

PGCONF.RUSSIA 2024

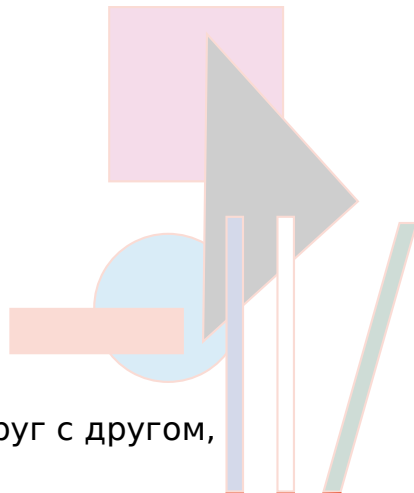
Консистентность в
распределенных системах на
базе PostgreSQL

Никита Печёнкин
n.pechenkin@postgrespro.ru



Что такое консистентность?

$$C_{db} \neq C_{distributed\ system} \neq C_{ddb}$$



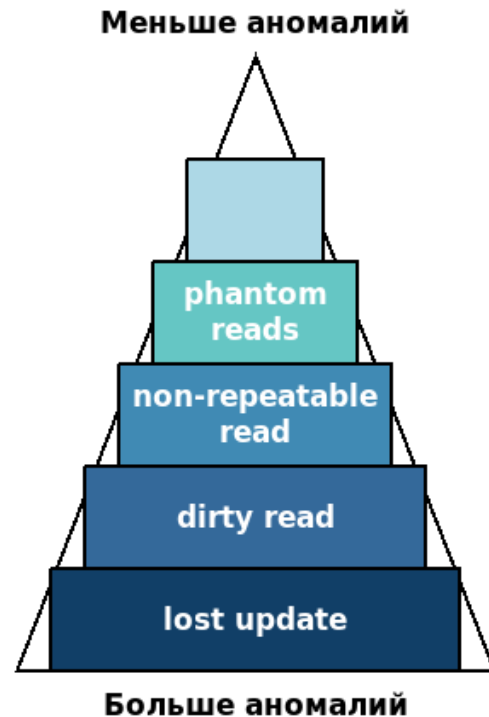
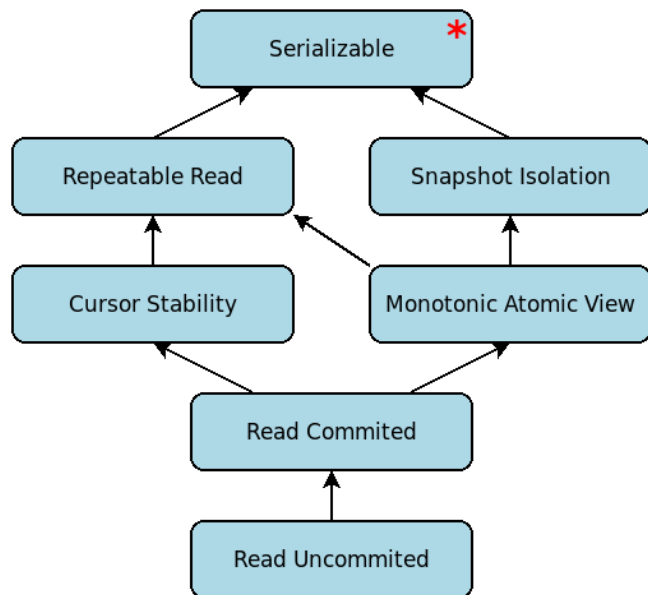
Консистентность данных (англ. data consistency) — согласованность данных друг с другом, целостность данных, а также внутренняя непротиворечивость (wikipedia.org)

Согласованность в СУБД

A = Atomicity
C = Consistency
I = Isolation
D = Durability

- Только **транзакции** переводят базу данных из одного согласованного состояния в другое согласованное состояние.
- Каждая транзакция может читать и изменять **более одного объекта**.
- Множество допустимых состояний определяется в том числе и **уровнем изоляции** транзакций.
- Иерархия уровней согласованностей совпадает с иерархией уровней изоляции транзакций.

Согласованность в СУБД



* порядок операций может не совпадать с реальным порядком выполнения операций в смысле физической упорядоченности

