



Backups incrementais com Postgres 17

Israel Barth Rubio,
Senior Staff SDE
07/11/2024

Backups incrementais no Postgres 17



Overview

O que é um backup incremental e por que precisamos de suporte no core do Postgres?

- Um backup full copia o banco de dados inteiro
 - Geralmente não é um problema se você tem um banco de dados de até alguns GB
 - Mas e se seu banco de dados tiver algumas centenas de GB ou até mesmo alguns TB?
- Um backup incremental deve copiar somente as partes do bancos de dados que foram alteradas desde o backup anterior
 - O objetivo é tornar os backups menores e mais rápidos
- Várias ferramentas já ofereceram alguma forma de backup incremental para Postgres
 - Barman ao nível de arquivo, pgBackRest ao nível de bloco, entre outras
 - Problemas
 - Cada ferramenta implementa de uma forma diferente
 - Cada uma teve e/ou tem seus próprios bugs
 - É uma implementação complicada, seria melhor ter uma solução comum utilizada e testada por todos



Implementação no Postgres

Fazendo backup corretamente e eficientemente

- Requisitos para um backup incremental
 - Preciso
 - Se incluirmos no backup algo que não foi alterado, nosso backup ficará maior do que o necessário
 - Se não incluirmos no backup algo que foi alterado, nosso backup perderá dados
 - Eficiente
 - Deve detectar as alterações sem muito esforço
 - Fácil implementação
 - Deve aproveitar o máximo de código existente
 - Não depender de funcionalidades do sistema operacional
 - Diferentes sistemas podem ter diferentes comportamentos
 - Depurar código próprio é mais fácil do que depurar algo no sistema operacional



Implementação no Postgres

Utilizando tecnologia validada

- Os arquivos WAL do Postgres contém informações sobre os blocos que foram alterados
 - WALs já são utilizados para vários propósitos no Postgres
 - O Postgres já possui ferramentas para depurar WALs, como **pg_waldump** e **pg_walinspect**
 - No entanto, são arquivos grandes e não podemos utilizá-los diretamente
- Um novo processo é adicionado: WAL summarizer
 - Executa em background
 - Continuamente extrai informações dos arquivos WAL e coloca em arquivos chamados WAL summaries
 - Somente com informações relevantes para backups incrementais
 - Arquivos pequenos e fáceis de se gerar
 - Uma nova ferramenta, chamada **pg_walsummary**, pode ser usada para depurar os WAL summaries



Implementação no Postgres

Detectando alterações

- O backup manifest do backup de referência contém a LSN do backup
 - A partir disso, sabemos quais WAL summaries são necessários
 - Lendo todos os WAL summaries entre a LSN do backup de referência e o momento atual, saberemos o que foi alterado
- O backup manifest também contém a lista de arquivos presentes no backup
 - Se os WAL summaries não mencionam alterações em algum dos arquivos
 - Ou o arquivo é antigo e não foi alterado desde o backup de referência
 - Ou é um arquivo novo introduzido pelo backup atual



Implementação no Postgres

Representando um backup incremental

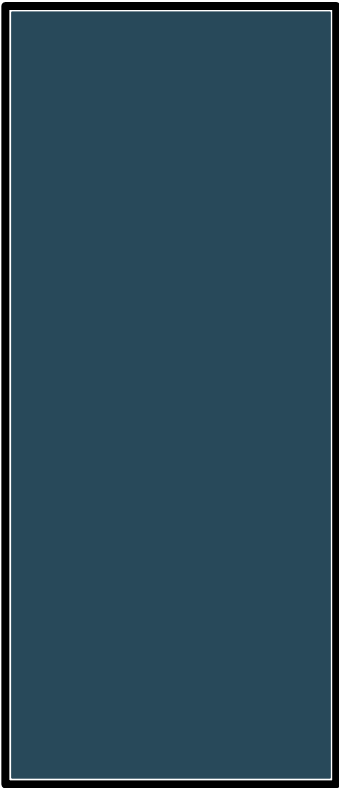
- No backup incremental, algumas tabelas são substituídas por arquivos **INCREMENTAL.nome_original**, que contêm:
 - Alguns metadados
 - Blocos alterados desde o backup de referência
- Os demais arquivos são semelhantes aos de um backup full
- Uma nova ferramenta, chamada **pg_combinebackup**, analisa o backup incremental e a cadeia de backups de referência para criar um backup sintético
 - Um backup full “recriado” a partir da cadeia de backups
 - A partir disso, você precisa fazer o procedimento tradicional de recovery



