Копия Размещение файлов кластера PostgresPro

Общие соображения

Потоки данных Postgres

В работе сервере Postgres можно выделить следующие относительно независимые потоки данных

Потоки данных	Нагрузка	Надёжность	Управляемость
Постоянные таблицы	Высокая нагрузка по чтению и записи, доступ рандомный	Потеря критична, но может быть восстановлена из WAL	Надо контролировать рост, переполнение приводит к падению сервера, не может быть быстро очищено
Индексы	Высокая нагрузка по чтению и записи, доступ рандомный	Потеря некритична, индекс может быть восстановлен	Надо контролировать рост, переполнение приводит к падению сервера, при критичной ситуации индекс можно временно удалить
Сегменты WAL	Высокая нагрузка по чтению (при архивации и восстановлении) и записи, но доступ последовательный	Потеря критична и необратима для незаархивированных сегментов, нужна максимальная надёжность	Надо контролировать рост, переполнение приводит к падению сервера и потере данных
Временные таблицы	Очень высокая у некоторых типов приложений (напр. 1С)	Временные данные, потеря некртитична	При переполнении может вызвать падение сервера.
Текстовые журналы (log_directory)	При некоторых профилях нагрузки может резко увеличиваться	Потеря некритична для СУБД, но может быть принципиальна для аудита безопасности	При переполнении остановится логирование без влияния на другие системы. Нужна авторотация или автоочистка
Резервные копии и архивы WAL	При включённом сжатии нагрузка умеренная, но чтение может быть критично для выполнения RTO	Потеря критична и невосстановима, нужно максимальное резервирование	При переполнении остановится архивация и начнёт расти каталог неархивированных сегментов WAL. Нужна авторотация или автоочистка

Рекомендации

Для систем, не испытывающих проблем с производительностью, можно дать следующие минимальные рекомендации:Резервные копии и архивы WAL размещать на отдельном сетевом устройстве или на отдельном сервере

1. Текстовые журналы размещать на разделе, не зависимом от $PGDATA, чтобы при переполнении логов не было остановки сервера. Кроме того, вынос данного каталога за пределы $PGDATA снижает объём резервного копирования
2. Если система не находится на круглосуточном мониторинге с дежурным администратором либо отсутствует возможность оперативно добавить место на диск, то надёжным методом является создание на каждом смонтированном разделе большого файла с ненужными данными (dd if=/dev/random bs=512 count=NNNN of=<имя файла>), чтобы потом этот файл можно было безболезненно удалить, освободив место для восстановления СУБД.

Для высоконагруженных систем можно дать следующие рекомендации:

1. Таблицы и индексы размещать на SSD, желательно - в разных табличных пространствах; если при нагрузке узким местом является контроллер или СХД - разносить эти табличные пространства на разные контроллеры или СХД;
2. Сегменты wal размещать на на отдельном дисковом устройстве, возможно, на отдельном физическом контроллере, допустим массив высокопроизводительных HDD, поскольку журналы wal пишутся последовательно, случайный поиск при штатной работе не производится
3. Под временные таблицы выделять табличное пространство, размещённое на tmpfs
4. Текстовые логи перенаправлять на внешнюю систему, например ElasticSearch, для независимого аудита и анализа, и для экономии дискового ввода-вывода.

Для систем с критичными данными можно дать следующие рекомендации:

1. Для таблиц и индексов подходят массивы RAID5, оптимизированные под производительность
2. Для сегментов WAL рекомендуется использовать RAID1 (зеркало) для максимальной защиты данных
3. Резервные копии полезно разделять на оперативные, хранимые в максимально доступном с сервера СУБД месте (локальный диск, по возможности на отдельном контроллере или хренилище) и архивные, размещённые на выделенном сервере с копированием на географически удалённую площадку.