Согласованность в распределенных системах

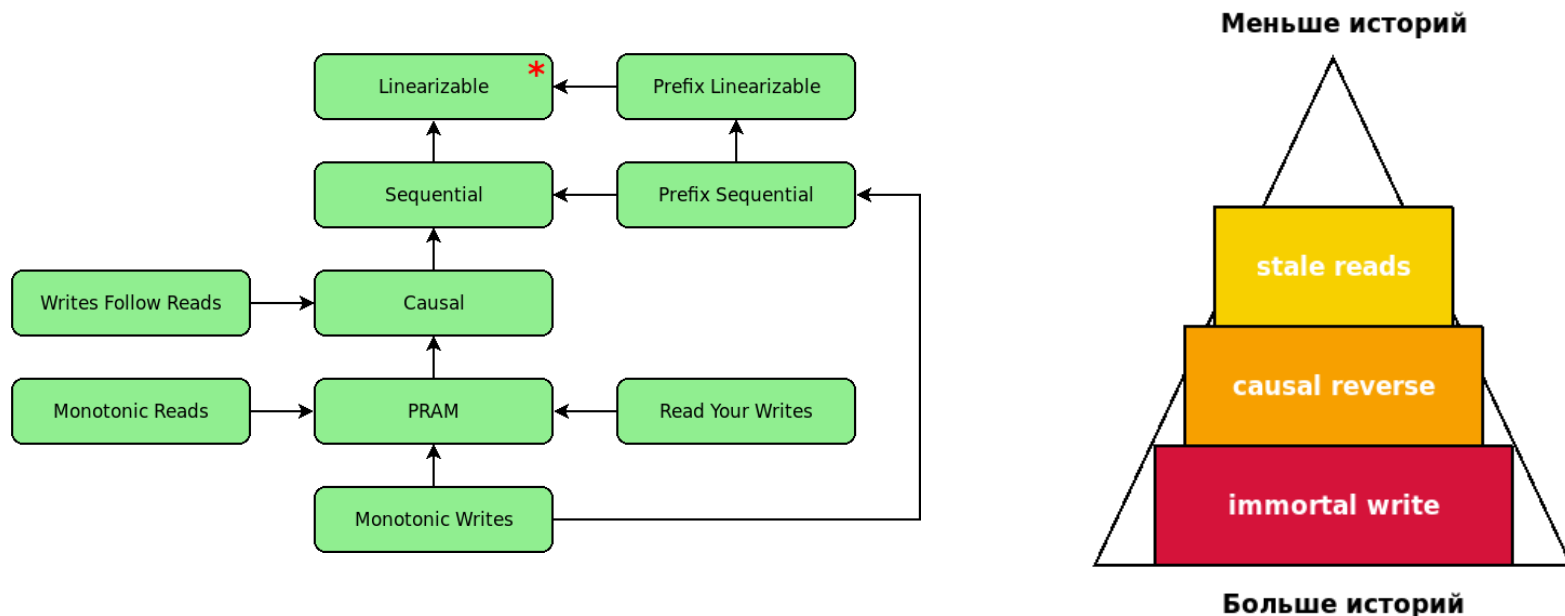
C = Consistency

A = Availability

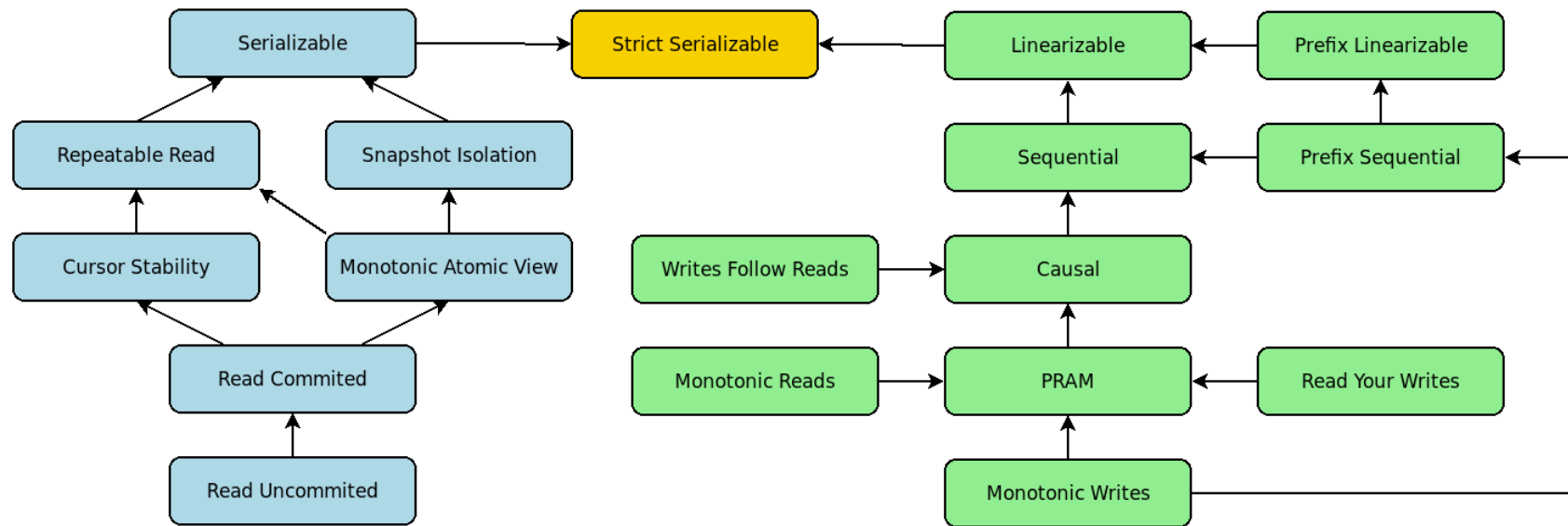
P = Partition Tolerance

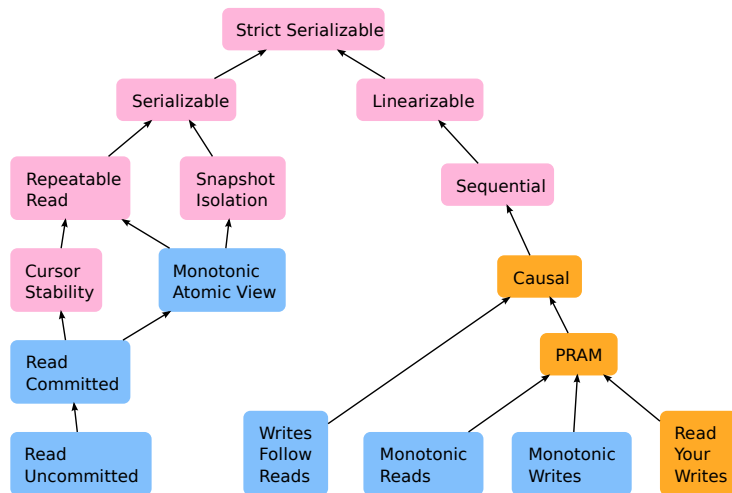
- Изменение системы является следствием выполнения **операций**.
- Консистентность в распределенных системах подразумевает линеаризацию операций чтения и изменения **отдельных значений**.
- Аномалии связаны с сетевым обменом, репликацией и распределением данных по узлам.
- Не рассматриваются аномалии над группой объектов.

Согласованность в распределенных системах



* предполагает строгую упорядоченность - результат записи становится видимым для всех в точности один раз в некий момент времени между началом и окончанием операции.





Legend

Unavailable

Not available during some types of network failures. Some or all nodes must pause operations in order to ensure safety.