Visualizando o **pg_combinebackup**

O que ele faz por trás das cortinas?

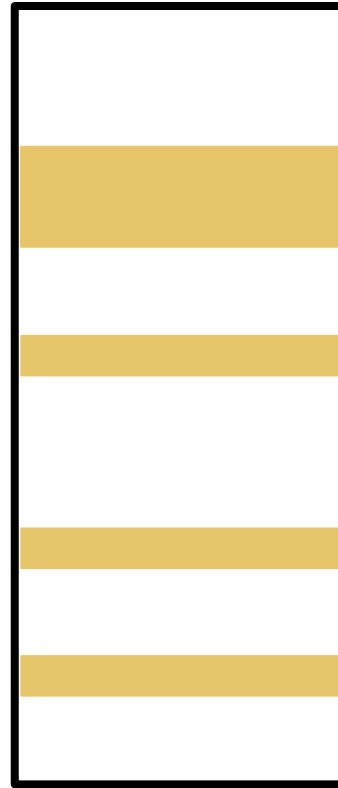
DOMINGO



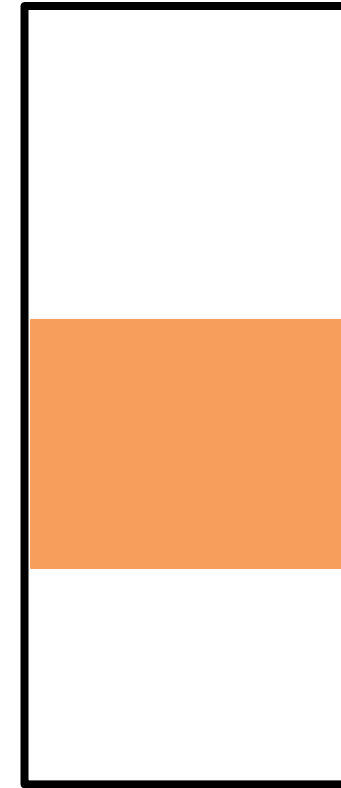
SEGUNDA



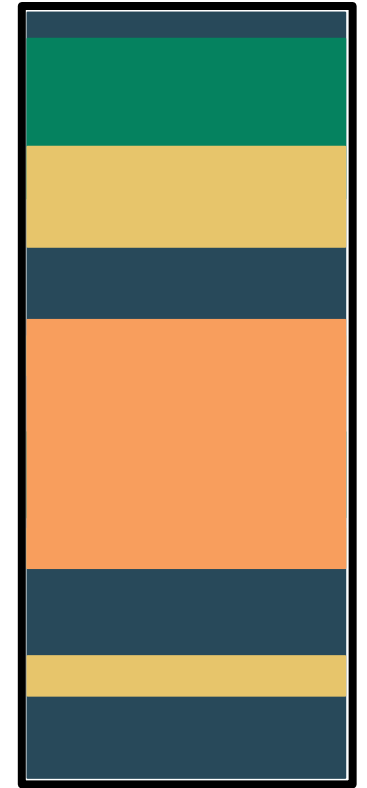
TERÇA



QUARTA



SINTÉTICO



Visualizando o **pg_combinebackup**

O que ele faz por trás das cortinas?