Размещение в файловой системе

Постановка задачи

1. На сервере может работать несколько экземпляров Postgres.
2. Сервер должен поддерживать процедуру мажорных апгрейдов через pg_upgrade, причём как в сценарии с возможностью отката на предыдущую версию, так и в сценарии с быстрым апгрейдом с помощью параметра --link.
3. Используются дополнительные табличные пространства.
4. Каталог log_directory вынесен за пределы $PGDATA.

Предлагаемая структура

<pg_base>/

    <instance1>-<major_version>/

        data/

        ts/

        wal/

    ...

    <instanceN>-<major_version>/

        data/

        ts/

        wal/

    log/

        <instanceN>-<major_version>/

    backup/

        pg_dump/

        pg_probackup/

...где:

<pg_base> - корневой каталог Postgres, см. ниже

<instanceN> - логический код экземпляра, например ERP

<major_version> - мажорная версия Постгрес

Кластер тогда инициализируется с параметрами

```
initdb --pgdata=<pg_base>/ERP-17/data --waldir=<pg_base>/ERP-17/wal --set log_directory='<pg_base>/log/ERP-17/'
```

pg_probackup инициализируется с параметрами

```
pg_probackup init -B <pg_base>/backup/pg_probackup
```

далее все экземпляры добавляются внутри этого каталога

Корневой каталог Postgres

С точки зрения стандарта каталогов линукс правильным корневым каталогом будет /var/lib/pgpro/ или /var/lib/postgres/, в которых по необходимости можно создавать точки монтирования. Преимуществом этого подхода является единый логический каталог кластера, недостатком - некоторая хаотичность возникновения точек монтирования на разных ветках дерева каталогов. Данный вариант может применяться, когда на сервер не является выделенным для СУБД и может выполнять другие задачи.

Для Oracle DBA, работающих с выделенными каталогами СУБД более привычным является выделением корневых каталов /u01.../u0N с отдельными точками монтирования. Этот подход хорош тем, что все точки монтирования отслеживаются на уровне структуры ФС, что облегчает обслуживание и мониторинг. Сложностью является необходимость планировать структуру каталогов заранее, а при добавлении новых точек монтирования - переносить файлы либо создавать символьные ссылки, а в случае табличных пространств - переносить объекты БД. Даный вариант подразумевает, что сервер выделен под Postgres, и ни для какого другого приложение точки монтирования в корне не создаются.

Выбор подходящего варианта оставляется на усмотрения администраторов конкретного сервера.

При любом подходе важно учитывать следующие соображения:

- если на сервере работает антивирус, все каталоги данных Postgres должны быть добавлены в исключения;
- не рекомендуется создавать точки монтирования в каталогах, в которые Postgres непосредственно пишет данные (например $PGDATA), поскольку в точке монтирования могут создаваться служебные файлы и каталоги, такие как lost+found, что может вызвать ошибки в работе СУБД, кроме того, если такой каталог не смонтируется при старте, СУБД начнёт писать файлы в пустой каталог, что может привести к нарушению целостности БД и потере данных
- исключением из предыдущего правила являются каталоги временных табличных пространств , которые по своей природе пересоздаются при каждом запуске операционной системы, поэтому точки монтирования создаются непосредственно в каталоге табличного пространства.

Права

Все каталоги должны принадлежать пользователю postgres, права на них должны быть 700, за исключением каталог текстовых журналов, на которые может быть установлены права 740 в случае, если журналы читаются сторонними системами (аудит или мониторинг)

Точки монтирования

С учётом вышеприведённых рекомендаций рассмотрим несколько вариантов конфигураций.
Важно понимать, что это не исчерпывающий список, а только граничные примеры того, что можно построить.

Слабонагруженная система с размещением файлов в каталоге /var/lib/

Точка монтирования	Назначение	Комментарий
/var/lib/pgpro	Корневой каталог баз постгрес	В такой конфигурации $PGDATA будет в каталоге /var/lib/pgpro/ERP-17/data/ , wal будет в /var/lib/pgpro/ERP-17/wal/, log_directory будет в /var/lib/pgpro/log/ERP-17/
/var/lib/pgpro/log (опционально при большом количестве журналов)	Каталог журнальных файлов	Внутри создаётся каталог инстанса, все журналы пишутся в него.
/var/lib/pgpro/backup	Каталог резервных копий и архивов WAL	На продуктивных серверах всегда должен монтироваться отдельно, даже при небольшой нагрузке. Внутри создаются подкаталоги для различных методов резервного копирования и архивации, например pg_dump, bp_basebackup, pg_probackup, и.т.д.

