事务的历史与SSI

刘智龙

CONTENTS

01. 事务的基础

一些关于事务的基本知识，包括ACID，隔离级别等概念

02. 事务的历史

数据库事务的发展历程以及弱隔离存在的问题

03. SSI理论知识

理解SSI的理论和它解决的问题，并描述写偏序异常现象和不可串行化的问题

04. PostgreSQL中的SSI

展示PostgreSQL数据库中的SSI的行为， 以及它实现了SSI又做了哪些优化

事务的基础

事务的原本含义

事务=>transaction=>交易

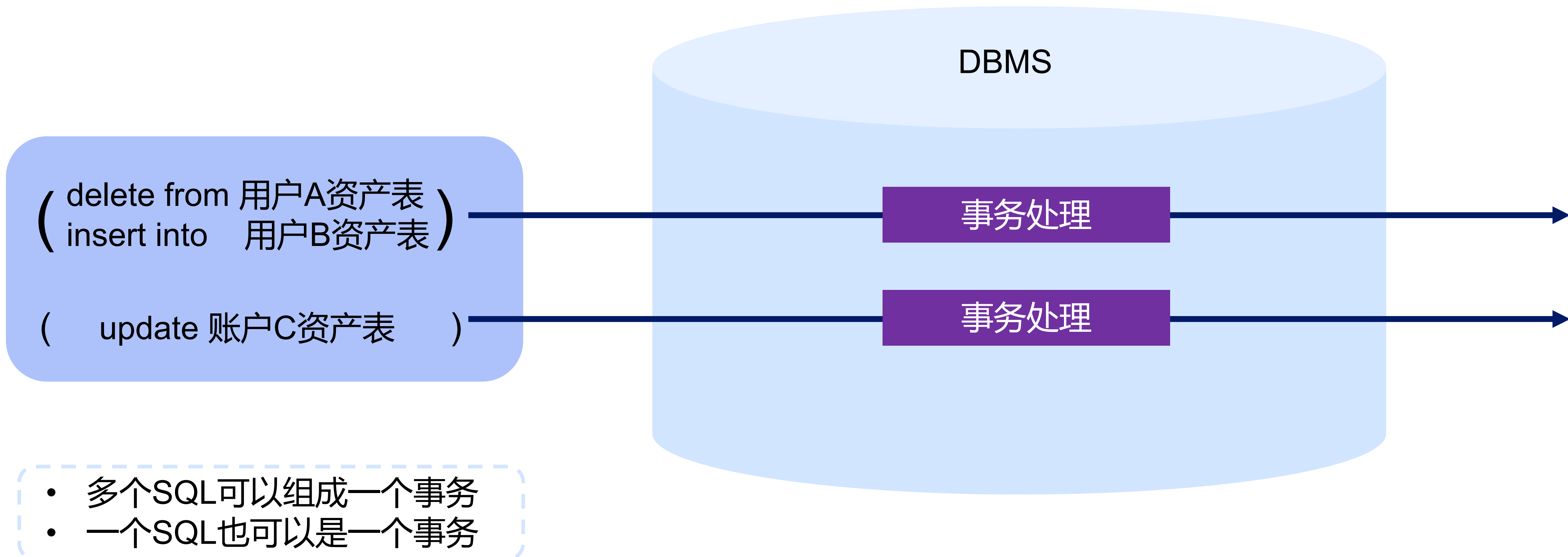
“一手交钱，一手交货”
这是一个交易

交易是事务原本的含义，而我们数据库中的事务就是从交易这个词汇而来。

事务的基础

数据库中的事务

- 事务是关系型数据库的基本工作单元



事务的基础

CHINA POSTGRESQL ASSOCIATION

A

Atomicity
原子性

C

Consistency
一致性

I

Isolation
隔离性

D

Durability
持久性

事务的基础

CHINA POSTGRESQL ASSOCIATION

Atomicity

原子性

事务中的各个操作要么全部完成，要么全部取消.

就像化学中的原子一样不可分割、不可分裂。如果一个事务执行到一半有问题执行失败了，执行不下去了，那么这个事务必须全部回退，就像刚才那个案例一样，一个事务中有2条sql，一条跑完了，另一条失败了，那么这个执行成功的sql也必须回退，因为事务必须具备原子性。

事务的基础

CHINA POSTGRESQL ASSOCIATION

Consistency
一致性

事务完成时，所有数据都保持一致状态.

这个定义其实比较模糊。事务一般会操作数据，数据库中数据的状态会被更新，因为事务操作，数据会从一个状态到另一个状态，这个状态必须是合理合法的，数据逻辑必须和真实世界的逻辑一致。这里可能比较抽象，打个比方：比如说甲有100元，乙有200元，他们合起来的钱是300元，这时乙向甲转账100元，那么甲就有200元，乙有100元，他们合起来的钱仍然是300元。这个虚拟世界的数据变化应该与真实世界的逻辑保持一致。

事务的基础

CHINA POSTGRESQL ASSOCIATION

Isolation
隔离性

多个事务并发执行的结果必须和分开单独执行的结果一致.

