Real-time High Traffic DB Optimization

Overview:

This Client for this project is a leading provider of electronic mobile top-up service. It also specializes in providing bill payment, customer loyalty and customer management services. This project dealt with finding a solution to address the need for taking real time database backups from a high traffic web application. Percona tool was used to achieve the same.

Client details:

Client Name: Confidential | **Location**: Ireland | **Industry**: Retail

Technologies:

MySQL, PHP, Symfony,

Project Description:

The client had a need to archive large log tables with billions of rows of data. The problem was to ensure zero downtime since the production site experienced multiple transactions per day and the only downtime allowed was between 10-15 minutes.

The challenges which stood in the way were

- Limited disk space on server making it impossible to archive table
- Data upto 1 year was required from Transaction and Partner-Transaction tables
- Information for the previous 30 days was needed for the Request-Log table
- There were on an average 30 queries per second for cash transactions. This did not allow downtime of more than 10-15 mins
- 20-30 million records in a table needed to be archived
- MySQL Procedure and Events triggering a low-priority delete of data in chunks could not be used
- Adding sleep time resulted in the process running into an infinite loop

- There was no index on date table to find data older than 1 year
- Too much load on CPU

Solution Steps:

- The team at Mindfire used a toolkit suggested by the client which does not alter with zero downtime and has a proven track record of successful implementation – Percona.
- They purged 30 days of data from the log table using the pt-archiver option.
- Optimized the Log table using pt-online-schema-change module, a feature in the tool, by changing it to InnoDB. This was effectively performing an OPTIMIZE TABLE function in a non-blocking fashion because it was already an InnoDB table.
- Archived 1 year old records from transaction and partner transaction tables using pt-archiver option; with each followed by optimization as mentioned in previous step.
- All operations (archive, delete, optimize) were done in chunks of 10000 records, with average CPU load, and committing each chunk operating in zero downtime. The best part of the execution was that none of the incoming cash transactions failed at any level during the entire process.

Real-time High Traffic DB Optimization

Architecture: