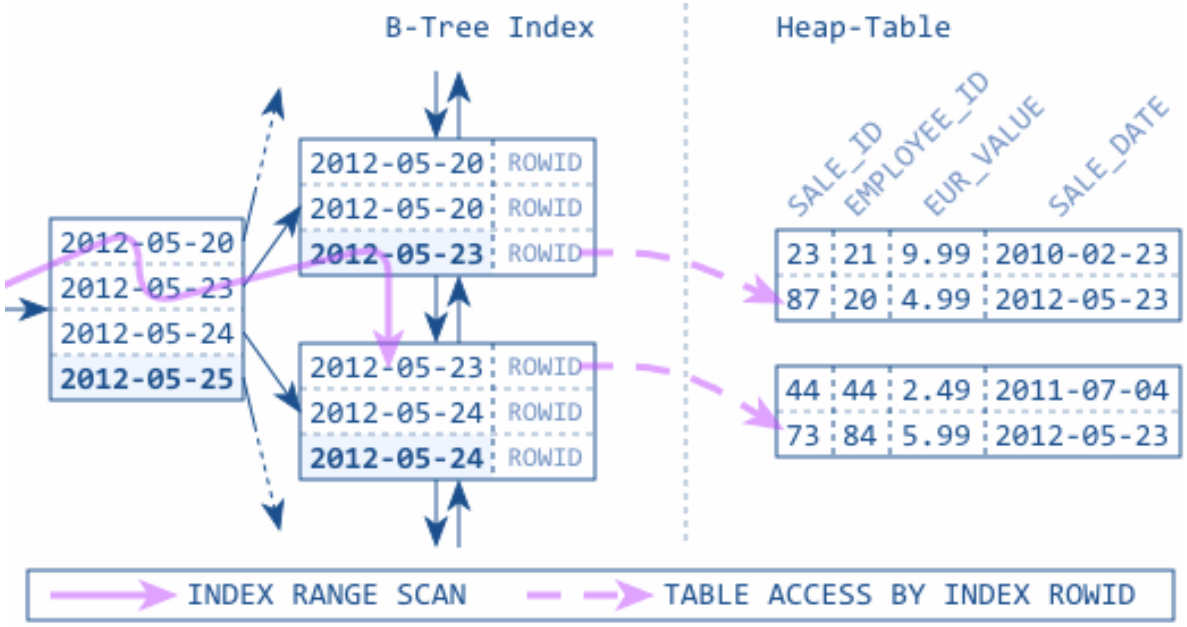
- 旧版本回收和管理问题比较大



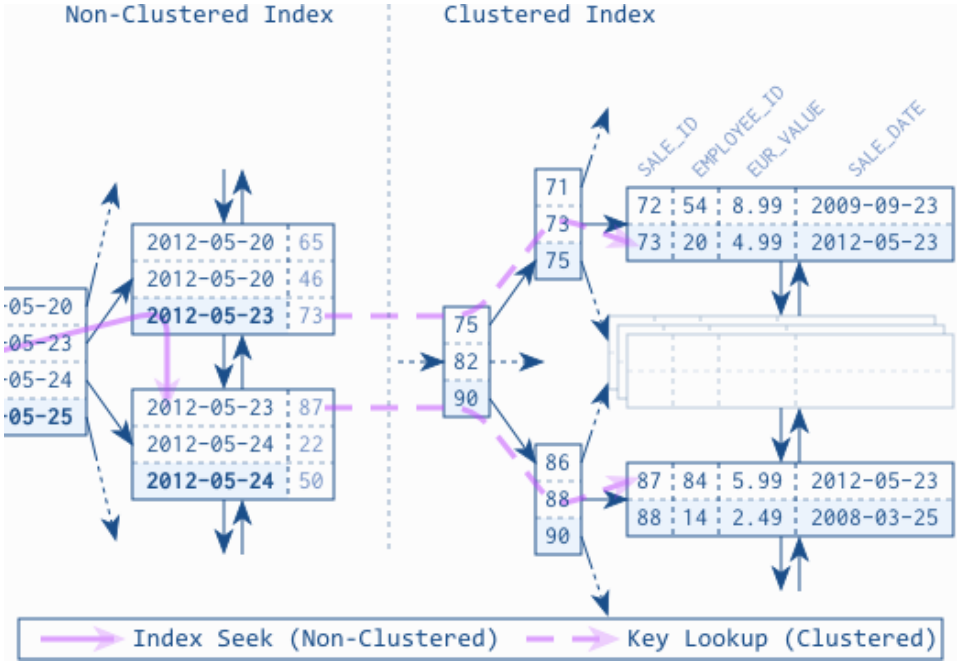
- 写的路径比较长
- 中途需要读旧版本的话，代价比较大

扫码观看大会视频

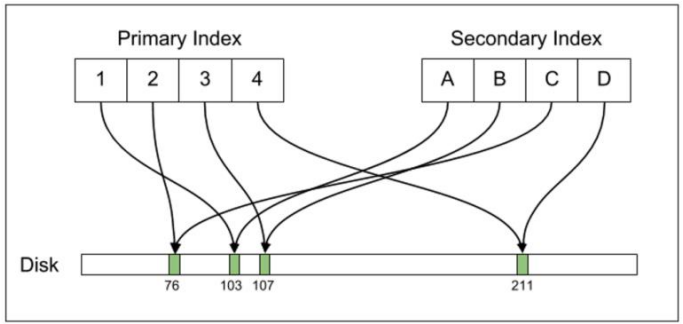
Postgres索引与数据的关系（B-Tree + Heap）



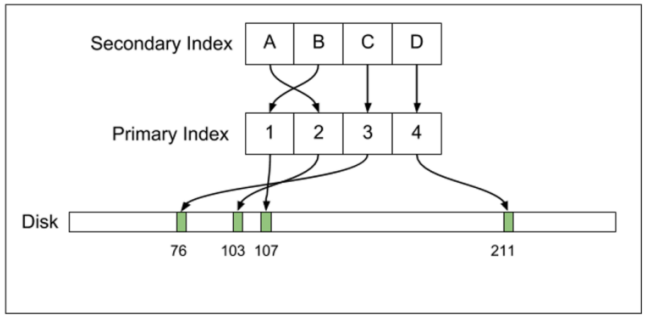
MySQL索引与数据的关系(Clustered Index)



索引结构导致的差异



- 记录物理位置的变更，会导致所有索引的变更



- 二级索引的检索需要进行两次索引检索
- 如果主索引的数据量大的话，比较消耗空间

Uber宣称的写放大问题-表结构

ctid	id	first	last	birth_year
A	1	Blaise	Pascal	1623
B	2	Gottfried	Leibniz	1646
C	3	Emmy	Noether	1882
D	4	Muhammad	al-Khwārizmī	780
E	5	Alan	Turing	1912
F	6	Srinivasa	Ramanujan	1887
G	7	Ada	Lovelace	1815
H	8	Henri	Poincaré	1854

表结构

id	ctid
1	A
2	B
3	C
4	D
5	E
6	F
7	G
8	H

主索引

first	last	ctid
Ada	Lovelace	G
Alan	Turing	E
Blaise	Pascal	A
Emmy	Noether	C
Gottfried	Leibniz	B
Henri	Poincaré	H
Muhammad	al-Khwārizmī	D
Srinivasa	Ramanujan	F

