PGConf.Russia 2023

PostgresPro

RTT,RTO,RPO и синхронная репликация

Влияние сетевых задержек на производительность при использовании синхронной репликации

PostgresPro

Павел Конотопов
руководитель кластерной группы департамента внедрения и технической поддержки

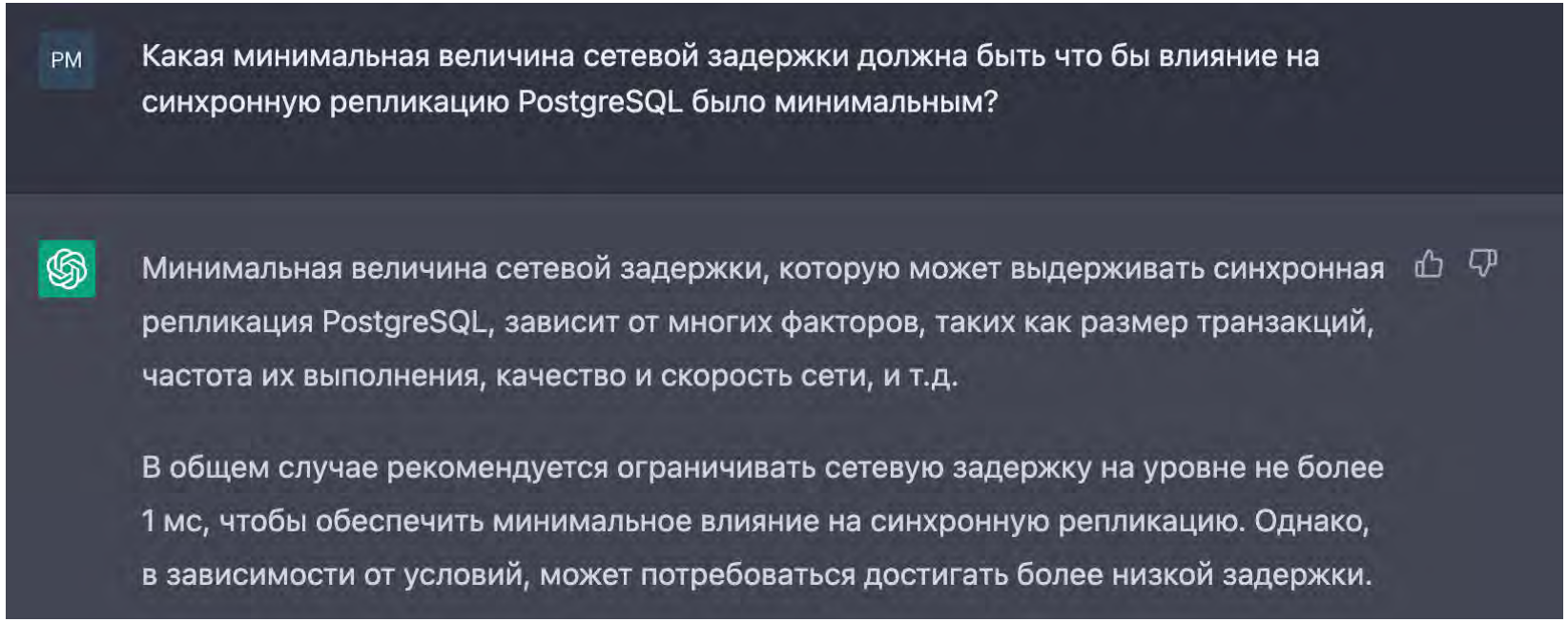
- 20+ лет в ИТ
- Инженер по отказоустойчивости PostgreSQL
- Последние пять лет работаю с PostgreSQL
- Последние два года работаю в Postgres Professional

Email:
p.konotopov@postgrespro.ru

PostgresPro

Самый частый вопрос

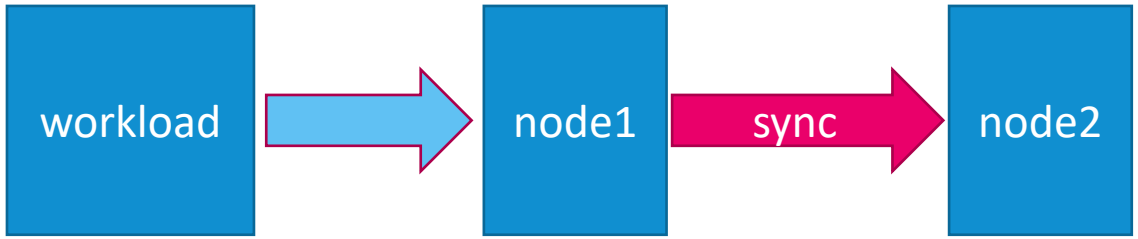
- Каким должно быть значение network latency при использовании синхронной репликации в PostgreSQL, чтобы производительность не падала?
- ChatGPT сказал – сетевая задержка должна быть 1 ms! ☺



PostgresPro

Задача

- Мастер и синхронная реплика
- Должен обеспечивать производительность от 100 до 300 TPS