Resumo do mecanismo

Compilando as ideias

- Alterações efetuadas no Postgres são escritas em arquivos WAL
- Os arquivos WALs são sumarizados por um processo separado em arquivos WAL summaries
- O backup manifest do backup de referência é utilizado para identificar os WAL summaries relevantes para o backup incremental
- Com base nas alterações descritas pelos WAL summaries, o backup incremental terá os arquivos originais substituídos por arquivos **INCREMENTAL.nome_original**
- **pg_combinebackup** utiliza os arquivos **INCREMENTAL.nome_original** junto com os backups de referência na cadeia para reconstruir um backup full



Interface do usuário

Como as coisas funcionam na prática

- No **postgresql.conf**, configurar **summarize_wal = on** e talvez alterar **wal_summary_keep_time**
- `pg_basebackup -D domingo`
 - Um backup full de domingo
- `pg_basebackup -D segunda --incremental domingo/backup_manifest`
 - O backup de segunda-feira contém somente o que foi alterado desde domingo
- `pg_basebackup -D terca --incremental domingo/backup_manifest` OU `pg_basebackup -D terca --incremental segunda/backup_manifest`
 - O backup de terça-feira pode conter alterações do que foi alterado desde domingo, ou desde segunda-feira
- `pg_combinebackup domingo segunda terca -o terca_full`
 - Reconstruímos um backup similar a um backup full que tivesse sido feito na terça-feira



Pontos de atenção

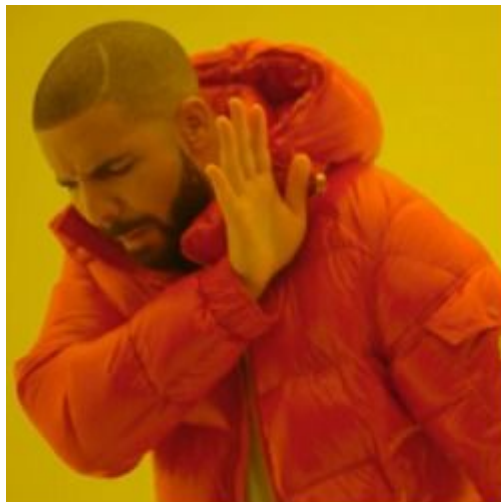
Limitações e pontos a se levar em consideração

- **pg_combinebackup** não suporta backups compactados
 - Se você executou **pg_basebackup** com opção de compactação, você precisará descompactar os backups antes de executar **pg_combinebackup**
- Você pode enfrentar erros caso **data_checksums** tenha tido alterações na cadeia de backups, e o último backup incremental tenha **data_checksums** habilitado
 - O ideal é executar **pg_checksums** para desabilitar e reabilitar checksums no seu banco de dados
- Você precisa manter o seu catálogo de backups e o relacionamento entre eles



Ferramentas

Tá, e agora?



**Scripts
caseiros e
grandes, com
lógica incompleta?**



**Ferramenta
escrita e
suportada
profissionalmente!**



Backups incrementais no Barman



Antes do Postgres 17

Características e limitações

- Backups incrementais a nível de arquivo
 - Requer um sistema de arquivos que suporte hardlinks
 - Utiliza **rsync** para copiar somente arquivos que foram alterados
- Limitações
 - Copia o arquivo inteiro caso haja alterações, mesmo que apenas uma alteração minúscula tenha sido feita
 - Não funciona com backups compactados
 - Não funciona com backups criptografados



Implementação no Barman

Utilizando tecnologia validada

▪ Sobre backups

- Adicionadas referências entre os backups nos metadados para conseguir construir a cadeia de backups
- Adicionada a opção **--incremental** no comando **barman backup** para especificar o backup de referência
- Algumas checks adicionadas para auxiliar o usuário
 - Se **summarize_wal** está habilitado, se houve alterações em **data_checkums**, entre outras

▪ Sobre recovery

- **barman recover** abstrai a execução de **pg_combinebackup**
- Nova opção **--local-staging-path** e configuração **local_staging_path** para especificar onde criar o backup sintético

▪ Sobre políticas de retenção

- Backups incrementais não podem ser restaurados por si só, eles demandam a cadeia de backups de referência
 - Somente backups full contam para retenção
 - Se um backup full é removido, toda a cadeia de backups incrementais abaixo dele também é removida
 - **barman keep** não pode ser executado em backups incrementais