比如2个事务，一个接着一个的串行执行，必须和并行执行结果一致。这个好像容易理解没什么问题，我们按住不表，先记住这个逻辑.

事务的基础

Durability
持久性

事务处理结束后，对数据的修改就是永久的.

如果更新后数据放入内存中，关机就没了对吧，那么他理应放入磁盘里，那么放入磁盘是安全的吗？如果磁盘损坏了呢？我们可以有高可用架构写多份数据，再延伸一下还可以有地域级别的容灾，那如果我们再杠一下，多个地域都挂了呢？如果从架构上来介绍，这个问题好像没有答案。但是从用户的角度来看，其实比较容易理解。比如用户在存钱的时候，他把钞票放进去了，那么账户上应该显示他存入的钱的数字，这个数字对于用户来说是永久的，用户认为即使天塌下来，他的账户里都应该有这个数字，这是持久性的含义。

事务的基础

我们再来到另一个著名的概念，1992年美国国家标准ANSI SQL-92标准中定义了**4种隔离级别**和**3种异常现象**。

虽然现在的数据库行业基本都参照ISO国际标准了。

但这个92年的标准对数据库行业影响非常大，相信很多玩数据库小伙伴的都知道4隔离级别。

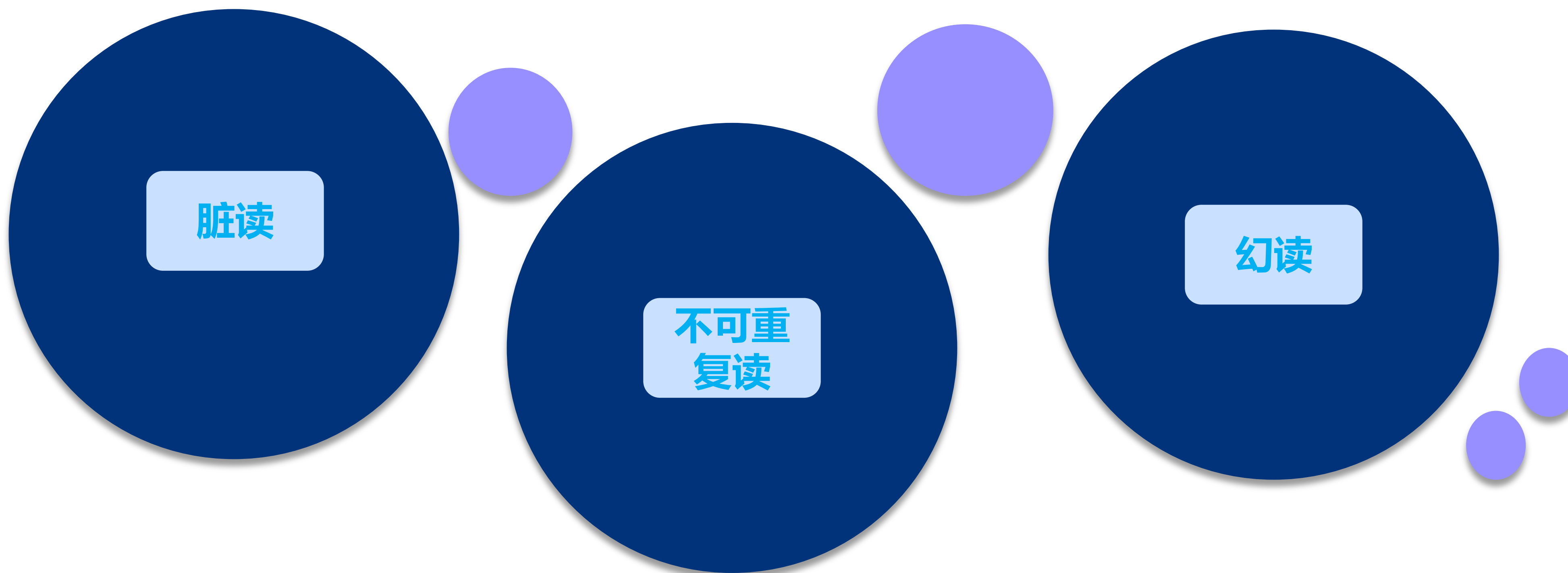
事务的基础

CHINA POSTGRESQL ASSOCIATION

ANSI SQL-92标准的**4种隔离级别**

事务的基础

ANSI SQL-92标准的**三种异常现象**



*92标准中定义了3种异常现象，网上有很多相关的定义，其实很多说的都不太准确，我们这里直接从92标准文档中摘取出来3种异常现象的定义

事务的基础

脏读

事务T1更新了一行，事务T2可以在事务T1提交前读到这个行。
如果T1执行了回退，T2会读到一个从未提交的行

脏读有个明显的问题，用户可能不知道钱是不是到账了，在事务没有完成前用户就可以查到账户里有钱已经转入了，但是某些原因失败了导致事务回退，钱又没了，这对用户来说是难以理解的。