PostgresPro

Тесты

- Тесты
  - pgbench
  - go-tpc

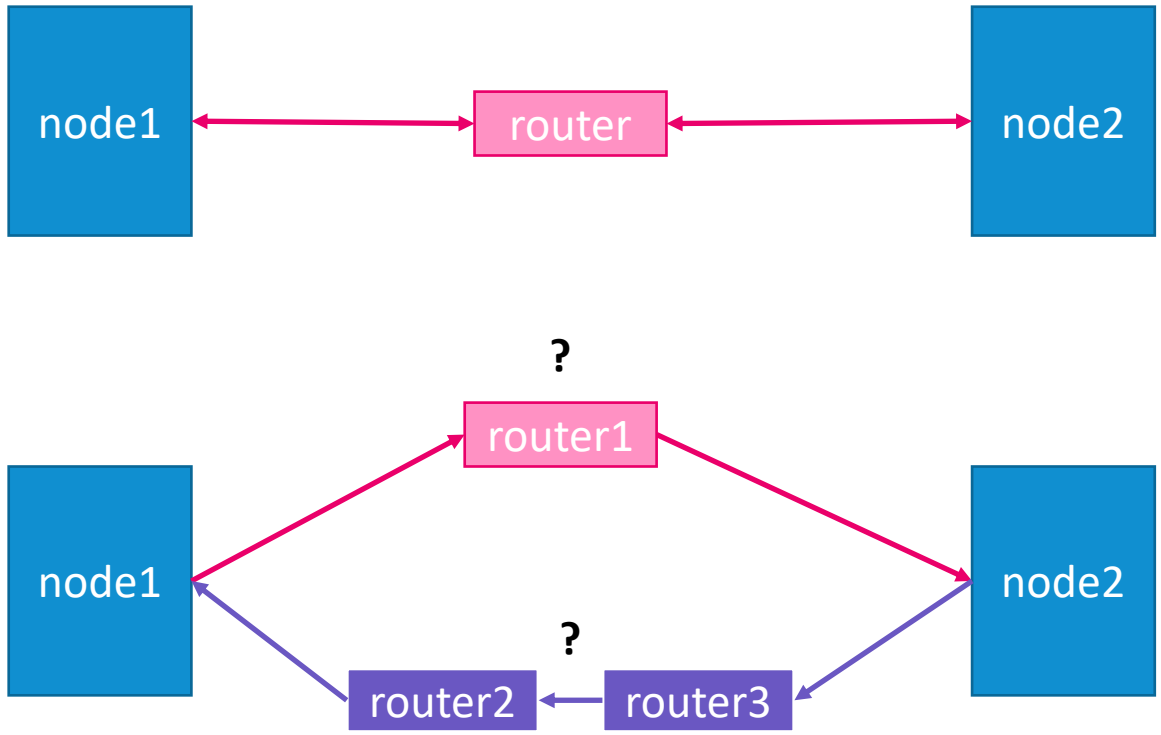
PostgresPro

Network latency и синхронная репликация

- Настройки репликации:
  - syncronous_commit
    - remote_write
    - on
    - remote_apply
  - commit_delay = 0
  - commit_siblings = 5
  - sycnronous_standby_names = node2
- Нестабильная сетевая среда:
  - увеличиваем latency от 1ms до 60ms
  - 1-20ms – шаг 1ms, 20-60ms – шаг 5ms

PostgresPro

Что такое network latency



PostgresPro

Эмуляция latency

- `comcast`
  - `Обертка над iptables/traffic control`

Пример измерения network latency

- **comcast**
  - node1
    - node1 -> node2 5ms
    - comcast --device=eth0 --latency=5 --target-addr=node2
  - node2
    - node2 -> node1 3ms
    - comcast --device=eth0 --latency=3 --target-addr=node1
- **chrony/ntp**
  - Настраиваем синхронизацию времени

PostgresPro

Чем измерять network latency

- iperf 2/3
  - умеет измерять двунаправленные сетевые задержки
- owamp
  - A One-way Active Measurement Protocol (OWAMP)
- twamp
  - A Two-way Active Measurement Protocol (TWAMP)

PostgresPro

Как измерять network latency

- owamp
  - Есть в каждом Linux дистрибутиве (почти)
  - Установка:
    - apt install chrony owamp
  - Запуск сервера:
    - owampd -f
  - Запуск измерений:
    - owping -c 60 -i 0.1 hostname

PostgresPro

Пример измерения network latency

- **ping**
  - node1:
    - ping node2
    - 64 bytes from node2 (192.168.21.114): icmp_seq=1 ttl=64 time=**8.41** ms
  - node2
    - ping node1
    - 64 bytes from node1 (192.168.21.113): icmp_seq=1 ttl=64 time=**8.43** ms
  - ?
- **owamp**
  - node1
    - owamp -c 60 -i .1 node2
    - one-way delay min/median/max = **4.58/4.8/4.9** ms, (err=0.497 ms)
  - node2
    - owamp -c 60 -i .1 node1
    - one-way delay min/median/max = **3.53/3.7/3.82** ms, (err=0.497 ms)

PostgresPro

Пример измерения network latency

```
--- owping statistics from [node1]:9395 to [node2]:36607 ---
SID:    c0a81572e7cbf116a9c2e33e4bf460a4
first:  2023-03-27T13:40:23.793
last:   2023-03-27T13:40:28.560
60 sent, 0 lost (0.000%), 0 duplicates
one-way delay min/median/max = 4.58/4.8/4.9 ms, (err=0.497 ms)
one-way jitter = 0.1 ms (P95-P50)
hops = 0 [1] (consistently)
no reordering


--- owping statistics from [node2]:58088 to [node1]:9728 ---
SID:    c0a81571e7cbf116aae2eb1c9ba31062
first:  2023-03-27T13:40:23.704
last:   2023-03-27T13:40:29.129
60 sent, 0 lost (0.000%), 0 duplicates
one-way delay min/median/max = 3.53/3.7/3.75 ms, (err=0.497 ms)
one-way jitter = 0.1 ms (P95-P50)
hops = 0 [2] (consistently)
no reordering
```

