

データベースの不具合対処について

常三島技術部門
地域協働グループ

飯田 仁 (HIDA Hitoshi)

1. はじめに

筆者は、学生出席管理システムの機能の一部であるカードリーダーを管理している。このカードリーダーは、学生が学生証をかざすと、かざした時刻等を記録するもので出欠確認の基本（打刻）情報となる^[1]。学生出席管理システムでは、この打刻情報を中継サーバに保存し、全学システムである教務情報システムに登録することでシステム連携を行っている。今回、中継サーバのデータベースへの打刻情報登録に時間が掛かる不具合が発生したため、原因調査と対策について述べる。

2. 出席管理システムについて

出席管理システムの概要を図1に示す。カードリーダーに蓄積されている出席情報を、中継装置を介して中継サーバに送信する。中継サーバでは各出席情報をデータベースに取り込む。その後、中継サーバの出席情報をバッチ処理にて教務情報システムに登録する。

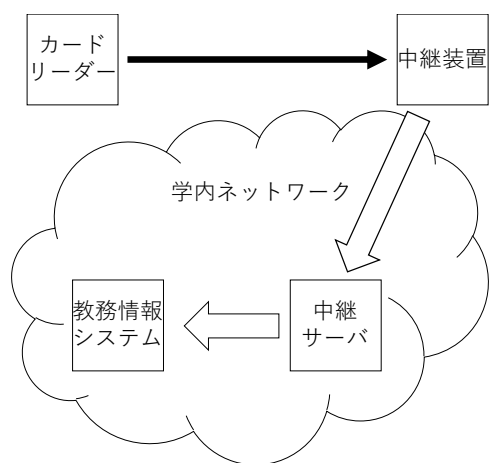


図1 出席管理システムの概要

今回、中継サーバへの登録が時々できないという不具合が発生していた。

中継サーバのデータベーステーブル attendance_data のフィールド詳細を表1に示す。

表1 attendance_data のフィールド詳細

フィールド名	説明
key	主キー
stud_id	学生番号
card_id	学生証（カード）固有番号
att_d	出席（打刻）日付
att_t	出席（打刻）時刻
att_w	出席（打刻）曜日
up_date	登録日時
ip_address	中継装置 IP アドレス
relay_id	中継装置管理番号
node_id	カードリーダー管理番号
room_id	講義室番号
reissue	学生証再発行回数
sync_date	教務情報システム登録日時
read	カードリーダーフラグ
valid	有効フラグ

3. 不具合状況について

具体的な不具合の内容だが、中継サーバへ出席データを登録する際に、登録済み出席データをデータベースから検索し、未登録の出席データのみを登録する処理としている。今回、発生した時々登録できない不具合の状況をより詳細に確認すると、登録の前処理として実行している登録済み出席データの抽出に時間が掛かっていたことが判明した。この抽出に時間が掛かることにより、登録処理全体の設定時間（2[sec]に設定）を超過し登録できていなかった。また、時々発生する状況は、サーバの登録処理時の負荷状況によるものと思われる。

4. 原因と対策について

登録済み出席データの抽出に時間を要している状況が分かったので、インターネット上で検索した結果に基づき^[2-4]不具合時のデータベースの状態を確認した。調査手法の詳細

は参考文献に譲るが最終的には、『登録済み出席データの抽出時の検索SQL文の前に“EXPLAIN ANALYZE”付け実行，結果を確認』を実施した。

詳細な検索SQL文と結果の一部の記載は割愛するが，調査時の実行結果を図2に示す。

```
EXPLAIN ANALYZE "SQL 文"  
QUERY PLAN  
-----  
Unique  
(cost=212052.11..212052.14 rows=1 width=62)  
(actual time=1449.660..1449.660 rows=0 loops=1)  
  中略  
  Rows Removed by Filter: 3445031  
Total runtime: 1449.748 ms  
(13 行)
```

図2 調査時の実行結果

図2の結果より，抽出に1449.748[msec]要していることが分かった。また，図2中の『Rows Removed by Filter: 3445031』は検索SQL文が対象としたレコード(出席データ)数であり，この時の全レコードを対象としている事が分かった。

全レコードを対象としていたことから，検索の方法に問題があると考え，当該データベーステーブルにインデックスを作成し状況の変化を確認することとした。運用開始から，この時までインデックスは未作成であった。

インデックスは上記調査で使用するSQL文のWHERE句に含まれるフィールドを表1から選択するが，WHERE句には次の『stud_id,att_d,room_id,reissue,sync_date』が含まれており，今回は『att_d(打刻日)』のみとした。インデックスは，

『CREATE INDEX at_date_191028_idx ON attendance_data (att_d);』を実行し作成した。

作成後に，上記と同様の調査を実施し，その結果を図3に示す。

インデックスを追加することで検索SQL文が対象とするレコード数(9554)と，抽出に要する時間(10.849[msec])の双方が激減したことが分かる。

```
EXPLAIN ANALYZE "SQL 文"  
QUERY PLAN  
-----  
Unique  
(cost=11.71..11.74 rows=1 width=62)  
(actual time=10.691..10.691 rows=0 loops=1)  
  中略  
  Rows Removed by Filter: 9554  
Total runtime: 10.849 ms  
(15 行)
```

図3 インデックス作成後の実行結果

5. まとめ

インデックスについての知識不足から今回のような不具合の発生を未然に防止することが出来なかったが，良い経験(勉強)となったと思う。ただし，意味のないインデックスを作成してもデータベースのパフォーマンスが低下するので注意が必要である。

なお，今回の不具合は突然発生するようなものではなく，出席データが増加することにより，徐々にパフォーマンスが低下してきたことに気付かなかったことも原因の一つと考える。

6. 今後の課題

今回，インデックスの作成で対応できたが，古い出席データは，サーバのHDD容量も圧迫することから定期的な削除を検討する。また，『att_d』以外のフィールドを含む最適なインデックスを検討したい。

参考文献

- [1] 飯田仁，吉永哲也：“学生証を用いた授業出席管理システムの開発”徳島大学教育研究ジャーナル，第7号，pp.131-137，2010
- [2] http://dqn.sakusakutto.jp/2011/12/postgresql_1.html
- [3] <https://morizyun.github.io/blog/postgresql-index-planner-optimize/index.html>
- [4] <https://language-and-engineering.hatenablog.jp/entry/20110121/p1>