Hospitality Company Using Couchbase NoSQL Database
Gaganpal Kaur Dhillon, HarinderPal Dhillon
Department Of Computer Sc.
University of Bridgeport, Bridgeport, CT

Abstract
NoSql database is designed to support an ever increasing number of users and data and at the same time it maintains the performance efficiently.

We worked with Couchbase distributed NoSql document oriented database that reads and write big data in JSON and GeoJSON objects. Couchbase Server delivers high throughput required for real time big data applications. Spatial views support the GeoJSON format and allow bounding-box queries on complex geometries that are represented as points, line-strings, or polygons.

In Hospitality Company, spatial and geospatial views allow you to extract specific fields and information from Json data and create an index.

Why Couchbase??
You can easily manage your Couchbase cluster--adding, deleting, and fetching documents from the Couchbase Management GUI, for which MongoDB has no counterpart. Couchbase Server supports agile development with a flexible data model and a powerful query language, scalable deployment with a single node type and topology-aware clients.

- Deploy as a distributed cache for low latency reads
- Deploy as a key/value store for high performance reads and writes
- Deploy as a document database for powerful querying and lightweight analytics

Viber replaced MongoDB with 50% fewer Couchbase servers. AT&T, Comcast, eBay, GE, LinkedIn, Marriott, PayPal, Tesco, Verizon, VISA, Wells Fargo and many other customers are using Couchbase.

Implementation Features
We have tried to implement the features of Couchbase in our Hospitality Company using ASP.net Web Forms and N1QL for querying the JSON data. Some of the key features of our project are:

- Storing the data in JSON format using serialization and deserialization.
- Search various hotels and restaurants filtered by city/state/price from within the document attribute.
- Stores the user personal info, booking record in single document for high performance.
- Stores the user search history in order to provide recommendations
- Retrieve data faster using spatial and Geospatial views.
- Various queries using map and reduce functions.
- Sample query and its low elapsed time is shown below:

<table>
<thead>
<tr>
<th>Couchbase Physical Data Model:</th>
<th>Relational Databases</th>
<th>Couchbase Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Documentstorageformat</td>
<td>Documentstorageformat</td>
</tr>
<tr>
<td>Table</td>
<td>Documentstorageformat</td>
<td>Documentstorageformat</td>
</tr>
<tr>
<td>Index</td>
<td>Indexing (value in key document)</td>
<td>Indexing (value in key document)</td>
</tr>
<tr>
<td>Structure</td>
<td>Structure</td>
<td>Structure</td>
</tr>
<tr>
<td>Metadata</td>
<td>Metadata</td>
<td>Metadata</td>
</tr>
<tr>
<td>Index</td>
<td>Index</td>
<td>Index</td>
</tr>
</tbody>
</table>

Couchbase stores the data in JSON format. The example below shows the data stored in relational database and the couchbase server.

Architecture
Couchbase Server Architecture:
Couchbase Server is built on a memory-first architecture and managing memory effectively is central to achieving high performance and scalability. Cluster Manager runs on all the nodes of the cluster and orchestrates cluster wide operations.

Conclusion
Database management system are useful when working with a huge quantity of data. System are distributed, non-relational database.

The main target is the consistency and the availability of the up to date data for the consumers which could be easily met using couchbase server.

Acknowledgement
Special thanks to Professor Jeongkyu Lee for his indispensable help. We would also like to thank Computer Science Deptt. For providing the wonderful opportunity to research on NoSql databases.