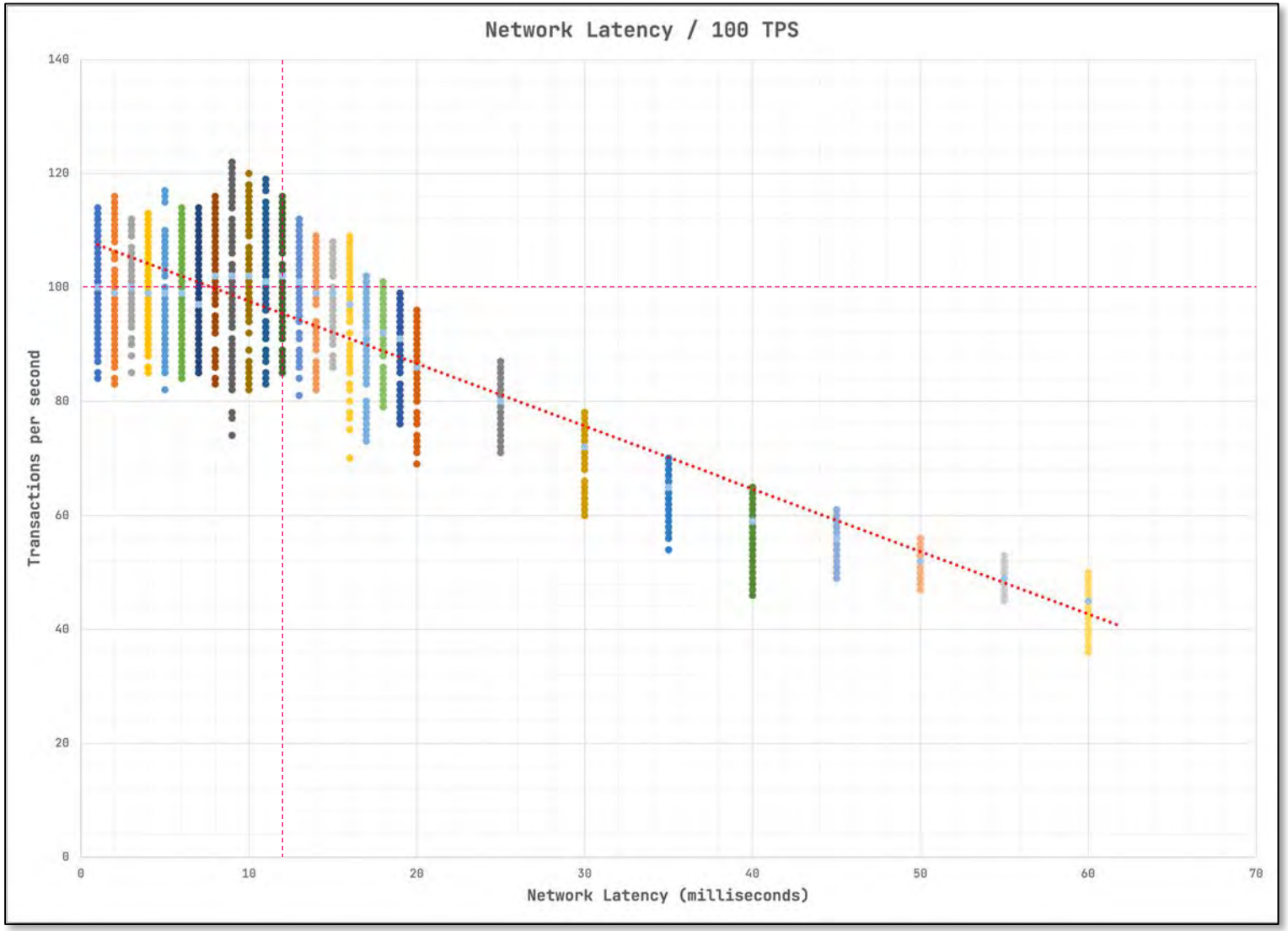
PostgresPro

pgbench

- Тесты: TPC-B
  - **pgbench**
    - --rate=100 {200,400}
    - --jobs=4
    - --clients=4
    - --time=120