事务的基础

CHINA POSTGRESQL ASSOCIATION

不可重复读

事务T1读取了一行，事务T2更新或删除了这行并提交了。如果T1再次读取这行，它会发现行被更改或删除了

幻读

事务T1通过某些条件读取到了N行，事务T2执行sql生成了行并满足了这个条件，T1重复读取时发现了不一致的行结果

不可重复读和幻读的区别在于一是其他事务的更新或删除导致同一事务内读数不一致，一个是其他事务的插入导致同一事物内读数不一致

事务的基础

CHINA POSTGRESQL ASSOCIATION

不同的隔离级别分别对应不同的异常现象。

ANSI SQL-92标准的隔离级别和异象：

隔离级别	脏读	不可重复读	幻读
未提交读	可能	可能	可能
已提交读	不可能	可能	可能
可重复读	不可能	不可能	可能
可串行化	不可能	不可能	不可能

postgresql的隔离级别和异象：

隔离级别	脏读	不可重复读	幻读
未提交读	不可能	可能	可能
已提交读	不可能	可能	可能
可重复读	不可能	不可能	不可能
可串行化	不可能	不可能	不可能

事务的基础

CHINA POSTGRESQL ASSOCIATION

ANSI SQL-92标准的隔离级别和异象：

隔离级别	脏读	不可重复读	幻读
未提交读	可能	可能	可能
已提交读	不可能	可能	可能
可重复读	不可能	不可能	可能
可串行化	不可能	不可能	不可能

pgsql的隔离级别和异象：

隔离级别	脏读	不可重复读	幻读
未提交读	不可能	可能	可能
已提交读	不可能	可能	可能
可重复读	不可能	不可能	不可能
可串行化	不可能	不可能	不可能

这里简单的解释一下。未提交读为什么跟已提交读一样呢？
因为postgresql中没有未提交读，如果我们在postgresql数据库中设置事务隔离级别为未提交读，它会被当做已提交读来处理。然后再看，可重复读怎么跟可串行化又这么像呢？难道他俩在pgsql库中也一样？其实不是的，只是在92标准中看上去是一样的。

事务的基础

CHINA POSTGRESQL ASSOCIATION

为什么会发生PostgreSQL数据库隔离级别与“92标准”不一致的情况呢?

- 为什么未提交读与“92标准”不一致？未提交读实在是太奇怪了，在关系型数据库中基本想不到使用未提交读的场景。它严重违反了事务的隔离性。PostgreSQL将“未提交读”视为“已提交读”
- 为什么可重复读与“92标准”不一致？PostgreSQL通过快照实现MVCC多版本并发控制，PostgreSQL中的可重复读级别实际上是快照隔离级别，这个隔离级别实际上没有幻读这个异常现象
- 虽然“92标准”影响深远，但是还是有许多数据库没有全部实现它。
- ANSI SQL-92标准定义模糊。“92标准”在数据库行业很有代表性，它很好，但不够好。

事务的历史

CHINA POSTGRESQL ASSOCIATION

1981	Jim Gray提出事务的3个特征：ACD。原子性、一致性、持久性。
1983	Theo Haerder and Andreas Reuter提出了ACID。
1992	美国国家标准协会认证了ANSI SQL-92标准，也就是广泛流传的4种隔离级别和3种异象。
1995	微软工程师等对ANSI SQL-92做出批判，92标准定义模糊，而且有许多隔离级别和异常现象未定义。此时隔离级别已不止4个，异常现象也更多。
1999	由于锁模式的不同发展出过多的隔离级别， Atul Adya整理了这些现象，并根据异常现象和功能将众多隔离级别回溯到ANSI SQL92标准进行对应。
2005	由于绝大部分数据库声称他们是可串行化的，但他们实际上是快照隔离， Alan Fekete etal 提出“使快照隔离可串行化” 。在snapshot isolation级别基础上实现可串行化。
2008	Fekete 扩展了可串行化，并提出在数据库层面实现“使快照隔离可串行化”，称之为快照隔离可串行化 (Serializable Snapshot Isolation， SSI)。
2012	PostgresSQL第一个在商业数据库中实现SSI。

事务的历史

CHINA POSTGRESQL ASSOCIATION

- 在92标准出来不久，一些微软工程师和学者对92标准做了批评，并在提出了更多的隔离级别和异常现象
- 之前92标准中定义了4个隔离级别和3个异常现象，在“对92的批评”中有6种隔离级别和8种异常现象

Table 4. Isolation Types Characterized by Possible Anomalies Allowed.