Средненагруженная система с размещением файлов в каталоге /var/lib/

Точка монтирования	Назначение	Комментарий
/var/lib/pgpro	Корневой каталог баз постгрес	$PGDATA по-прежнему размещается в /var/lib/pgpro/ERP-17/data/, но в ней размещаются только файлы конфигурации, системные объекты (ТП pg_global) и объекты, созданные в ТП по умолчанию (pg_default)
/var/lib/pgpro /ERP-17/ts	Табличные пространства	Необходимо в явном виде создать табличные пространства и контролировать, что объекты БД создаются в них.
/var/lib/pgpro /ERP-17/ts/temp /PG_17_<date>	Каталог временного табличного пространства	Монтируется в tmpfs после создания каталога командой CREATE TABLESPACE <spcname> LOCATION '/var/lib/pgpro /ERP-17/ts/temp/'. При таком размещении не нужно выполнять дополнительные манипуляции при рестарте ОС. Необходимо при интенсивном использовании больших временных таблиц (если при temp_buffers=128MB заметны большие значения temp_files и temp_bytes в pg_stat_database).
/var/lib/pgpro /ERP-17/wal /pg_wal	Каталог файлов WAL	Некоторая тавтология (wal/pg_wal) требуется, чтобы постгрес не писал файлы непосредственно в точку монтирования (см. выше)
/var/lib/pgpro/log	Каталог журнальных файлов	Аналогично предыдущему варианту

/var/lib/pgpro /backup	Каталог резервных копий и архивов WAL	Аналогично предыдущему варианту

Низконагруженная система с точками монтирования в корне

Точка монтирования	Назначение	Комментарий
/u01	Корневой каталог postgres	В такой конфигурации $PGDATA будет в каталоге /u01/ERP-17/data/ , wal будет в /u01/ERP-17/wal/, log_directory будет в /u01/log/ERP-17/, если нет необходимости в подробном журналировании
/u30 (опционально при большом количестве журналов)	Журнальные файлы	Создаётся каталог /u30/log/ERP-17/, он прописывается в log_directory
/u40	Каталог резервных копий и архивов WAL	Создаётся каталог /u40/backup, далее аналогично предыдущим вариантам

Высоконагруженная система с точками монтирования в корне

Точка монтирования	Назначение	Комментарий
/u01	Корневой каталог postgres	$PGDATA размещается в /u01/ERP-17/data/, но в ней размещаются только файлы конфигурации, системные объекты (ТП pg_global) и объекты, созданные в ТП по умолчанию (pg_default)
/u02	Табличные пространства с таблицами	Создаётся каталог /u02/ERP-17/ts/data, в нём создаются ТП для таблиц, дополнительно можно создать /u02/ERP-17/ts/data_cfs для сжатых табличных пространств
/u03	Табличные пространства с индексами	Создаётся каталог /u03/ERP-17/ts/index, в нём создаются ТП для индексов, включая сжатые дополнительно можно создать /u03/ERP-17/ts/index_cfs для сжатых табличных пространств
/u10/pgpro/ERP-17/ts /temp/PG_17_<date>	Каталог временного табличного пространства	Аналогично предыдущему варианту. Нумеруются начиная с u10 для отделения от постоянных табличных пространств
/u20	Файлы WAL	Создаётся каталог /u20/ERP-17/wal/, непосредственно в нём размещаются файлы wal
/u30	Журнальные файлы	Создаётся каталог /u30/log/ERP-17/, он прописывается в log_directory
/u40	Каталог резервных копий и архивов WAL	Создаётся каталог /u40/backup, далее аналогично предыдущим вариантам