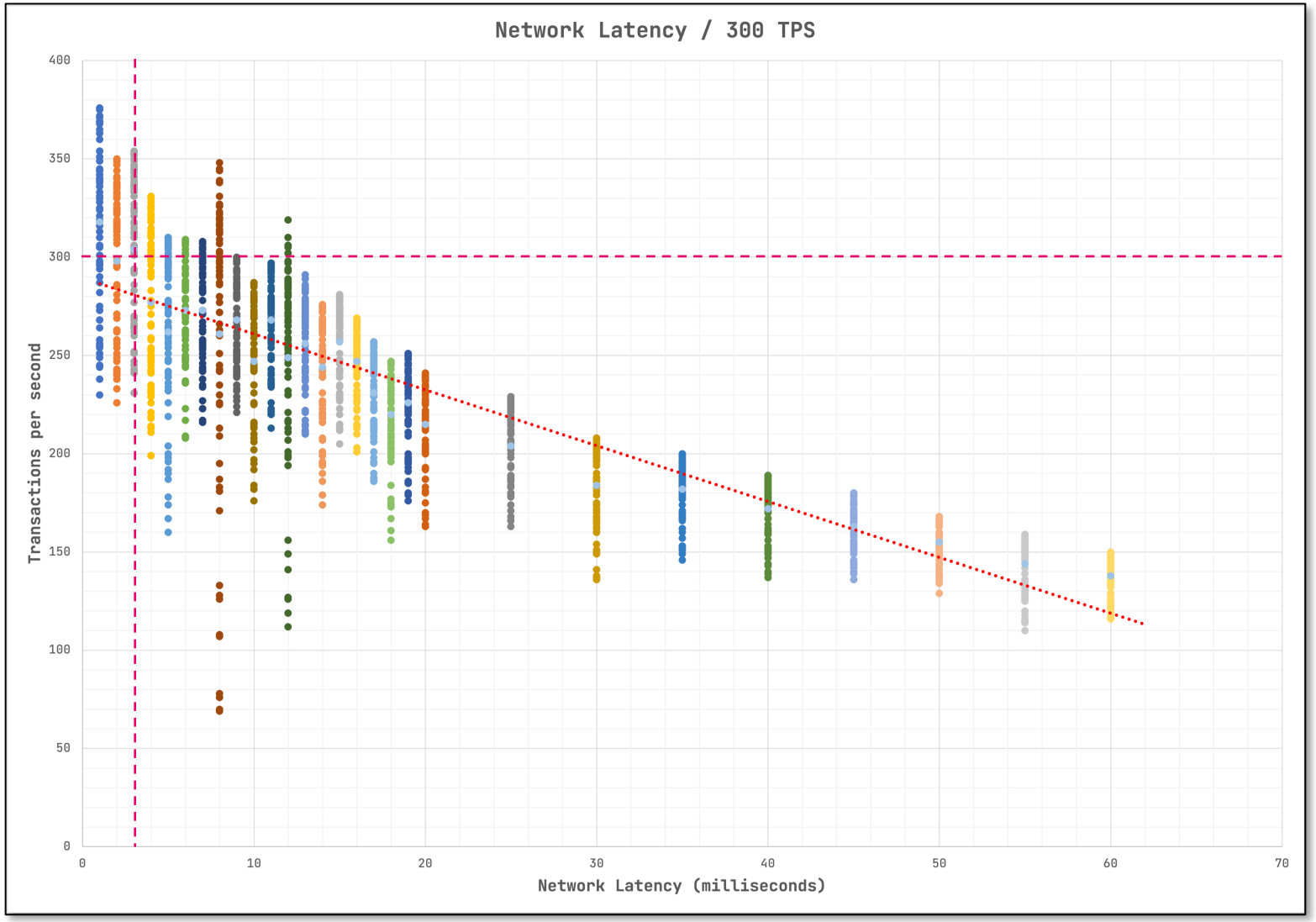
PostgresPro

pgbench



PostgresPro

pgbench



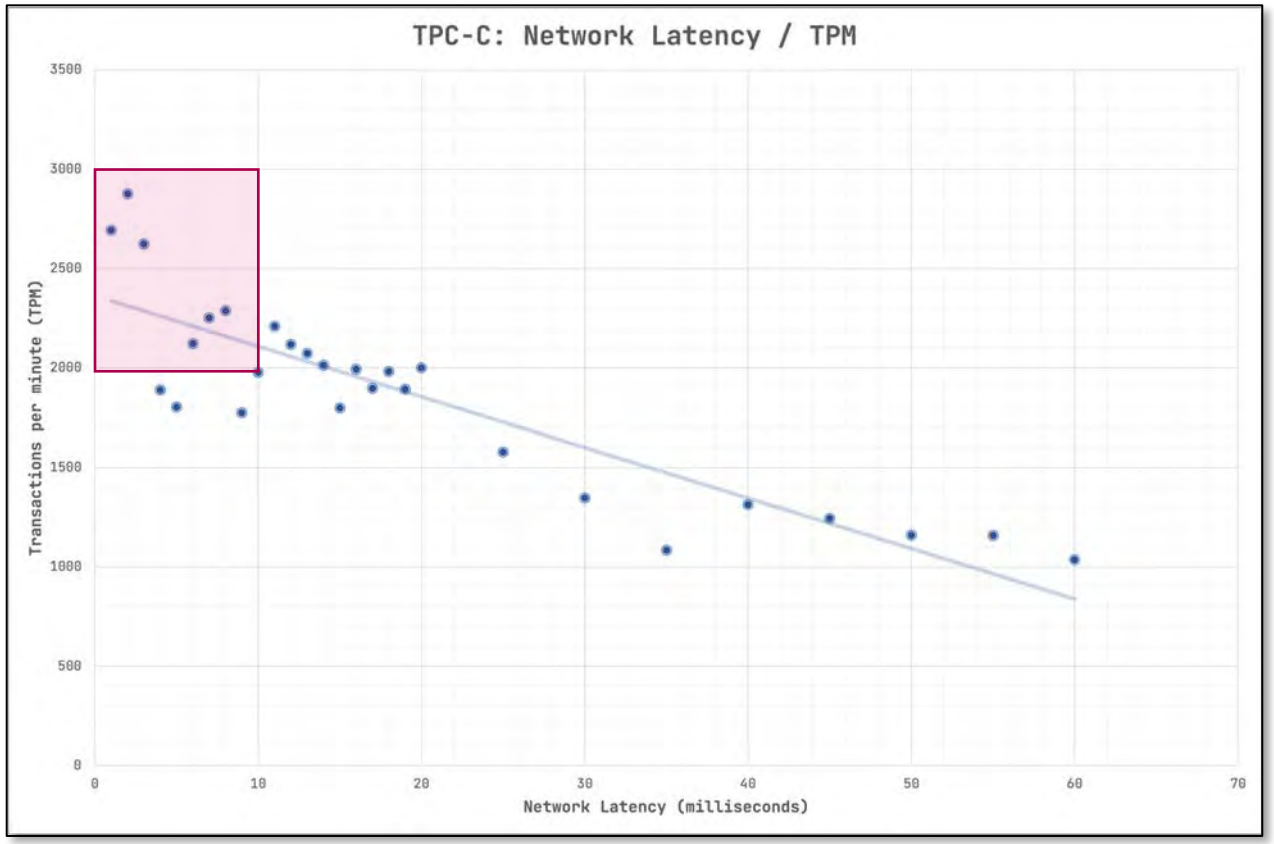
PostgresPro

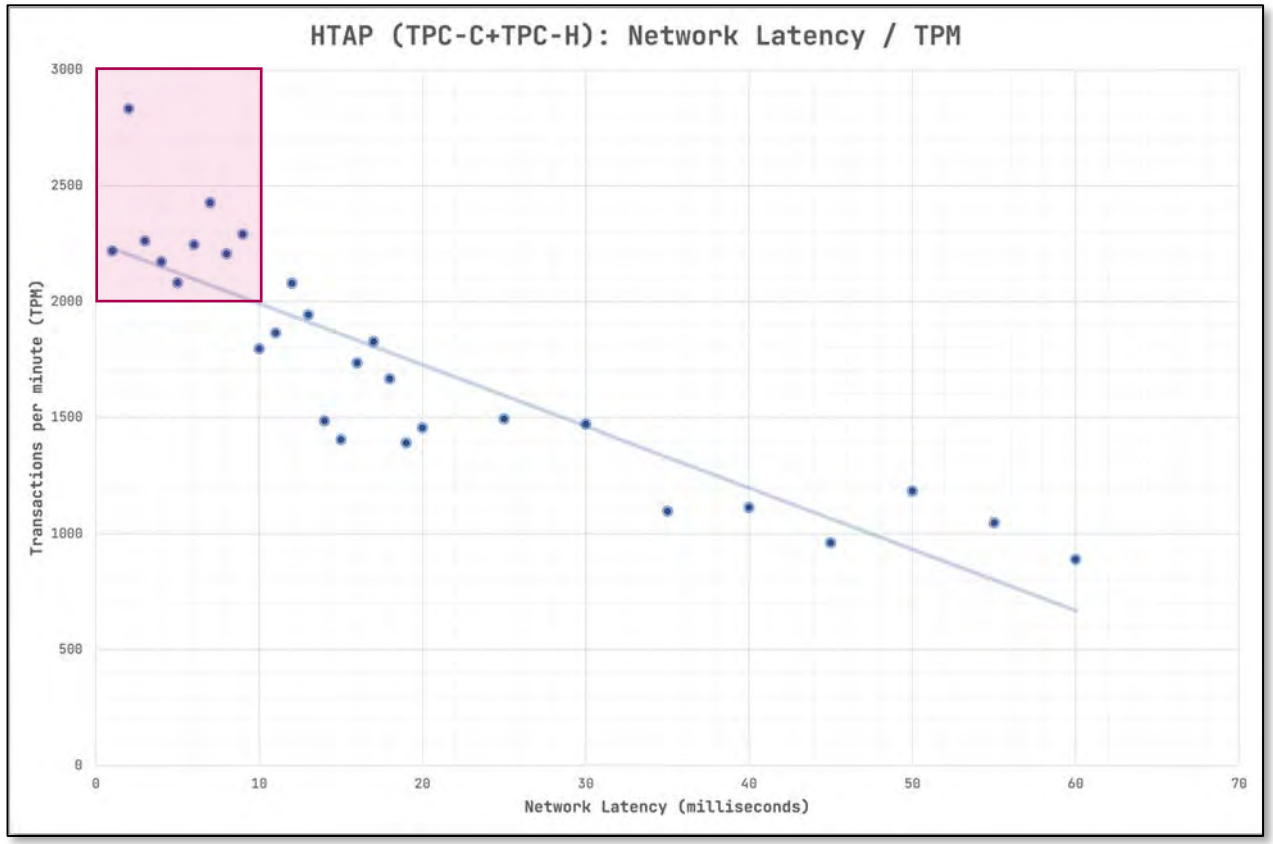
go-tpc

- Тесты: TPC-C, TPC-H, CH
  - **go-tpc**
    - tpcc (OTLP)
    - ch (TPC-C + TPC-H)
      - --threads=4
      - --acThreads=1
      - --time=120s