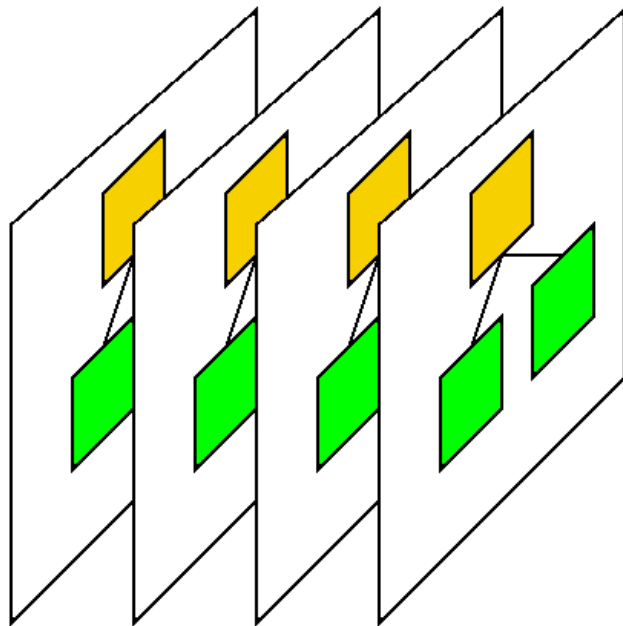
Sticky Available

Available on every non-faulty node, so long as clients only talk to the same servers, instead of switching to new ones.

Total Available

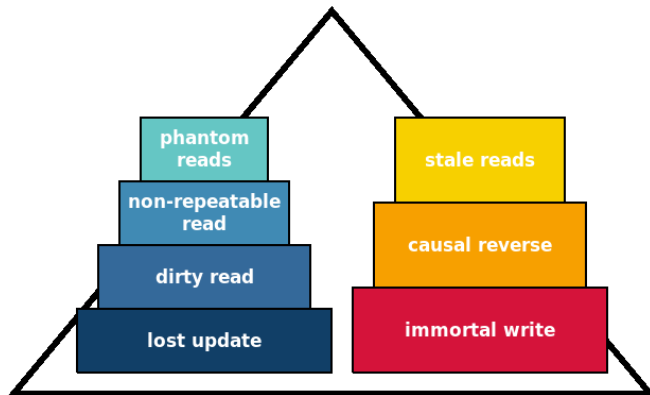
Available on every non-faulty node, even when the network is completely down.

- Peter Bailis et al. 2013. **Highly available transactions: virtues and limitations.**
- Paolo Viotti and Marko Vukolić. 2016. **Consistency in Non-Transactional Distributed Storage Systems.**



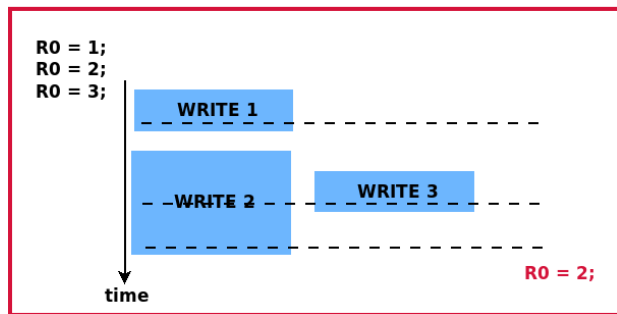
- 1) **partition** - для операций внутри одной партиии
- 2) **session** - для операций внутри одной сессии
- 3) **write** - только для пишущих операций
- 4)

- Гарантии **строже**
- Уровень изоляции **выше**
- Доступность **ниже** (CAP)
- Производительность **ниже** (PACELC)

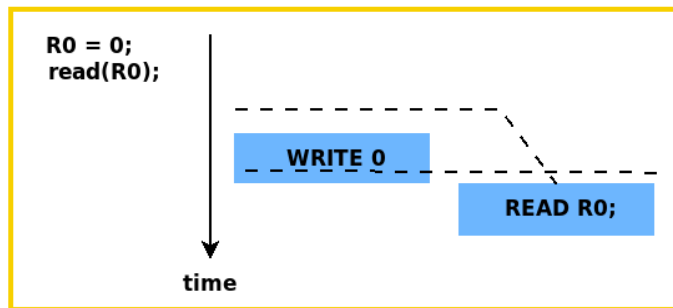


- Гарантии **мягче**
- Уровень изоляции **ниже**
- Доступность **выше** (CAP)
- Производительность **выше** (PACELC)

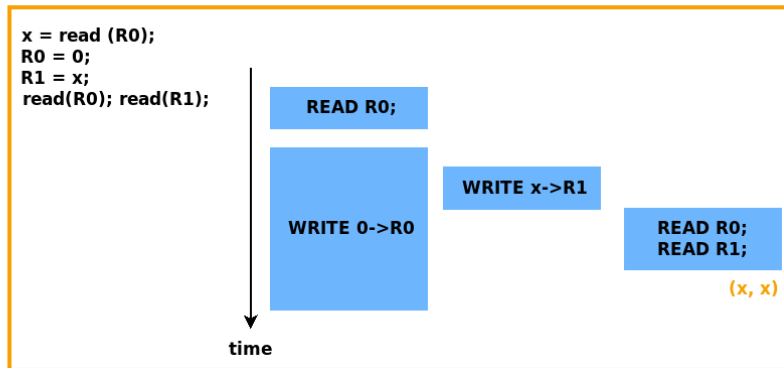
Immortal write



Stale reads



Causal reverse

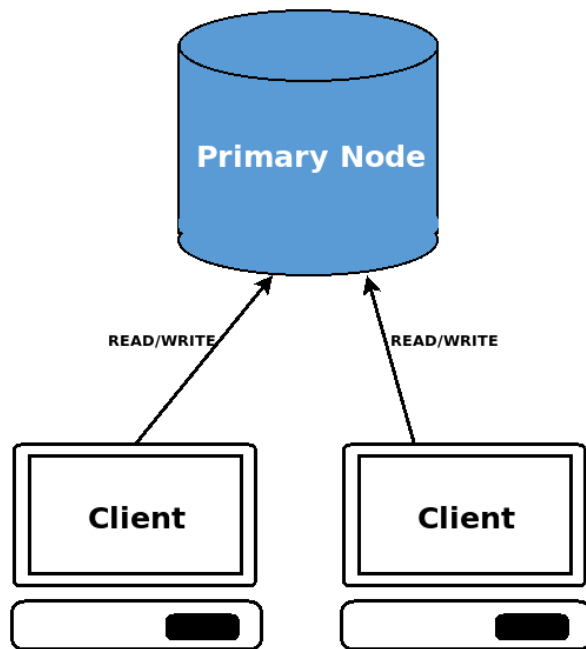


System Guarantee	Dirty read	Non-repeatable read	Phantom Read	Write Skew	Immortal write	Stale read	Causal reverse
READ UNCOMMITTED	POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE
READ COMMITTED	NOT POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE
REPEATABLE READ	NOT POSSIBLE	NOT POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE
SNAPSHOT ISOLATION	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE
SERIALIZABLE / ONE COPY SERIALIZABLE / STRONG SESSION SERIALIZABLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	POSSIBLE	POSSIBLE	POSSIBLE
STRONG WRITE SERIALIZABLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	POSSIBLE	NOT POSSIBLE
STRONG PARTITION SERIALIZABLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	POSSIBLE
STRICT SERIALIZABLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE	NOT POSSIBLE

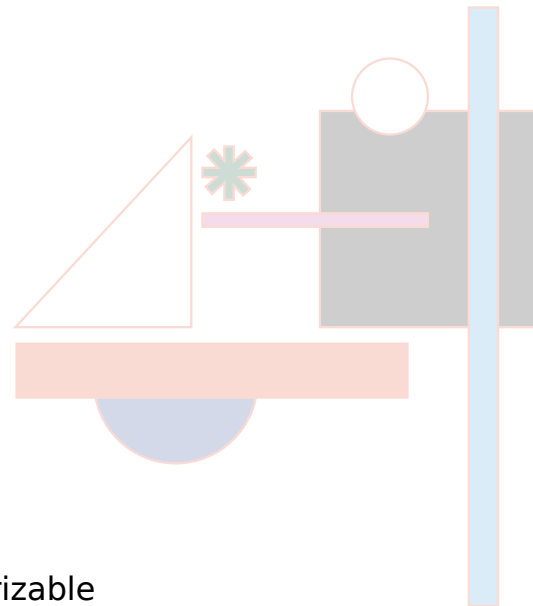
По материалам Daniel Abadi, **Correctness Anomalies Under Serializable Isolation**

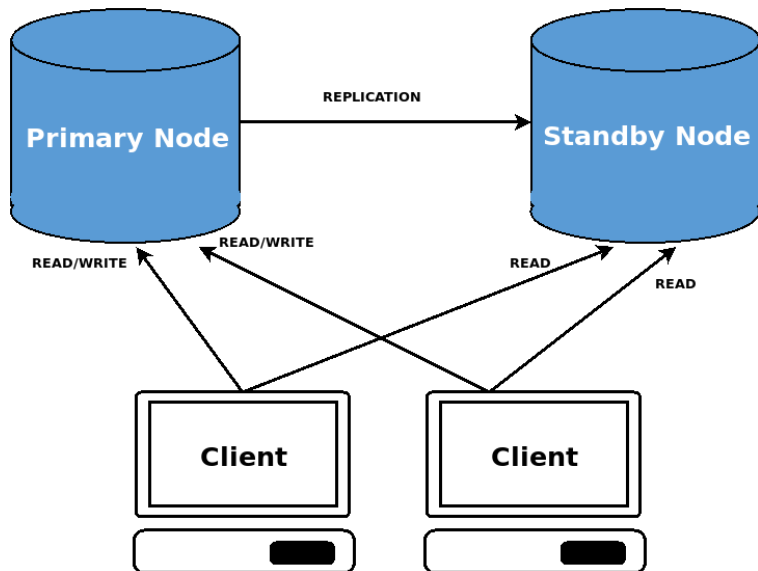
PostgresPro

Одиночный узел



isolation level serializable, consistency level linearizable





Синхронная реплика

isolation level serializable, consistency level write
(linearizable, sequential)

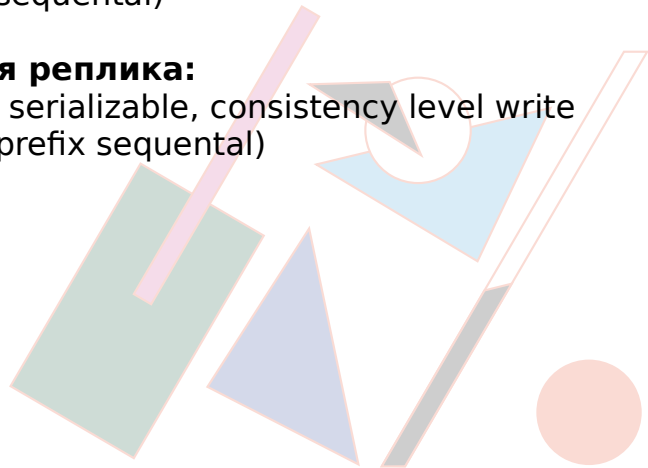
Синхронная реплика

(`synchronous_commit = remote_apply`):

isolation level serializable, consistency level write
(linearizable, sequential)