二级索引

birth_year	ctid
780	D
1623	A
1646	B
1815	G
1854	H
1887	F
1882	C
1912	E

二级索引

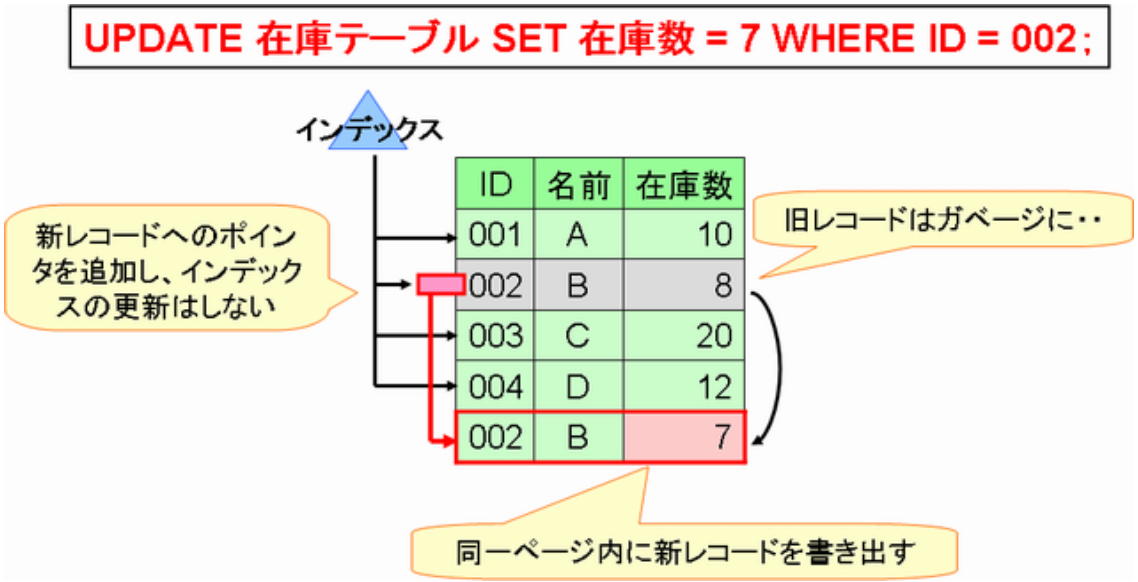
Uber宣称的写放大问题（一次更新、四次写）

- 写数据 Write the new row tuple to the tablespace
- 更新主索引 Update the primary key index to add a record for the new tuple
- 更新二级索引 Update the (first, last) index to add a record for the new tuple
- 更新二级索引 Update the birth_year index to add a record for the new tuple

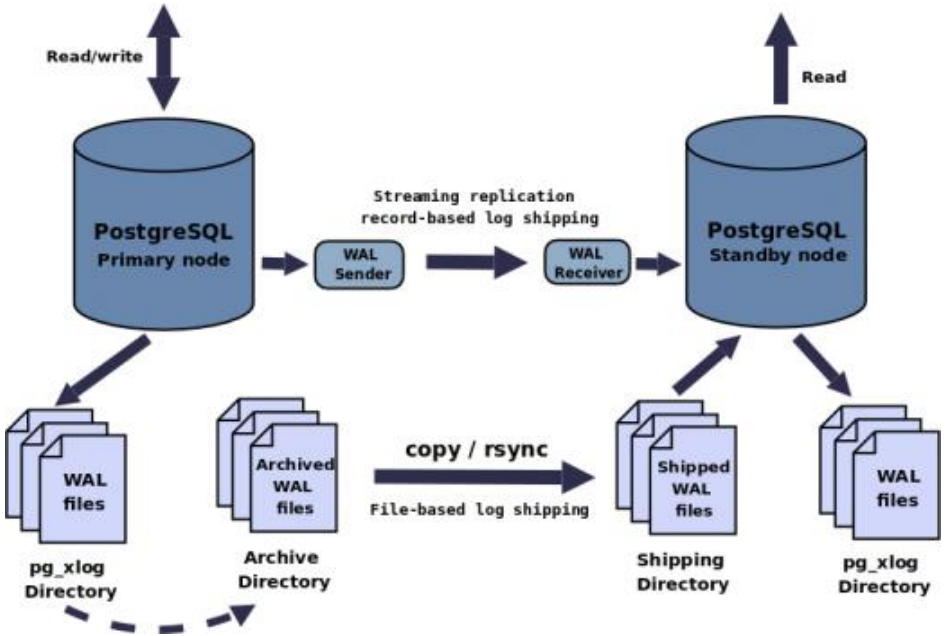
Uber宣称的写放大问题

- 数据的更新需要更新所有索引 PostgreSQL always needs to update all indexes on a table when updating rows in the table. MySQL with InnoDB, on the other hand, needs to update only those indexes that contain updated columns.
- “if we have a table with a dozen indexes defined on it, an update to a field that is only covered by a single index must be propagated into all 12 indexes to reflect the ctid for the new row”.