PostgresPro

tpc-c и tpc-h





PostgresPro

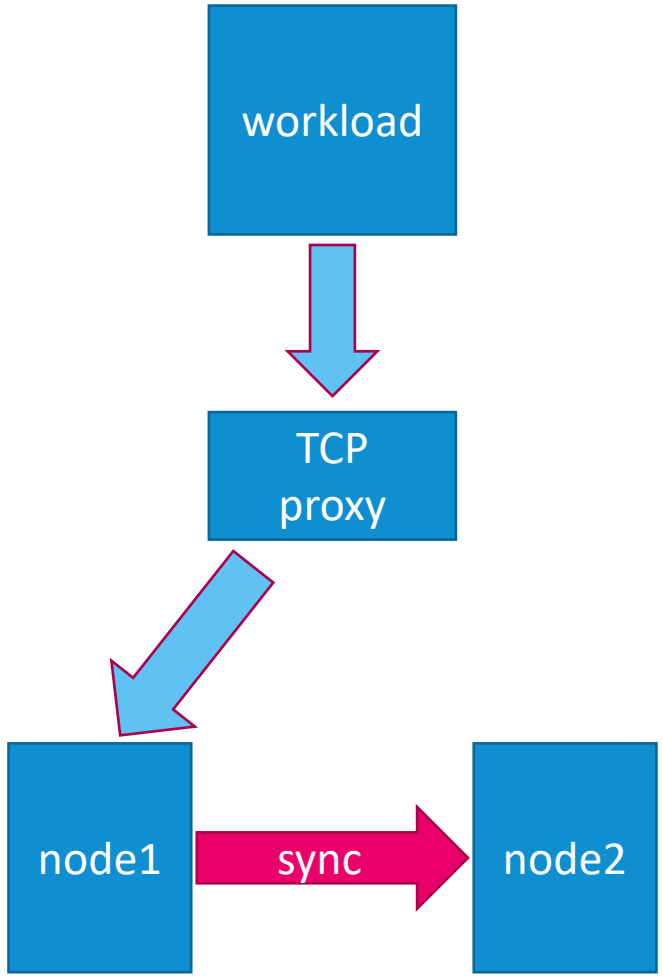
Второй самый часто задаваемый вопрос

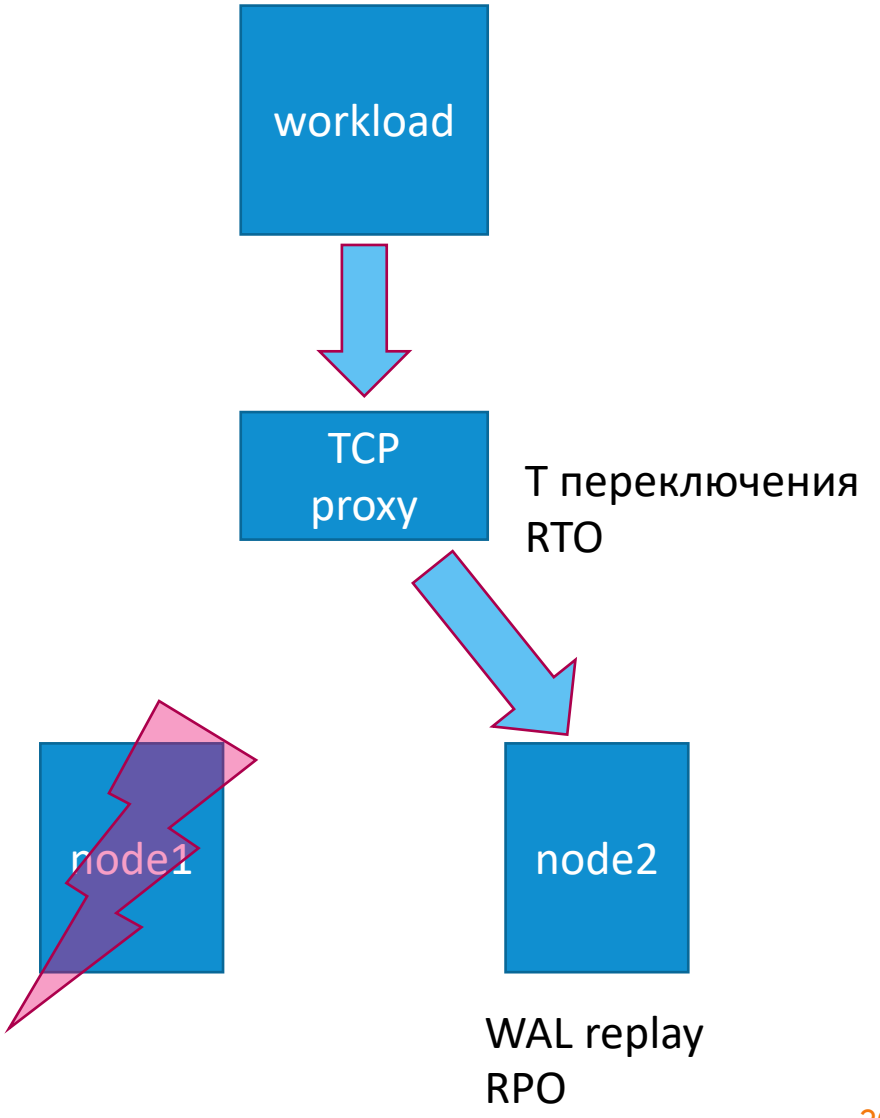
- Какими должны быть значения RTO и RPO?
- А как нам проверить целевые значения метрик?
- ChatGPT ничего толкового не сказал ☹