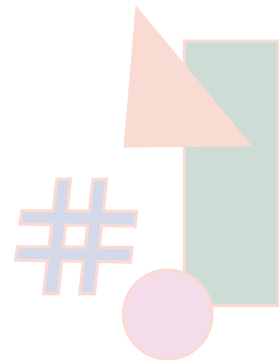
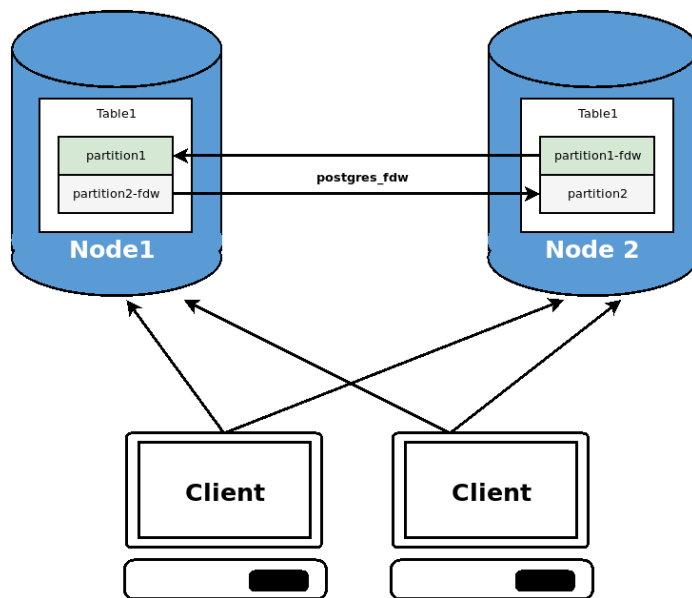
Асинхронная реплика:

isolation level serializable, consistency level write
(linearizable, prefix sequential)



PostgresPro

Секционирование



isolation level partition (snapshot, read uncommitted), consistency level -

Решения на базе Postgres (и не только)

CocroachDB	Почти strict serializability (возможен causal reverse)
YugaByte DB	Почти strict serializability (при условии согласования времени)
CitusDB	Snapshot в рамках одного узла, Read uncommitted, Linearizable на узле
Greenplum	Snapshot isolation, Linearizable
Shardman	Snapshot isolation, prefix-sequential linearizable (при условии согласовании времени)