Postgres的免索引更新机制（HOT更新）



Postgres的流复制



Uber宣称的Postgres的流复制的问题

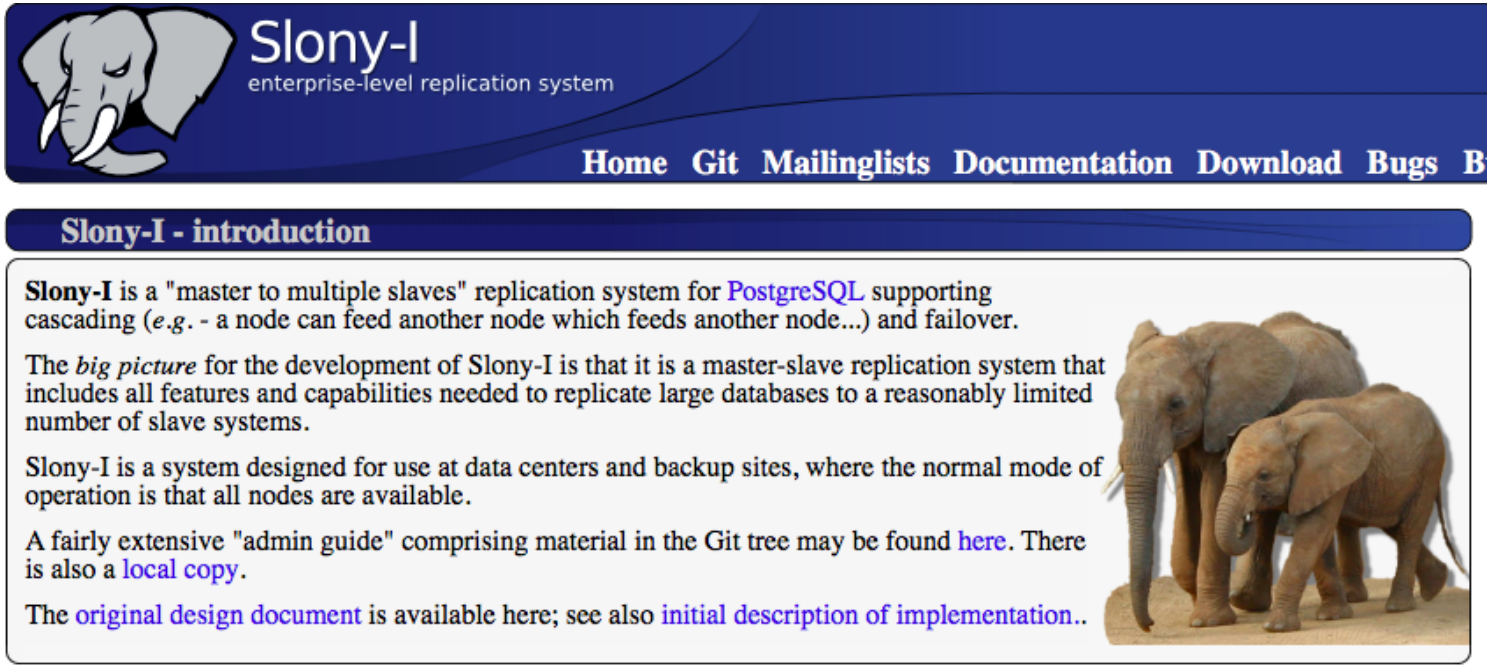
- 写放大：This write amplification issue naturally translates into the replication layer as well because replication occurs at the level of on-disk changes.
- 物理复制带来的潜在数据损坏的危险：During a routine master database promotion to increase database capacity, we ran into a Postgres 9.2 bug.
- 版本的升级问题：Because replication records work at the physical level, it’s not possible to replicate data between different general availability releases of Postgres.
- 对于事务的影响：Postgres does not have true replica MVCC support.