PostgresPro

RTO и RPO

- RTO
  - Время простоя
- RPO
  - Потеря данных





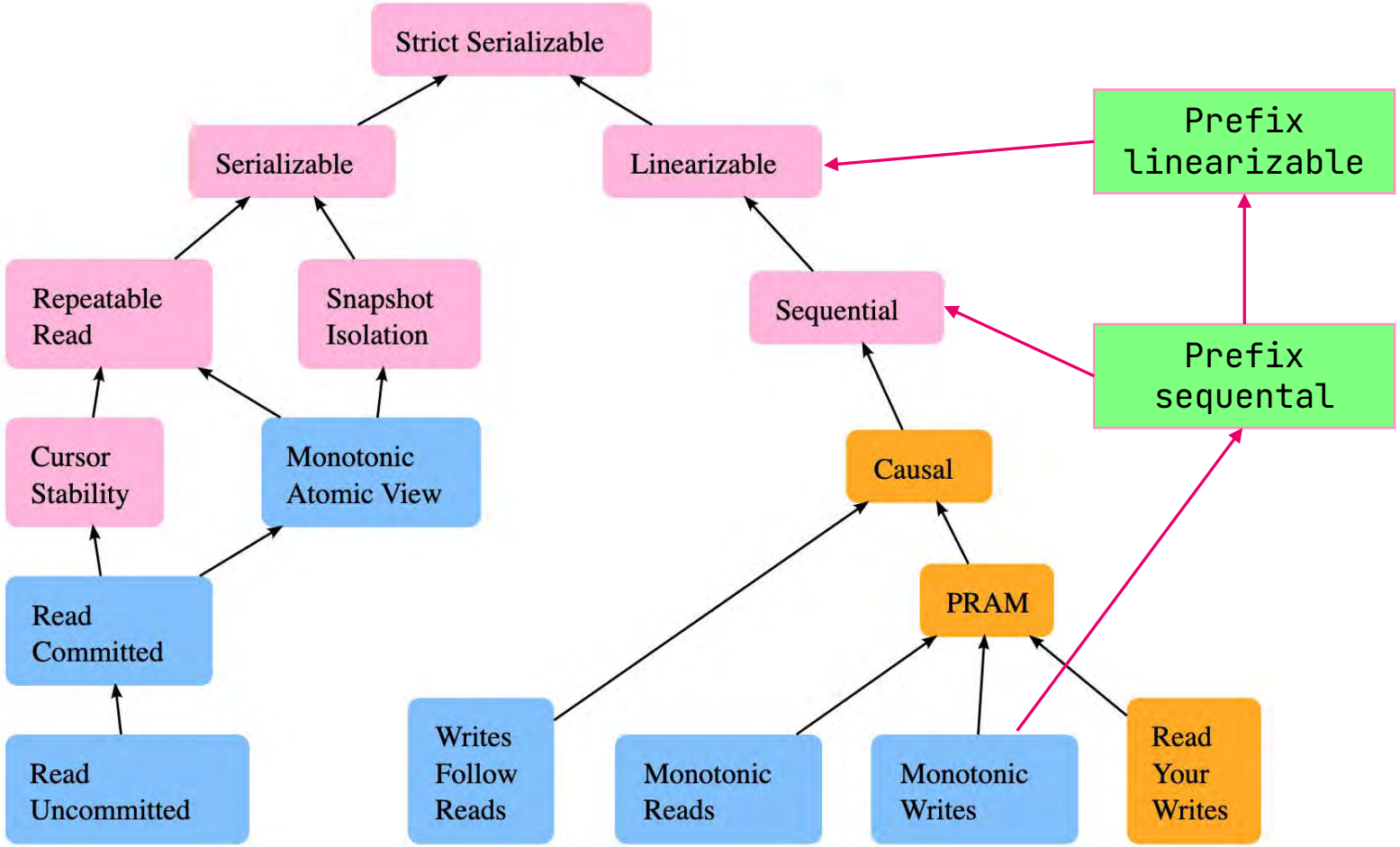
PostgresPro

Stale reads

- Асинхронная реплика
- Синхронная реплика
  - **syncronous_commit**
    - on (fsync на реплике == on)
    - remote_write (fsync на реплике == off)
- Записали данные на мастер
- Тут же пошли на реплику
- Прочитали старые данные