Isolation level	P0 Dirty Write	P1 Dirty Read	P4C Cursor Lost Update	P4 Lost Update	P2 Fuzzy Read	P3 Phantom	A5A Read Skew	A5B Write Skew
READ UNCOMMITTED == Degree 1	Not Possible	Possible	Possible	Possible	Possible	Possible	Possible	Possible
READ COMMITTED == Degree 2	Not Possible	Not Possible	Possible	Possible	Possible	Possible	Possible	Possible
Cursor Stability	Not Possible	Not Possible	Not Possible	Sometimes Possible	Sometimes Possible	Possible	Possible	Sometimes Possible
REPEATABLE READ	Not Possible	Not Possible	Not Possible	Not Possible	Not Possible	Possible	Not Possible	Not Possible
Snapshot	Not Possible	Not Possible	Not Possible	Not Possible	Not Possible	Sometimes Possible	Not Possible	Possible
ANSI SQL SERIALIZABLE == Degree 3 == Repeatable Read Date, IBM, Tandem, ...	Not Possible	Not Possible	Not Possible	Not Possible	Not Possible	Not Possible	Not Possible	Not Possible

- 更多的隔离级别和异常现象出现，它们没有在ANSI SQL-92中定义
- 快照隔离在可重复读和可串行化间，这也是为什么pgsql的可重复读跟可串行化这么像的原因之一
- 写偏序异常提出

事务的历史

CHINA POSTGRESQL ASSOCIATION

3款最热门的关系型数据库的默认隔离级别和支持的最大隔离级别：

Database	Default isolation	Maximum Isolation
Mysql	RR	S
Oracle	RC	SI
PostgreSQL	RC	S

* RC:read committed，RR：repeatable read， S：serializability，SI：snapshot isolation

- mysql在serializability隔离级别下，读对数据加读共享锁，读会阻塞写;
- oracle其实也可以设置serializability隔离级别，并号称是支持可串行化的，但是它不是真正的可串行化，只是snapshot isolation;
- PostgreSQL支持serializability，它在snapshot isolation基础上实现可串行化，全称为可串行化快照隔离Serializable Snapshot Isolation (SSI)， 读写不会相互阻塞

事务的历史

CHINA POSTGRESQL ASSOCIATION

为什么oracle欺骗了我们?

如果我们把snapshot isolation加到ANSI SQL-92标准中

Isolation level	Dirty read	Nonrepeatable read	Phantom
Read uncommitted	Yes	Yes	Yes
Read committed	No	Yes	Yes
Repeatable read	No	No	Yes
Snapshot	No	No	No
Serializable	No	No	No

- “92标准”的异象定义较少，也没有定义快照隔离，以92标准来看，snapshot isolation看上去跟serializable差不多
- 大部分关系型数据库都以“92标准”为标准，包括oracle。但是随后更好的标准出现后，他们没有做出改变

事务的历史

CHINA POSTGRESQL ASSOCIATION

为什么弱隔离级别在学术上有问题，实际上没出现严重问题?

1.非可串行化隔离级别的异常现象，一般都需要在高并发情况下才会发生，低并发数据库不太会出现问题；

2.当异常现象真的发生的时候，有些应用可能没发现异常现象或检查到异常但对他们不重要；

3.有可能数据异常了，但应用只是返回报错，并进入数据异常处理程序；

4.成本过高。不仅是数据库串行化隔离级别开发成本高，应用对可串行化也需要适应成本。光是理解这部分复杂的理论就不是一件容易的事；

5.高级别的隔离会丢失一些性能。大量的改造工作可能是吃力不讨好的，应用需要在“高并发”和“无异常现象”间做抉择；

6.业务基于机制开发，而不是规则开发。业务多少有点适应弱隔离级别的异常现象，特别是RC 。

事务的历史

CHINA POSTGRESQL ASSOCIATION

可串行化有什么意义?

1.虽然应用适应了弱隔离级别，但是不代表他们真的理解了
2.使用可串行化，应用可以极大的减少对数据异常的担忧
3.除了可串行化外，其他的隔离级别都有各自的异常现象，他们也不完全满足ACID的隔离性
4.可串行化可以消灭异常现象这个“蛀虫”，完全保证数据的安全性
5. 可串行化在理论上已经证明可以实现
6.一些可串行的实现确实极大的降低了并发性，但是还有其他的可串行化实现，对并发性影响很小。比如说，可串行化快照隔离SSI可串行化的实现

