Authentic 2 - Bug #69606

Performance: mauvaise requête SQL sur les statistiques du journal

27 septembre 2022 12:41 - Pierre Ducroquet

Statut:	En cours	**Début:**	27 septembre 2022
Priorité:	Normal	**Echéance:**	
Assigné à:	Pierre Ducroquet	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

On voit beaucoup de requêtes de ce genre passer sur la prod, malgré notre machine de course :

```
SELECT MIN("month"), MAX("month") FROM (
    SELECT
        DATE_TRUNC('month', "journal_event"."timestamp") AS "month",
        ("journal_event"."data" ->> 'service_name') AS "service_name",
        COUNT("journal_event"."id") AS "count"
    FROM "journal_event"
    WHERE ("journal_event"."type_id" = 3 AND "journal_event"."timestamp" >= '2019-01-01T00:00:00+0
1:00'::timestamptz)
    GROUP BY DATE_TRUNC('month', "journal_event"."timestamp"), ("journal_event"."data" ->> 'servic
e_name')
) subquery;
```

Cette requête est complètement inefficace (800ms sur glc), surtout en l'absence d'index sur (type_id, timestamp).
Si on ajoute un tel index, la requête ne s'en sort pas mieux à cause de son écriture bien trop complexe.

Première simplification:

```
SELECT MIN("month"), MAX("month") FROM (
    SELECT
        DATE_TRUNC('month', "journal_event"."timestamp") AS "month"
    FROM "journal_event"
    WHERE ("journal_event"."type_id" = 3 AND "journal_event"."timestamp" >= '2019-01-01T00:00:00+0
1:00'::timestamptz)
) subquery;
```

On tombe à 350/400ms, ~250ms avec l'index supplémentaire, ce qui est bien mais pas encore bon.
Le cœur du problème est l'inversion de l'ordre logique : on ne veut pas le minimum/maximum du mois des événements, on veut le mois du minimum/maximum des événements.
Si on écrivait la requête comme suit, on obtiendrait un bien meilleur résultat:

```
SELECT DATE_TRUNC('month', MIN(timestamp)), DATE_TRUNC('month', MAX(timestamp)) FROM "journal_even
t"
    WHERE ("journal_event"."type_id" = 3 AND "journal_event"."timestamp" >= '2019-01-01T00:00:00+0
1:00'::timestamptz);
```

Là, on tombe à moins d'une milliseconde avec l'index supplémentaire (350 sans l'index).

Côté Django, j'ai identifié le code responsable :

```
# src/authentic2/apps/journal/utils.py
class Statistics:
    ...
    def __init__(self, qs, time_interval):
        self.time_interval = time_interval
        self.x_labels = self.build_x_labels(qs)
```

```
        self._x_labels_indexes = {label: i for i, label in enumerate(self.x_labels)}
        self.series = {}
        self.y_labels = []

    ...

    def build_x_labels(self, qs):
        if self.time_interval == 'timestamp':
            return list(qs.distinct().values_list(self.time_interval, flat=True))

        aggregate = qs.aggregate(min=Min(self.time_interval), max=Max(self.time_interval))
```

Je vois deux façons de régler le problème :
1) dans build_x_labels, ~~massacrer~~ altérer l'objet qs pour en extraire les critères WHERE, les paramètres du date_trunc, et construire une nouvelle requête à partir de tout ça,
2) modifier les signatures pour pouvoir ajouter un interval_qs qui retournera le min/max de l'axe X.

Ma requête simplifiée se construit a priori bien en Django:

```
Event.objects.filter(type=3, timestamp__gte=datetime.datetime(2020, 1, 1)).aggregate(min=Trunc(Min
("timestamp"), kind='month'), max=Trunc(Max("timestamp"), kind='month'))
```

Historique

#1 - 27 septembre 2022 13:12 - Pierre Ducroquet

- Fichier 0001-performance-use-a-distinct-query-for-statistics-X-ax.patch ajouté

- Tracker changé de Support à Bug

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Premier patch, il manque au moins la migration, mais je pense que ça devrait faire le taf.

#2 - 27 septembre 2022 15:20 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

- Assigné à mis à Pierre Ducroquet

Il manque un import pour Min/Max (Trunc aussi peut-être).

#3 - 07 mars 2023 14:40 - Robot Gitea

Pierre Ducroquet (pducroquet) a ouvert une pull request sur Gitea concernant cette demande :

- URL : https://git.entrouvert.org/entrouvert/authentic/pulls/15
- Titre : WIP: improve statistics performance (#69606)
- Modifications : https://git.entrouvert.org/entrouvert/authentic/pulls/15/files

Fichiers

0001-performance-use-a-distinct-query-for-statistics-X-ax.patch	4,21 ko 27 septembre 2022	Pierre Ducroquet