PostgresPro

Модели конситентности

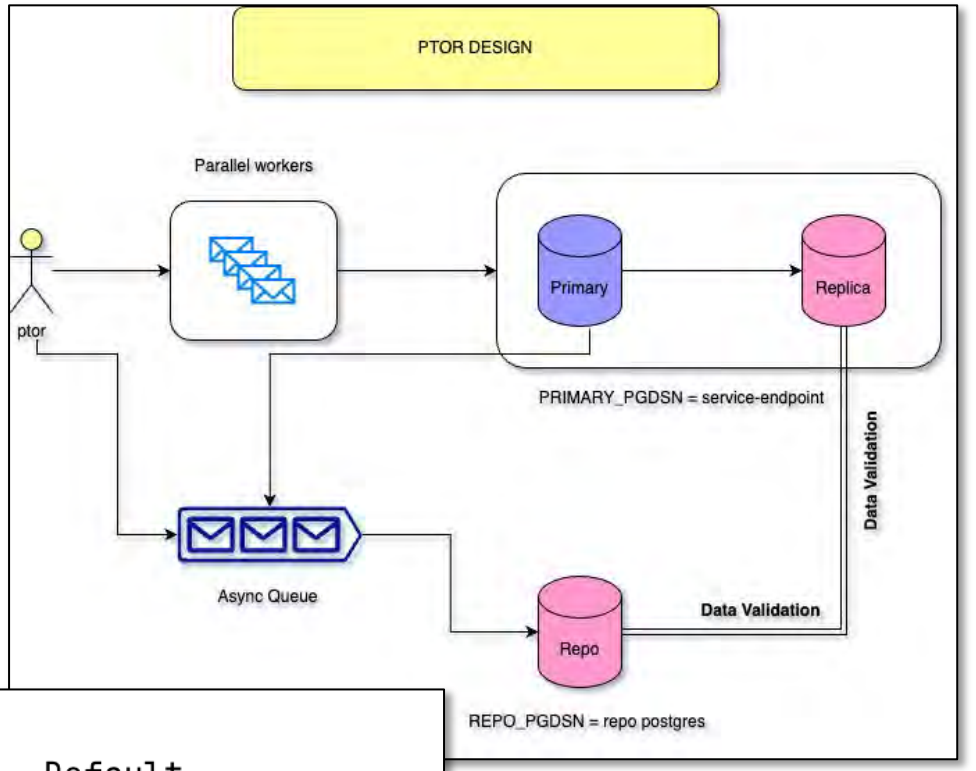


- **Одиночный сервер**
  - serializable
  - linearizable
- **С синхронной репликой** (remote_apply)
  - serializable
  - write linearizable
  - read sequential
- **С асинхронной репликой**
  - Serializable
  - Read-write linearizable
    Read-only prefix linearizable

PostgresPro

ptor

- ptor
  - Автор – Dinesh Kumar
    - PostgreSQL High Performance Cookbook
  - Вычисляет RTO и RPO
  - Автоматизирует процедуру тестирования
  - Табличка воркера:




```
                                Table "ptor.worker"
   Column    |            Type             | Collation | Nullable |            Default
-------------+-----------------------------+-----------+----------+-------------------------------
 id          | bigint                      |           | not null |
 t           | character(8192)             |           |          |
 last_update | timestamp without time zone |           |          | timezone('UTC'::text, now())
```

PostgresPro

Как выполняется замер

- ptor
  - RTO = $T_{восстановления} - T_{сбоя}$
  - RPO
    - Искаженные данные
    - Разница объема хранимых данных
    - Расхождение md5 сумм агрегированных строчек таблиц
    - Может быть выражена в секундах, байтах или строчках
  - SLA
    - $\frac{(количество\ ms\ в\ сутках)-(время\ недоступности\ кластера,\ ms)}{(количество\ ms\ в\ сутках)}$
    - например, 99.9932%

Пример

```
RTO =< 30s
RPO -> 0

ptor --reset

ptor --init --parallel-workers 4

ptor --insert-percent 20
     --update-percent 45
     --delete-percent 35
     --parallel-workers 4
     --rto-conn-timeout 30
     --full-data-validation
```

```
Trying to connect to primary instance ...
Trying to connect to repo instance ...
Checking for ptor schema on primary instance ...
Checking for ptor schema on repo instance ...
Primary latency:  179.292µs
Trying to connect to primary instance ...
Trying to connect to repo instance ...
Checking for ptor schema on primary instance ...
Checking for ptor schema on repo instance ...

Do the failover/switchover from Tessell UI
Got the connectivity error
Checking DNS Availability...
Checking RTO...
Checking RPO...
┌ Summary ──────────────────────────────────────┐
│ SLA                                99.99355   │
│ DNS Available                      4.754347334s │
│ RTO                                5.576582584s │
│ RPO                                0s         │
│ Quick Data Loss Check (bytes)      0          │
└───────────────────────────────────────────────┘

┌ Data Loss ──┐
│ NO DATA LOSS │
└──────────────┘
```

```
Trying to connect to primary instance ...
Trying to connect to repo instance ...
Checking for ptor schema on primary instance ...
Checking for ptor schema on repo instance ...
Primary latency:  93.917µs
Trying to connect to primary instance ...
Trying to connect to repo instance ...
Checking for ptor schema on primary instance ...
Checking for ptor schema on repo instance ...

Do the failover/switchover
Got the connectivity error
Checking Connection Availability...
Checking RTO...
Checking RPO...
Error while comparing data with primary...
┌ Summary ──────────────────────────────────────┐
│ SLA                                99.99183   │
│ Service Available                  4.699875084s │
│ RTO                                7.062736084s │
│ RPO                                0s         │
│ Quick Data Loss Check (bytes)      0          │
└───────────────────────────────────────────────┘

┌ Data Loss ──────┐
│ DATA LOSS DETECTED │
└─────────────────┘
```

PostgresPro

Параметр synchronous_commit

- **syncronous_commit**
  - remote_apply
    - Нет потери данных
  - remote_write
  - on
    - «потеря данных»?
    - переключение до завершения promote

PostgresPro

Задержка задержки

- Что делать, чтобы избежать stale read?
  - Исправить кластерное ПО ☺
  - Поправить приложение
  - Ввести искусственную задержку на TCP proxy
    - HAProxy
      - slowstart 20s maxconn 90 maxqueue 128 weight 100
  - Немного подождать

I have tested the utility on a Patroni cluster...
With **synchronous_commit = remote_write** we are getting message that we have a data loss.

...

This is due to the fact that when failover happened, we *immediately read the data from the new primary*, without taking into account that the promote has not reached the end and not all WALs have been replayed during the promote.

...

I suggest the **--validation-delay** option so that you can make sure the data is OK.

ptor don't know how to wait until the recovery complete (Because, its a kind of app simulator).

This tool, just waits for the new connection after the promote and will calculates the RPO, RTO & SLA.

...

Added this feature **--validation-delay** in the latest release.

Thank you again for validating this tool.

PostgresPro

Ссылки

- owamp
  - https://www.rfc-editor.org/rfc/rfc4656.html
  - https://github.com/perfsonar/owamp
- comcast
  - https://github.com/tylertreat/comcast
- go-tcp
  - https://github.com/pingcap/go-tpc
- ptor
  - https://github.com/dineshkumar02/ptor

PGConf.Russia 2023

PostgresPro

Q&A

Спасибо!