Postgres Pro Shardman

1

Прозрачное
горизонтальное
масштабирование

2

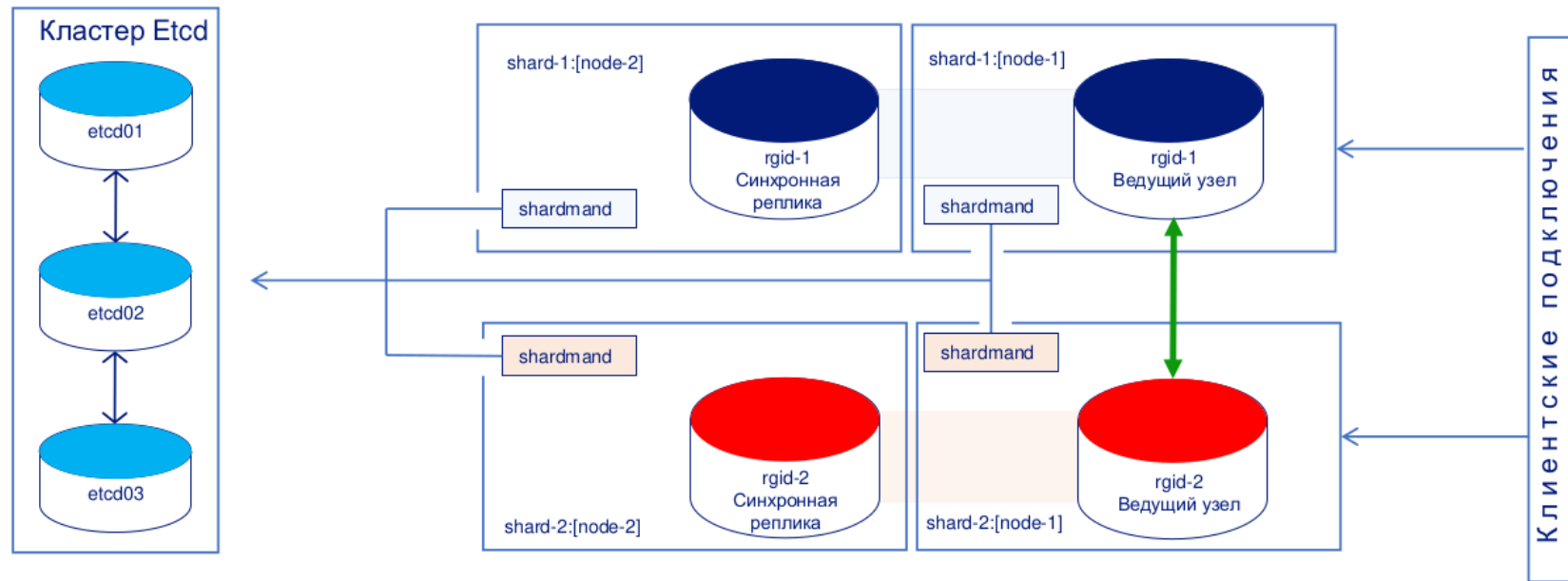
Доступ к кластеру
через любой узел

3

OLTP нагрузка

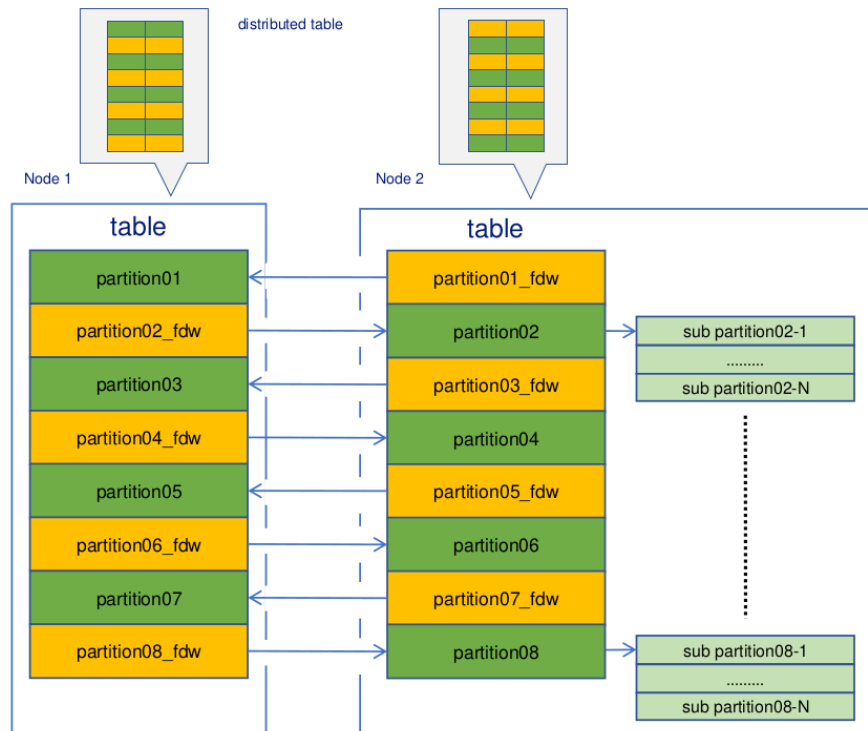
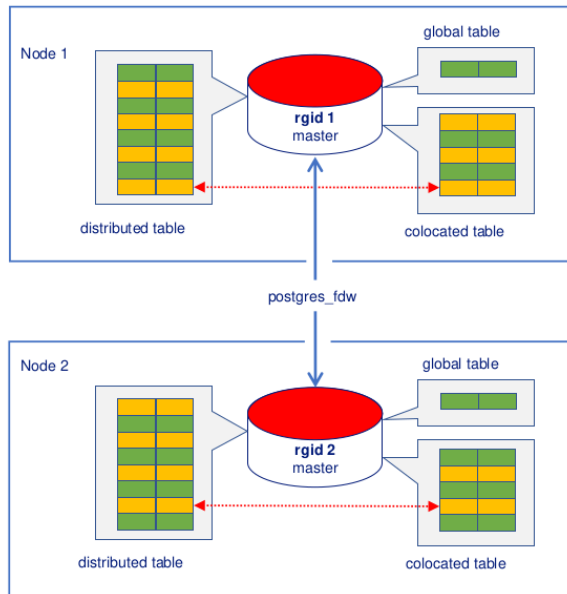
4

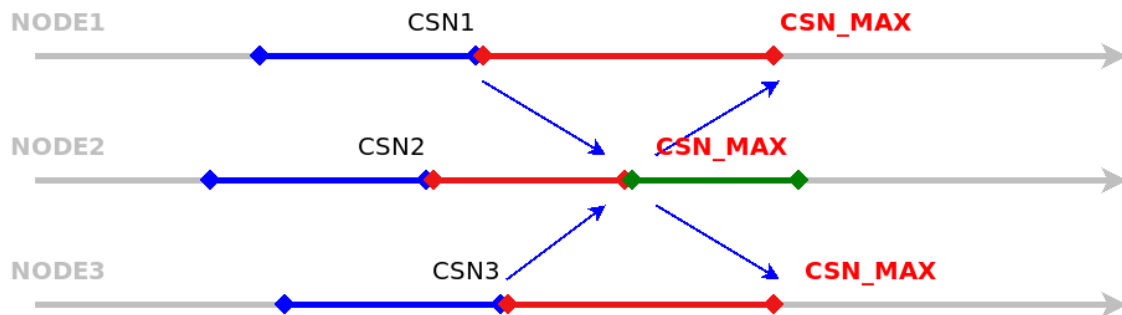
Распределенные
транзакции с высоким
уровнем изоляции





Postgres Pro Shardman





CSN_MAX = MAX(CSN1, CSN2, CSN3)

CSN_DELAY = CSN_MAX - CSN_MIN

- Уникальный CSN для снимков и коммитов.
- CSN - функция от времени.
- CSN глобально упорядочены и уникальны.
- Уровень изоляции Snapshot для любых распределенных транзакций.

Потенциальные аномалии в Shardman

- causal reverse при чтении из разных партиций и расхождении часов

пишущие транзакции могут писать в партиции, расположенные на разных серверах и использовать часы этих серверов, которые могут идти с обратным перекосом по времени, большим чем промежуток между транзакциями

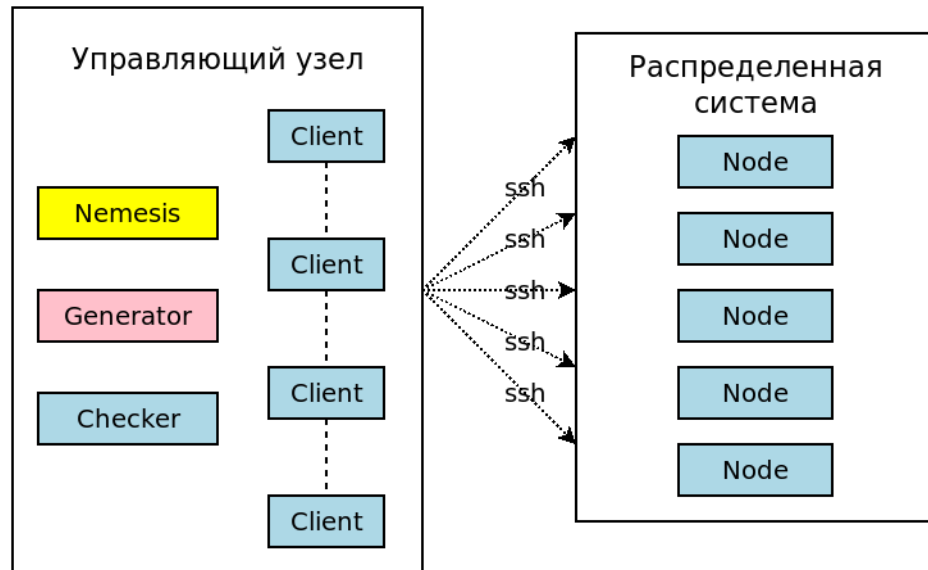
isolation level snapshot, consistency level partition (linearizable, sequential)

- stale reads при расхождении часов или изоляции узла

клиент сначала пишет через один сервер, а потом читает через другой, у которого отстают часы, при этом для пишущих операций обеспечивается актуальность данных

isolation level snapshot, consistency level write (sequential, prefix sequential)