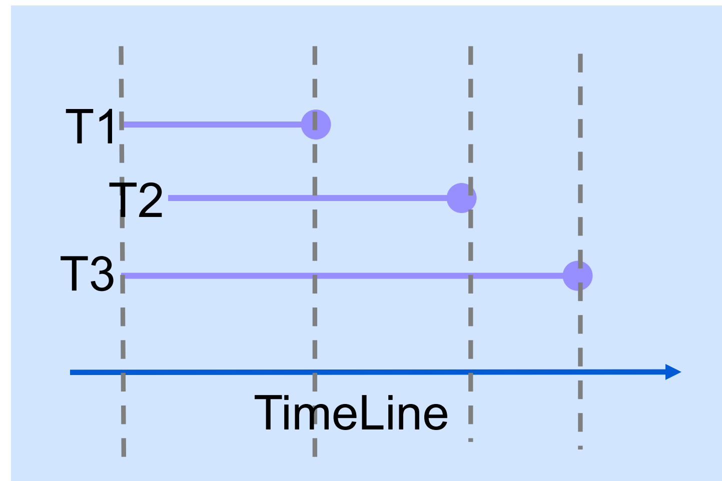
SSI理论知识

CHINA POSTGRESQL ASSOCIATION

可串行化

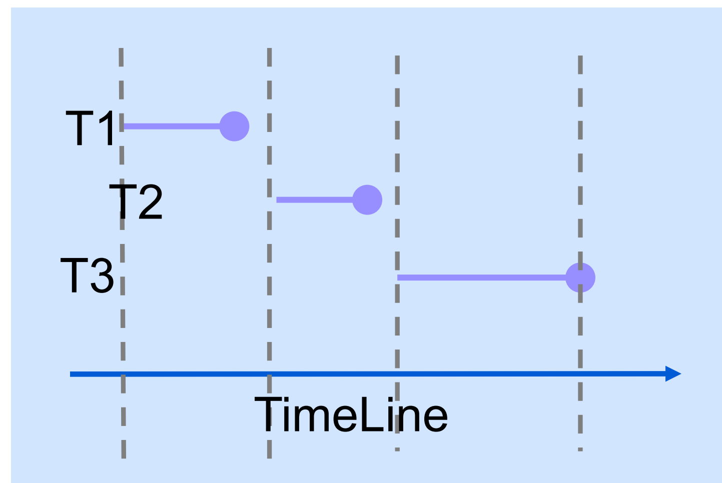
可串行化的含义

如果每个事务本身是正确的，即满足某些完整性条件，那么包括这些事务的任何串行执行的时间表是正确的（其事务仍然满足其条件）：“串行”意味着事务在时间上不重叠，并且不能相互干扰，即彼此之间存在完全隔离。



可串行化的实现

事务发展早期，可串行化（serializable）通过严格两阶段锁（SS2PL）实现，读写相互阻塞，直到事务结束。消除了异常现象但SS2PL丢失了高性能。
除了SS2PL实现可串行化，还有其他方式，比如可串行化快照隔离（SSI）。



可串行化的意义

为了保证没有异常，可串行化会丢失一些并发性（不同实现方式有所不同），但可以真正保证数据的ACID隔离性。也就是说没有实现串行化的数据库，其实没有完全支持ACID特性。
可串行化在理论上已经证明可以实现，但是真实的数据库世界有点”不正常“。实际上，可串行化是事务隔离级别中最高级的，也是学者和业界大佬强力推荐的隔离级别，不过绝大部分数据库在RC或快照隔离级别上运行。