Implementação no Barman

Utilizando tecnologia validada

- Sobre comandos de informação
 - **barman list-backup** inclui um label para identificar o tipo de backup

```
$ barman list-backup postgres
postgres 20241028T145719 - I - Mon Oct 28 14:57:21 2024 - Size: 70.6 MiB - WAL Size: 0 B
postgres 20241028T145619 - F - Mon Oct 28 14:56:25 2024 - Size: 1006.2 MiB - WAL Size: 80.0 MiB
```

- **barman show-backup** mostra o tamanho do backup e uma estimativa do tamanho do backup sintético reconstruído

```
$ barman show-backup postgres 20241028T145719
Backup 20241028T145719:
(...)
Estimated Cluster Size : 1009.2 MiB
(...)
Base backup information:
Backup Method           : postgres
Backup Type             : incremental
Backup Size             : 38.6 MiB (70.6 MiB with WALs)
WAL Size                : 32.0 MiB
Resources saved         : 970.6 MiB (96.18%)
```



Interface do usuário

Como as coisas funcionam na prática

- Configurar **postgresql.conf**, conforme mencionado anteriormente. No **server.conf** do Barman, configurar **backup_method = postgres**
- `barman backup server`
 - Um backup full para o servidor **server**
- `barman backup server --incremental backup_referencia`
 - Um backup incremental para o servidor **server**, com base no backup de referência especificado:
 - ID ou nome do backup de referência, ou
 - Atalhos como **latest** ou **latest-full**
- `barman recover server backup_incremental --local-staging-path caminho_local ...`
 - Restaura o backup **backup_incremental** do servidor **server**
 - Você pode configurar **local_staging_path** no **server.conf**, evitando a necessidade de usar **--local-staging-path**



Pontos de atenção

Limitações e pontos a se levar em consideração

- No momento, **barman recover** não suporta backups incrementais compactados
 - Por este motivo, **barman backup** ainda não permite backups incrementais compactados
- **barman recover** reconstrói o backup sintético localmente e depois o transmite para o destino
- Para evitar uma longa cadeia de backups na execução implícita do **pg_combinebackup**, o ideal é executar um backup full de vez em quando



Possíveis trabalhos futuros

Olhando na bola de cristal

▪ No Postgres

- Suporte a backups compactados no **pg_combinebackup**
 - Pode ser muito conveniente, principalmente para quem trabalha com cloud storage
- Suporte a cloud storage no **pg_combinebackup**
 - No momento, a cadeia de backups deve residir localmente. Seria conveniente ler os backups da cadeia diretamente do cloud storage

▪ No Barman

- Suporte a backups incrementais compactados no **barman recover**
 - Necessário descompactar todos os backups antes de executar **pg_combinebackup**
- Possibilidade de executar **pg_combinebackup** no destino ao invés de somente na origem
- Suporte a um método conhecido como **evergreen**
 - Executa-se somente um backup full. Todos os demais são incrementais, e periodicamente backups sintéticos serão criados:

```
$ pg_combinebackup domingo segunda -o segunda_full  
$ rm -rf domingo segunda && mv -f segunda_full segunda
```



EDB está contratando!



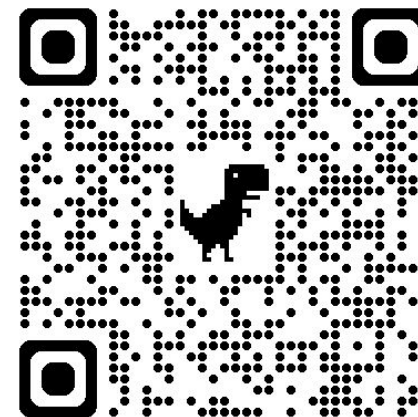
Leia o QR code ou acesse
<https://www.enterprisedb.com/careers>



Obrigado!

israel.barth@enterprisedb.com

[Barman GitHub](#)



[Blog post sobre backups](#)

