Intelligent Data Clustering for Cold-Data Storage in Main-Memory Databases
Atreyee Maiti (amaiti@andrew.cmu.edu), Andrew Pavlo (pavlo@cs.cmu.edu)
Carnegie Mellon University

Background: Anti-Caching
- Main-memory databases optimal for OLTP workloads
- Restricted by the requirement that entire dataset fit in RAM
- Techniques like Anti-caching and OS paging help in overcoming this
- The idea is to identify the cold data and evict it out to secondary storage as the data size grows

Modified block structure

<table>
<thead>
<tr>
<th>Number of Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table #1 ID, Number of Tuples</td>
</tr>
<tr>
<td>.....</td>
</tr>
<tr>
<td>Table #N ID, Number of Tuples</td>
</tr>
<tr>
<td>.....</td>
</tr>
<tr>
<td>Table 1 tuples</td>
</tr>
<tr>
<td>.....</td>
</tr>
<tr>
<td>Table N tuples</td>
</tr>
</tbody>
</table>

Solution: Clustering
- Anti-caching does not leverage correlation among tables.
- Naive table to block mapping
- A transaction may incur multiple disk reads and multi restarts for correlated evicted tables

Benefits
- Reduced disk reads
- Reduced cold data tracking
- Possibility of reduction in tuple offset pointer size

Experimental results:
Implemented for the H-Store database.
H-Store (http://hstore.cs.brown.edu/) is a main-memory distributed database with support for anti-caching.

Benchmark: A reddit clone (custom data) with entities: Articles, Comments, Users

Workload:
1) Get Article (35%)
2) Get Comments (35%)
3) Add a comment (20%)
4) Update user info (10%)