Ktimetracker - Functionality #3103

vue des punchs en chevauchement

2011-03-30 13:18 - Antoine Beaupré

Statut:	Closed	**Début:**	2011-03-30
Priorité:	Low	**Echéance:**	
Assigné à:	Yann Rocq	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0.00 heure
Version cible:		**Temps passé:**	0.00 heure
RT ticket:		**Points:**	1

Description

J'ai roulé une requête super lente qui me donne une liste de punchs qui se chevauchent... il y en a quand même pas mal!

```
mysql> SELECT k1.uid, k1.pid, from_unixtime(k1.begin), k1.duration, k2.pid, from_unixtime(k2.begin
), k2.duration FROM kpunch k2 JOIN kpunch k1 ON k1.uid = k2.uid WHERE k1.begin > k2.begin AND k1.b
egin < k2.begin+k2.duration LIMIT 100;
+-----+------+-------------------------+----------+------+-------------------------+----------+
| uid | pid  | from_unixtime(k1.begin) | duration | pid  | from_unixtime(k2.begin) | duration |
+-----+------+-------------------------+----------+------+-------------------------+----------+
|   5 | 6470 | 2009-11-13 14:03:22     |      290 | 6469 | 2009-11-13 14:00:00     |     3600 |
|   5 | 6468 | 2009-11-13 14:01:46     |       96 | 6469 | 2009-11-13 14:00:00     |     3600 |
|   5 | 6469 | 2009-11-13 14:00:00     |     3600 | 6458 | 2009-11-13 13:30:58     |     1848 |
|  42 | 6347 | 2009-11-12 13:02:29     |      730 | 6346 | 2009-11-12 13:00:44     |      705 |
[...]
+-----+------+-------------------------+----------+------+-------------------------+----------+
100 rows in set (29.40 sec)
```

Demandes liées:

Lié à Ktimetracker - Bug #3101: pi -s doesn't change previous punch	**Closed**	**2011-03-30**	**2013-10-18**

Historique

#1 - 2011-03-30 13:36 - Antoine Beaupré

J'ai tenté d'optimiser la requête en ajoutant un index sur begin puis sur duration. Avant l'index:

```
mysql> explain SELECT k1.uid, k1.pid, k1.begin, k1.duration, k2.pid, k2.begin, k2.duration FROM kpunch k2 JOIN
 kpunch k1 ON k1.uid = k2.uid WHERE k1.begin > k2.begin AND k1.begin < k2.begin+k2.duration LIMIT 100;
+----+-------------+-------+------+---------------+------+---------+-------------------------+-------+--------
-----+
| id | select_type | table | type | possible_keys | key  | key_len | ref                     | rows  | Extra
     |
+----+-------------+-------+------+---------------+------+---------+-------------------------+-------+--------
-----+
|  1 | SIMPLE      | k2    | ALL  | uid           | NULL | NULL    | NULL                    | 62851 |
     |
|  1 | SIMPLE      | k1    | ref  | uid           | uid  | 4       | travailkoumbit_1.k2.uid |  1832 | Using w
here |
+----+-------------+-------+------+---------------+------+---------+-------------------------+-------+--------
-----+
2 rows in set (0.00 sec)

mysql> explain SELECT k1.uid, k1.pid, k1.begin, k1.duration, k2.pid, k2.begin, k2.duration FROM kpunch k2 JOIN
 kpunch k1 ON k1.uid = k2.uid WHERE k1.begin > k2.begin AND k1.begin < k2.begin LIMIT 100;
+----+-------------+-------+------+---------------+------+---------+-------------------------+-------+--------
-----+
| id | select_type | table | type | possible_keys | key  | key_len | ref                     | rows  | Extra
     |
+----+-------------+-------+------+---------------+------+---------+-------------------------+-------+--------
-----+
|  1 | SIMPLE      | k2    | ALL  | uid           | NULL | NULL    | NULL                    | 62851 |
     |
|  1 | SIMPLE      | k1    | ref  | uid           | uid  | 4       | travailkoumbit_1.k2.uid |  1832 | Using w
here |
+----+-------------+-------+------+---------------+------+---------+-------------------------+-------+--------
-----+
```

```
2 rows in set (0.01 sec)
```

après l'index:

```
mysql> alter table kpunch add index (begin);
^[[AQuery OK, 62852 rows affected (0.80 sec)
Records: 62852  Duplicates: 0  Warnings: 0

mysql> alter table kpunch add index (duration);
Query OK, 62852 rows affected (1.32 sec)
Records: 62852  Duplicates: 0  Warnings: 0

mysql> explain SELECT k1.uid, k1.pid, k1.begin, k1.duration, k2.pid, k2.begin, k2.duration FROM kpunch k2 JOIN
 kpunch k1 ON k1.uid = k2.uid WHERE k1.begin > k2.begin AND k1.begin < k2.begin LIMIT 100;
+----+-------------+-------+------+---------------+------+---------+-------------------------+-------+--------
-----+
| id | select_type | table | type | possible_keys | key  | key_len | ref                     | rows  | Extra
     |
+----+-------------+-------+------+---------------+------+---------+-------------------------+-------+--------
-----+
|  1 | SIMPLE      | k2    | ALL  | uid,begin     | NULL | NULL    | NULL                    | 62852 |
     |
|  1 | SIMPLE      | k1    | ref  | uid,begin     | uid  | 4       | travailkoumbit_1.k2.uid |  1849 | Using w
here |
+----+-------------+-------+------+---------------+------+---------+-------------------------+-------+--------
-----+
2 rows in set (0.00 sec)
```

Peut-être qu'on peut pas utiliser l'index avec > et <?

#2 - 2011-03-30 13:39 - Yann Rocq

Combien de recouvrement trouves-tu ? Pour moi, la requête devrait plutôt être :

```
SELECT k1.uid, k1.pid, from_unixtime(k1.begin), k1.duration, k2.pid, from_unixtime(k2.begin), k2.duration FROM
 kpunch k2 JOIN kpunch k1 ON k1.uid = k2.uid WHERE k1.begin > k2.begin AND k1.begin < k2.begin+k2.duration-1 L
IMIT 100;
```

(noter le -1 pour exclure les punches qui commencent à la même seconde que le précédent. Es-tu sûr que ta syntaxe tbl, tbl est identique à un inner join ?

#3 - 2011-03-30 14:03 - Antoine Beaupré

J'ignore le nombre total de chevauchements parce que la requête sans le limit dépasse 60 secondes et je dois l'avorter sinon ça bloque toutes les requêtes sur le serveur (!!).

Je pense que -1 est une bonne idée.

Je ne sais pas si foo, bar est identique à un inner join, mais dans la requête je fais un "JOIN" (qui est un LEFT JOIN, je crois?).

#4 - 2011-05-05 14:03 - Guillaume Boudrias

- Assigné à mis à Yann Rocq

#5 - 2011-05-26 16:28 - Guillaume Boudrias

J'ai roulé la requête de Yann (sur une copie locale) et j'ai 1048 résultats. Je ne pense pas qu'en pratique ça soit productif de tenter de corriger tout ça, surtout considérant que beaucoup de punchs sont très vieux et ont été facturés déjà. Peut-être si on ne prenait que les très récents.

Quoiqu'il en soi, Yann suggérait de simplement voir quels utilisateurs ont le plus de punchs en chevauchement pour voir comment ils s'y prennent et donc mieux prévenir le problème.

Peut-être faut-il simplement s'assurer que le problème ne persiste pas. On pourrait retester la logique lorsqu'on modifie un punch.

#6 - 2019-08-01 17:02 - Gabriel Filion

- Statut changé de New à Closed