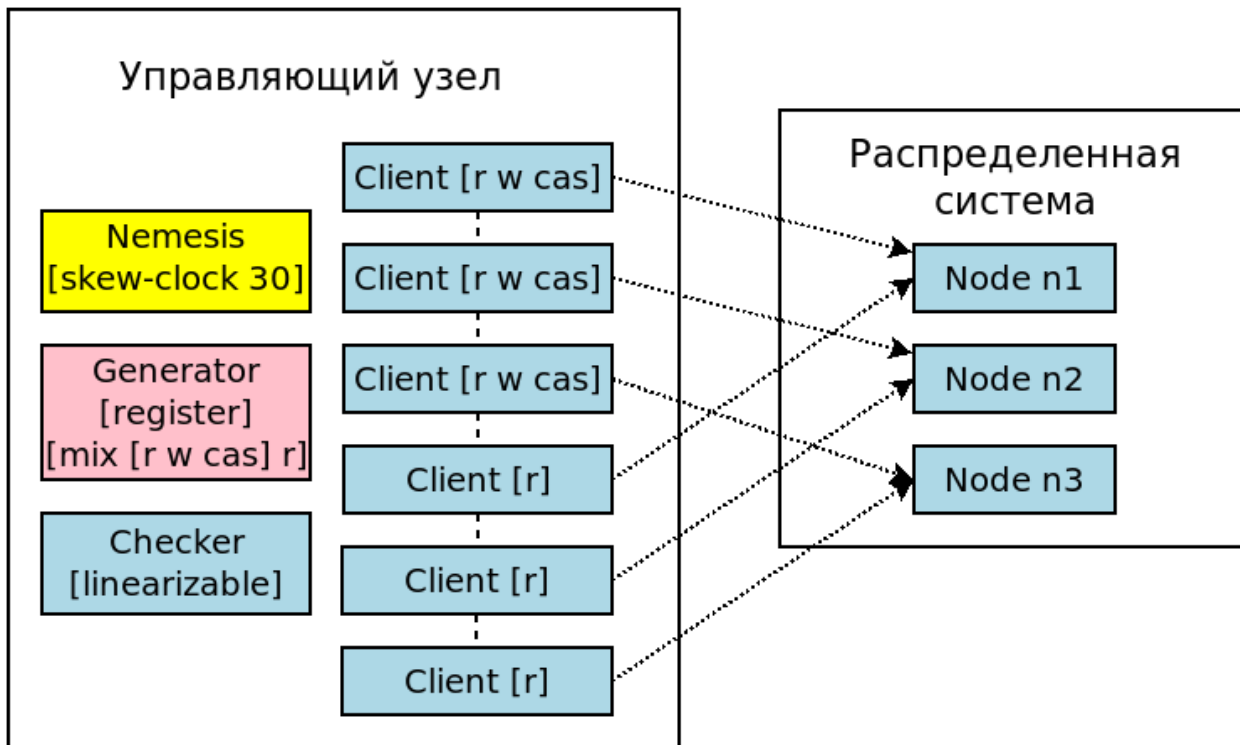


Типы нагрузок:

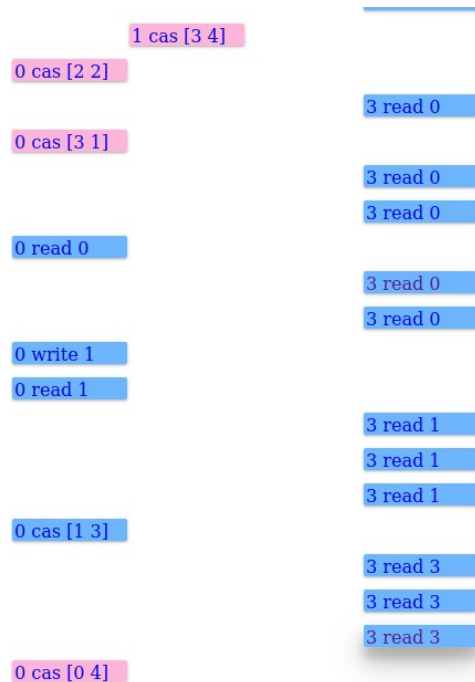
- register, bank, adya, causal, cycle

Типы аварий:

- partition-halves, clock-skew, node-start-stopper, noop, split-one



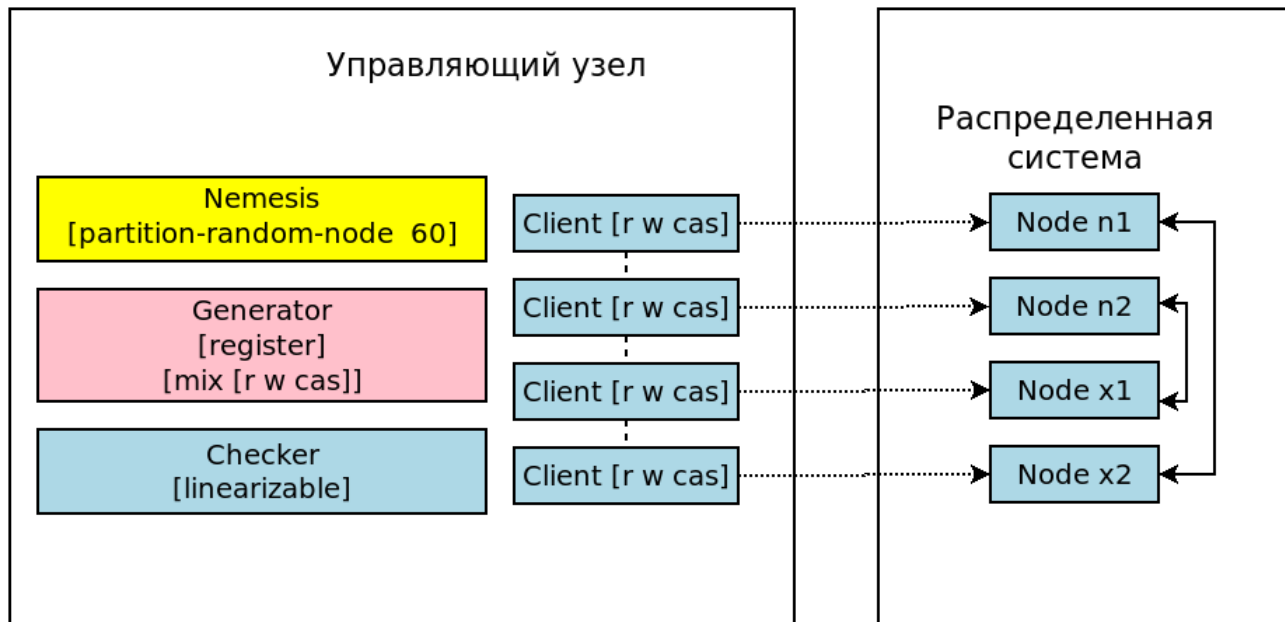
Аномалия сценария #1



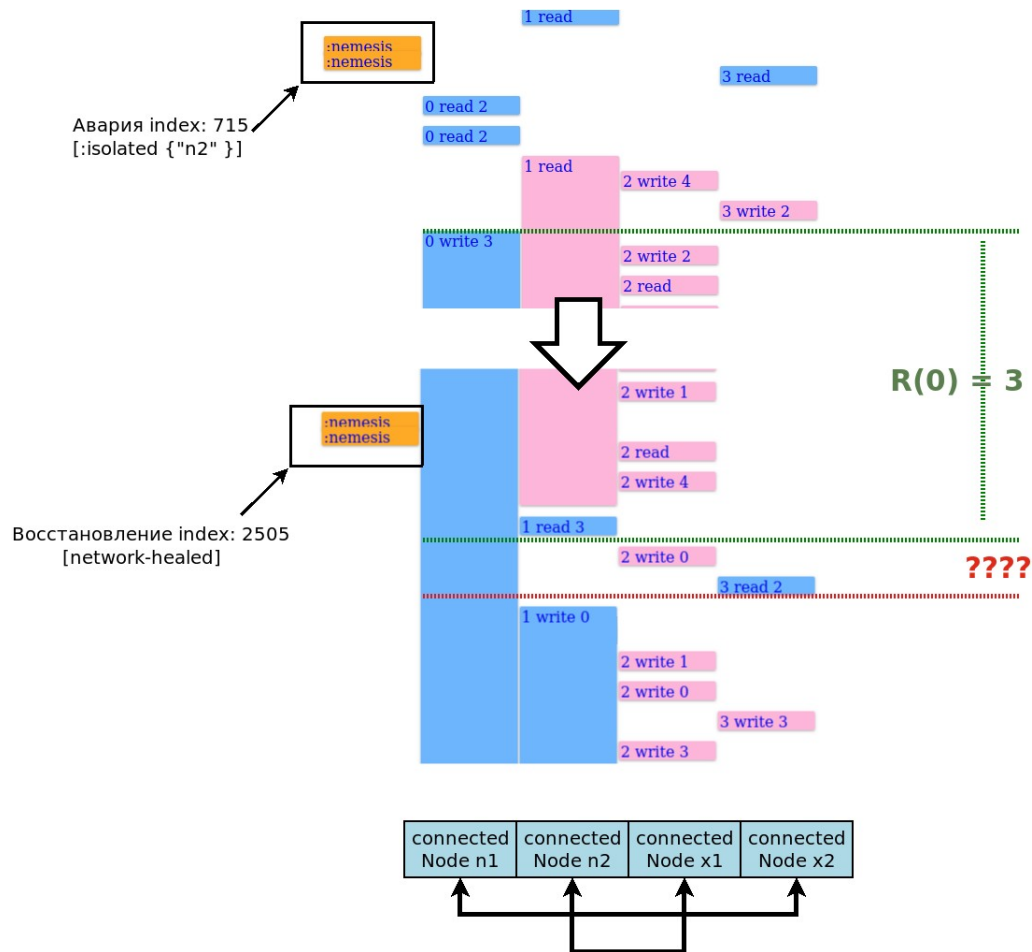
can't read 2 from register 0
 can't read 4 from register 0
 can't CAS 0 from 4 to 3

Everything looks good! \ (' - `) /

Сценарий #2: register + partition-random-node



Аномалия сценария #2



Как избавиться от аномалий?

1

задержать коммит на период неопределенности

2

не читать с реплик (реплики только для надежности)

3

отключать мастер в случае его сетевой изоляции

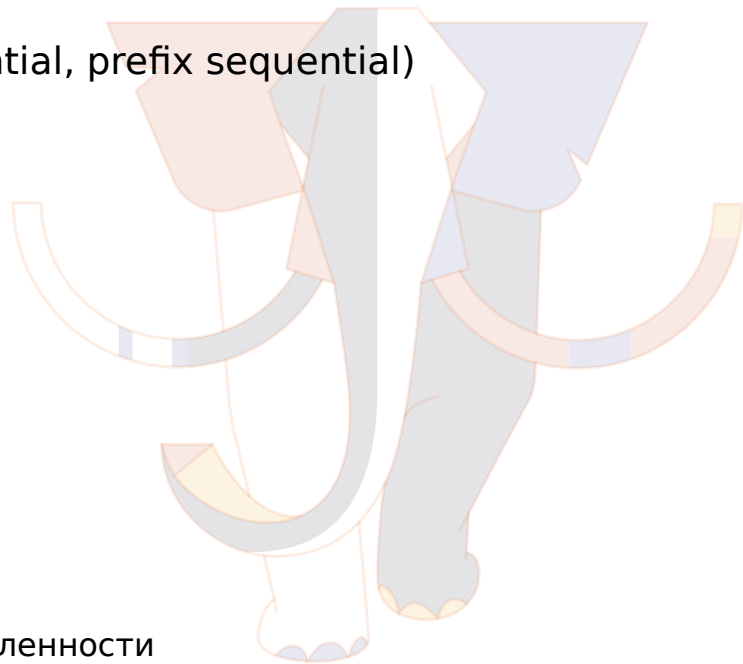


isolation level snapshot, consistency level write (sequential, prefix sequential)

ИЛИ

isolation level snapshot, consistency level linearizable *

* с включенным отложенным коммитом на период неопределенности



Спасибо за внимание!