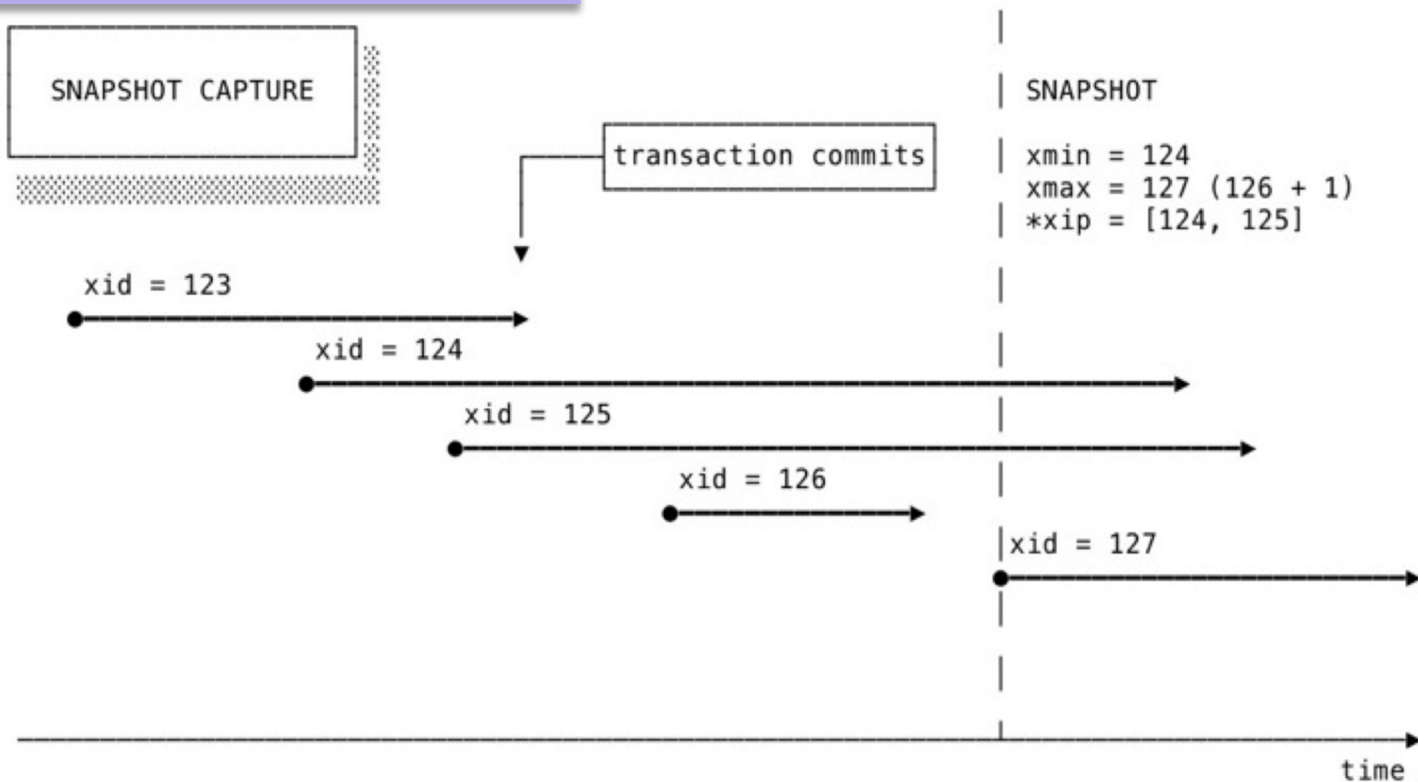
SSI理论知识

快照隔离

快照隔离的定义

在快照隔离下执行的事务是在事务开始时拍摄的数据库快照上操作的。当事务结束时，只有当事务更新的值自快照拍摄以来没有外部更改时，它才会成功提交。

快照隔离级别顾名思义就是就是使用了快照，这广泛用于实现MVCC，使多版本并发机制支持用户并发执行事务。



快照隔离的出现

ANSI SQL92并未定义快照隔离snapshot isolation(SI)，这个隔离级别随着数据库行业发展才出现。1992年 ANSI SQL92标准基于数据库的锁而定义，所以没有快照隔离级别这个定义。直到1995年《批判》的出现才被提出。

SSI理论知识

CHINA POSTGRESQL ASSOCIATION

SSI

可串行化快照隔离SSI

由于快照隔离的广泛应用，而可串行化是学术上数据库需要达到的隔离级别目标。可串行化快照隔离Serializable Snapshot Isolation (SSI) 顾名思义，在快照隔离的基础上实现可串行化。

为什么有SSI?

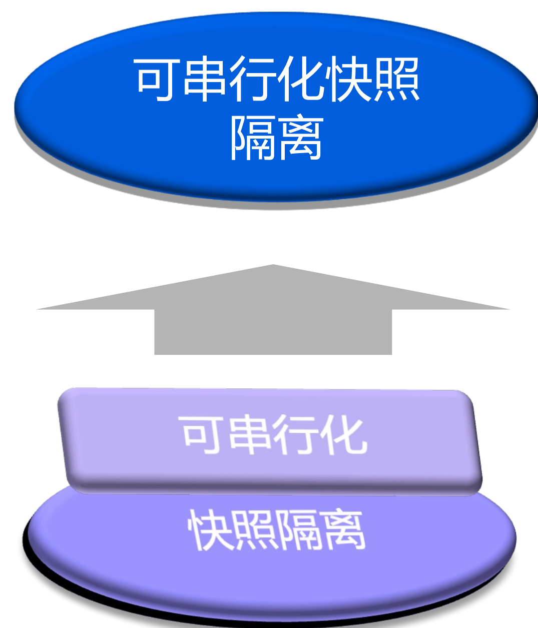
由于ANSI92标准的模糊性，虽然没有定义快照隔离，但许多数据库实际上就是使用的快照隔离。而快照隔离同样存在一些异常现象（包括写偏序），SSI的出现就是为了解决这些异常现象。