Uber宣称的Postgres的流复制的问题

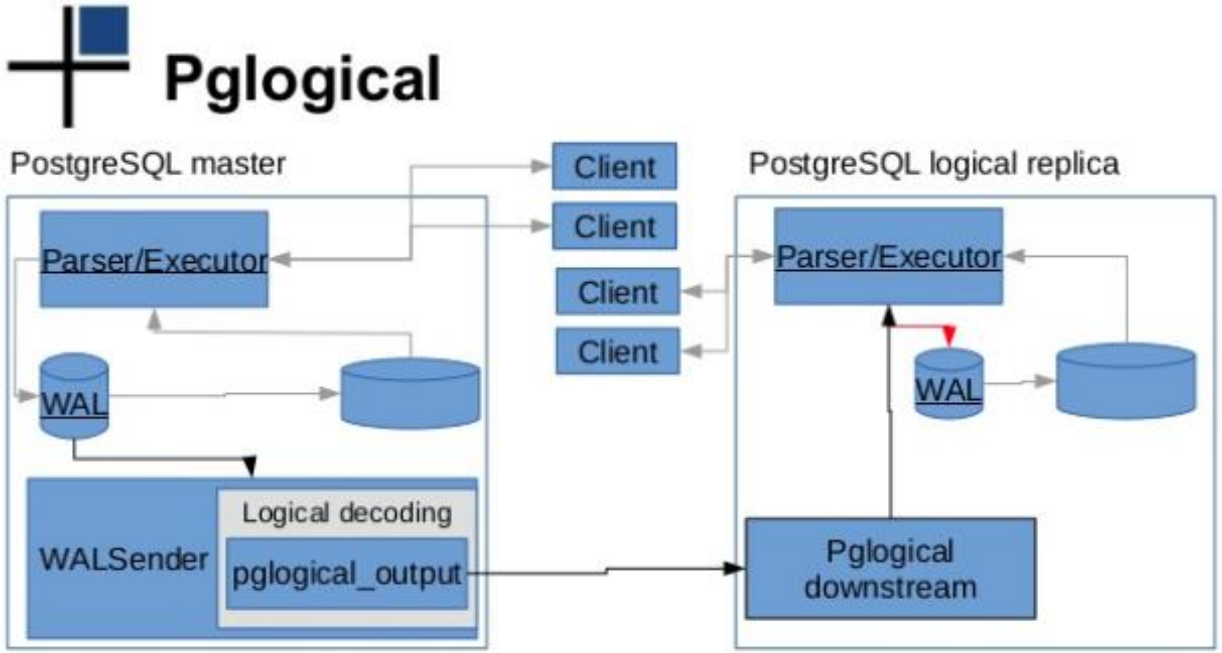
- 写放大：This write amplification issue naturally translates into the replication layer as well because replication occurs at the level of on-disk changes.
- 物理复制带来的潜在数据损坏的危险：During a routine master database promotion that increase database capacity, we ran into a Postgres 9.2 bug.
- 版本的升级问题：Because replication records work at the physical level, it's not possible to replicate data between different general availability releases of Postgres.
- 对于事务的影响：Postgres does not have true replica MVCC support.

逻辑复制 VS 物理复制

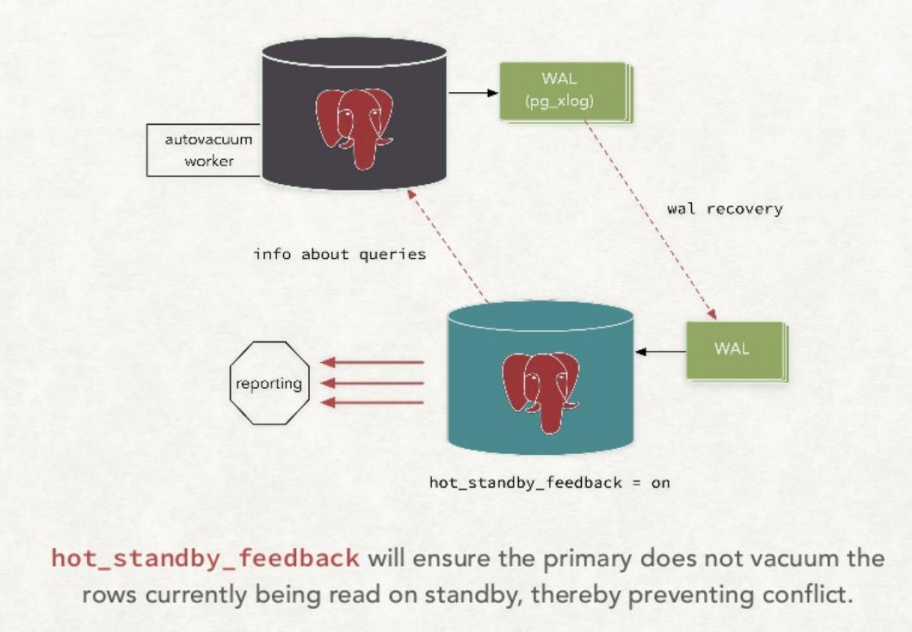
Postgres的逻辑复制解决方案



Postgres的逻辑复制解决方案



Postgres的复制的事务问题解决方案



2016 The Computing Conference

THANKS