COS 316
Precept:
Cache Eviction (Replacement)
Overview of Web Caching

• Basic idea:
  • Bring objects “closer” to clients
• Three primary features:
  • Reduce network bandwidth
  • Reduce client-perceived delays
  • Reduce load on server
• Cache Replacement Strategy
  • When a cache becomes full, which object should be **evicted/replaced**?
Cache Eviction Algorithms

- High level
  - Client requests a new object
  - If object is in cache, return the object
  - If object is not in cache:
    - Get object from server/provider and return the object
    - Cache full:
      - Identify an object in cache to evict
      - Evict the object in the cache
      - Replace with new object
    - Cache not full:
      - Admit the new object to the cache
Cache Eviction Algorithms

- Least recently used (LRU): Evict the object from the cache whose last request is the oldest

- First-in, First-out (FIFO): Evict the object from the cache that has been in the cache the longest

- Many others...
LRU

Cache capacity = 50
Cache size = 45

id:  8
size: 10
request: __:__
admit:  __:__

Current time: 14:00

id:  8
size: 10
request: 13:00
admit:   11:00

id:  3
size: 10
request: 13:45
admit:   13:45

id:  1
size:  3
request: 15:01
admit:   12:01

id:  4
size:  5
request: 11:53
admit:   11:33

id:  11
size:  8
request: 11:30
admit:   11:30

id:  7
size: 17
request: 13:30
admit:   13:30

id:  8
size: 10
request: 14:00
admit:   14:00

id:  7
size: 17
request: 13:30
admit:   13:30

id:  11
size:  8
request: 11:30
admit:   11:30

id:  1
size:  3
request: 15:01
admit:   12:01
<table>
<thead>
<tr>
<th>id</th>
<th>size</th>
<th>request</th>
<th>admit</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>2</td>
<td>13:00</td>
<td>11:00</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>13:45</td>
<td>13:45</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>15:01</td>
<td>12:01</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>11:53</td>
<td>11:33</td>
</tr>
<tr>
<td>7</td>
<td>17</td>
<td>11:30</td>
<td>11:30</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>11:30</td>
<td>11:30</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td><em><strong>:</strong></em></td>
<td><em><strong>:</strong></em></td>
</tr>
</tbody>
</table>

**Cache capacity = 50**
**Cache size = 45**

Current time: 14:00
LRU

Cache capacity = 50
Cache size = 45

id: 6
size: 2
request: 13:00
admit: 11:00

id: 3
size: 10
request: 13:45
admit: 13:45

id: 1
size: 3
request: 15:01
admit: 12:01

id: 11
size: 8
request: 11:30
admit: 11:30

id: 7
size: 17
request: 13:30
admit: 13:30

Current time: 14:00

id: 8
size: 10
request: __:__
admit: __:__

id: 6
size: 2
request: 13:00
admit: 11:00

id: 3
size: 10
request: 13:45
admit: 13:45

id: 1
size: 3
request: 15:01
admit: 12:01

id: 11
size: 8
request: 11:30
admit: 11:30

id: 7
size: 17
request: 13:30
admit: 13:30

Current time: 14:00

id: 8
size: 10
request: 14:00
admit: 14:00

id: 6
size: 2
request: 13:00
admit: 11:00

id: 3
size: 10
request: 13:45
admit: 13:45

id: 1
size: 3
request: 15:01
admit: 12:01

id: 11
size: 8
request: 11:30
admit: 11:30

id: 7
size: 17
request: 13:30
admit: 13:30

id: 8
size: 10
request: __:__
admit: __:__

id: 4
size: 5
request: 11:53
admit: 11:33

id: 7
size: 17
request: 13:30
admit: 13:30

id: 8
size: 10
request: 14:00
admit: 14:00

id: 4
size: 5
request: 11:53
admit: 11:33

Cache capacity = 50
Cache size = 47
FIFO

Cache capacity = 55
Cache size = 45

id: 6
size: 2
request: 13:00
admit: 11:00

id: 3
size: 10
request: 13:45
admit: 13:45

id: 1
size: 3
request: 15:01
admit: 12:01

id: 11
size: 8
request: 11:30
admit: 11:30

id: 7
size: 17
request: 13:30
admit: 13:30

id: 11
size: 8
request: 14:00
admit: 14:00

id: 4
size: 5
request: 11:53
admit: 11:33

id: 3
size: 10
request: 13:45
admit: 13:45

id: 4
size: 5
request: 11:53
admit: 11:33

Current time: 14:00

Cache capacity = 55
Cache size = 45

id: 8
size: 10
request: __:__
admit: __:__

id: 3
size: 10
request: 13:45
admit: 13:45

id: 1
size: 3
request: 15:01
admit: 12:01

id: 4
size: 5
request: 11:53
admit: 11:33

Current time: 14:00

Cache capacity = 55
Cache size = 53

id: 8
size: 10
request: 14:00
admit: 14:00

id: 3
size: 10
request: 13:45
admit: 13:45

id: 1
size: 3
request: 15:01
admit: 12:01

id: 4
size: 5
request: 11:53
admit: 11:33

id: 11
size: 8
request: 14:00
admit: 14:00

id: 7
size: 17
request: 13:30
admit: 13:30

id: 7
size: 17
request: 13:30
admit: 13:30

id: 11
size: 8
request: __:__
admit: __:__

id: 7
size: 17
request: __:__
admit: __:__
### FIFO

<table>
<thead>
<tr>
<th>ID</th>
<th>Size</th>
<th>Request</th>
<th>Admit</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>2</td>
<td>13:00</td>
<td>11:00</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>13:45</td>
<td>13:45</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>15:01</td>
<td>12:01</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>11:53</td>
<td>11:33</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>11:30</td>
<td>11:30</td>
</tr>
<tr>
<td>7</td>
<td>17</td>
<td>11:30</td>
<td>11:30</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>14:00</td>
<td>14:00</td>
</tr>
</tbody>
</table>

**Current time:** 14:00

**Cache capacity:** 55
**Cache size:** 45
Experiments

> cd <COS316-Public repo>  # directory containing Vagrantfile
> git pull # update with precept5 code and data
> vagrant up ## if instance is not already started
> vagrant ssh
> cd precepts/precept5/webcachesim-master
> make
Trace File Form

- Request traces must be given in a space-separated format with three columns
- time - long long int
- id - long long int, used to uniquely identify objects
- size should be a long long int, object's size in bytes

- Example

<table>
<thead>
<tr>
<th>time</th>
<th>id</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>64</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>120</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>120</td>
</tr>
</tbody>
</table>

- See test.tr
Using the Simulator*

> ./webcachesim test.tr LRU 1000

LRU: 1000 bytes, 10492 reqs, 8495 hits, 81 hits/reqs(%)

> ./webcachesim test.tr FIFO 1000

FIFO: 1000 bytes, 10492 reqs, 8206 hits, 78 hits/reqs(%)

* Derived from https://github.com/dasebe/webcachesim
Experiments

• Trace data from a production CDN
  • cd1-10M.tr *
  • 10 million requests /
    Object sizes from 10 byte
    to .7GB
• LIFO and FIFO
• Vary cache sizes
  • 16000000
  • 32000000
  • 64000000
  • 128000000
  • 256000000

• Create a Google Sheet
• Three columns
• SIZE LRU FIFO
• Copy results accordingly
• Select three columns to create line chart

* Practical Bounds on Optimal Caching with Variable Object Sizes Daniel S. Berger, Nathan Beckmann, Mor Harchol-Balter. ACM SIGMETRICS, June 2018