SSI相对于S2PL的优势

传统的可串行化通过S2PL实现，在S2PL下写操作会阻塞其他事务读写，虽然可以实现可串行化，不会有写偏序异常等问题，但会产生很多锁冲突，降低并发性能。而通过快照实现的MVCC读写互不阻塞，只有写写冲突。在此基础上实现的SSI对并发性的影响相比传统S2PL要小很多。

PG实现SSI

SSI在pg9.1开始实现SSI，是第一个实现SSI的商业数据库。



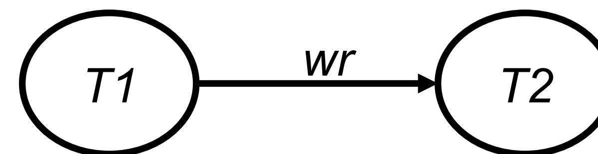
SSI理论知识

CHINA POSTGRESQL ASSOCIATION

3种依赖关系

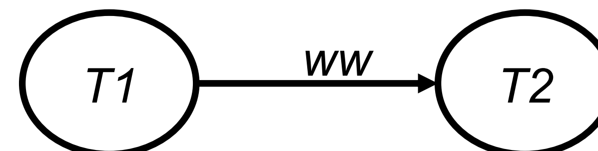
wr-dependencies

事务T1写数据项的一个版本，事务T2读**这个版本**，意味着T1先于T2



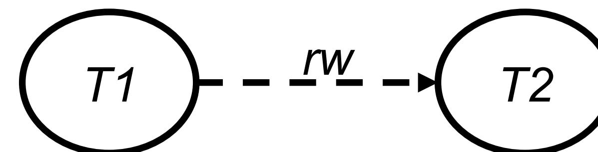
ww-dependencies

事务T1写数据项的一个版本，事务T2使用一个新版本替换**这个版本**，意味着T1先于T2



rw-antidependencies

事务T1写数据项的一个版本，事务T2读这个对象**之前的版本**，意味着T1后于T2执行



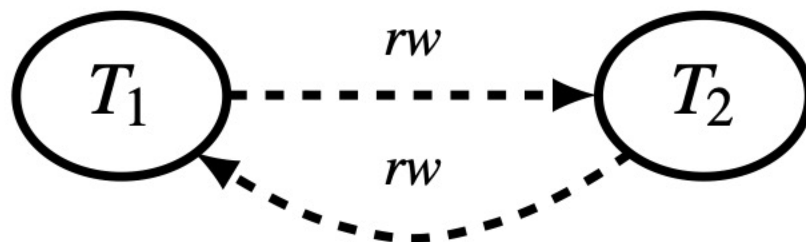
Precedence graph

SSI理论知识

CHINA POSTGRESQL ASSOCIATION

写偏序

由于某些冲突构成环，会出现串行化异常。也就是说，有些并行执行的事务，从理论上就是不可串行化的。其中比较容易理解的一个就是写偏序(write skew)。
写偏序只发生在rw模型，ww、wr均不会发生写偏序，并且事务必须在并发条件下才会出现。



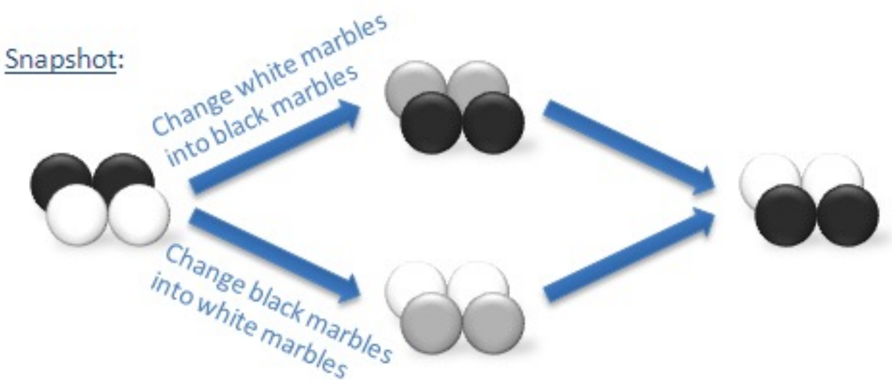
(a) Example 1: Simple Write Skew

简单写偏序：事务T1读写反依赖T2，T2又读写反依赖T1。这两个事务的并发执行是不可串行化的。

SSI理论知识

有许多现实案例可以出现写偏序异常，我们用一个经典的**黑白球问题**来理解写偏序
袋中有4个球，2个白球和2个黑球。此时有两个事务，P和Q。P将所有黑球改成白球，Q将所有白球改成黑球。此时可以有两个串行执行，<P，Q>或<Q，P>。在这两种情况下，最终结果是袋中有4个白球或者4个黑球。
但是，快照隔离允许另一种结果：
•事务 P 拿出2个黑球
•事务 Q 拿出2个白球
•事务 P 将手中所有黑球改成白球，放回袋中
•事务 Q 将手中所有白球改成黑球，放回袋中
此时袋中还是2个黑球和2个白球，这在任何一个串行执行中都是不可能的。但这在快照隔离中是有效：每个事务都维护数据库的一致视图，并且其写集不与任何并发事务的写集重叠，如此白球黑球发生交换。

黑白球问题说明：快照隔离执行结果与串行化执行结果不一致，快照隔离下发生写偏序异常，数据结果与预期不一致。



PostgreSQL中的SSI

set default_transaction_isolation = 'serializable';	set default_transaction_isolation = 'serializable';
begin; update dots set color = 'black' where color = 'white';	
	begin; update dots set color = 'white' where color = 'black';
commit	
	commit
	ERROR: could not serialize access due to read/write dependencies among transactions DETAIL: Reason code: Canceled on identification as a pivot, during commit attempt. HINT: The transaction might succeed if retried.

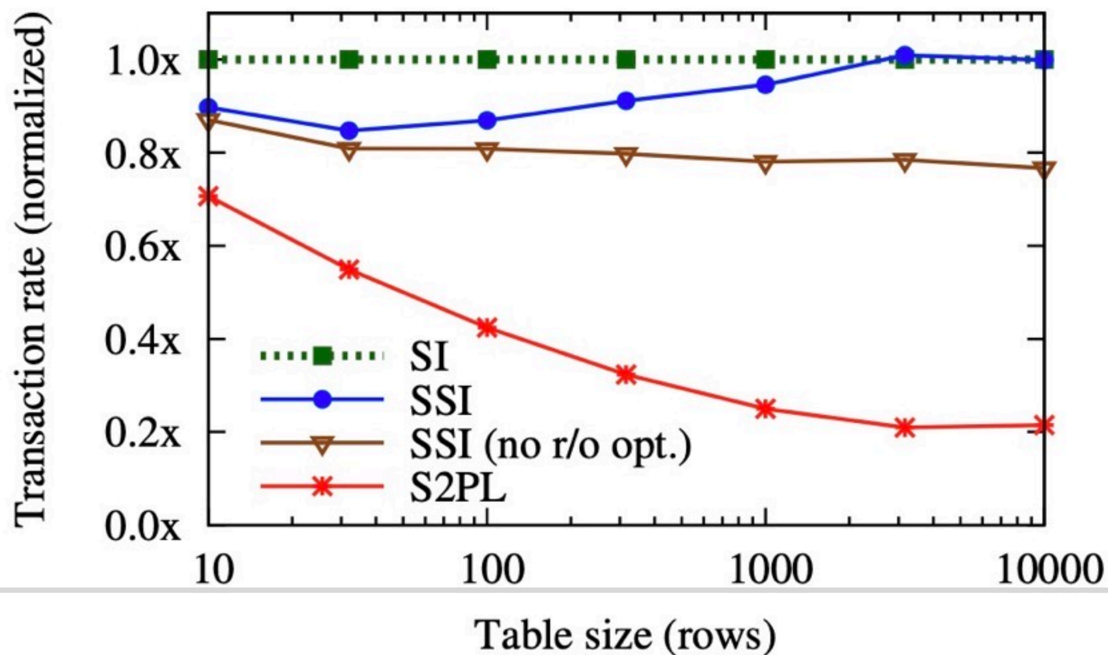
** pg SSI先提交者成功提交后提交者抛出报错*
** 在其他隔离级别下，不会抛出报错*

PostgreSQL中的SSI

CHINA POSTGRESQL ASSOCIATION

PostgreSQL对SSI实现的优化

- 安全快照：不会引起环形结构的只读事务，不必检测冲突，可减少检查负担和内存负担
- 延迟事务：延迟事务可重新尝试。在检查到”危险结构“后，延迟事务会被取消，然后再次尝试执行。延迟事务需要显示声明
- 检测粒度升级：多个细粒度的锁可合成粗粒度的锁减少内存开销



	Throughput (req/s)	Serialization failures
SI	435	0.004%
SSI	422	0.03%
S2PL	208	0.76%

PostgreSQL中的SSI

CHINA POSTGRESQL ASSOCIATION

- 可串行化能简化系统开发的问题，开发人员不需要管并发下事务的异常现象，特别是如今更多的高并发系统下。
- PostgreSQL的可串行化显然比严格两阶段提交模式更好。不仅性能更好，而且中止事务的概率也更低。
- PostgreSQL是第一个实现SSI的商业数据库，而很多传统关系型数据库根本不支持可串行化，pgsql数据库往前走了一大步。
- PostgreSQL不仅实现了SSI，还在此基础上做了很多优化，比如只读事务和内存优化，并且效果显著

THANKS

刘智龙 DBA

www.postgresqlchina.com