Context on Bayou: Disconnected Nodes [Stoica]

Early days: nodes always on when not crashed

- Network bandwidth always plentiful.
- Never needed to work on a disconnected node

Now: nodes detach then reconnect elsewhere

- Even when attached, bandwidth is variable
- Reconnection elsewhere means often talking to different replica
- Work done on detached nodes
Bayou

- "[R]eplicated, [eventually] consistent storage system designed for ... portable machines with less-than-ideal network connectivity."

- System developed at PARC in the mid-90’s

- First coherent attempt to fully address the problem of disconnected operation
Bayou

What is it?

Weakly consistent, replicated storage system

Goals:

Maximize availability, *support offline collaboration*
Minimize network communication
Agree on all values (eventually)
Bayou Update Protocol: Review from Class

- Client sends update to a server
- Updates uniquely identified by:
  \(<\text{Commit Sequence Number (CSN)}, \text{Local Timestamp}, \text{Node ID}\>\)
- Updates are either committed or tentative
  - CSNs increase monotonically
  - Tentative updates have commit-stamp = $\infty$
- Only Primary server can commit updates
  - Allocates CSN in monotonically increasing order
  - CSN is different from time-stamp
Anti-Entropy Exchange

- Each server keeps a version vector:
  - $R.V[X]$ is the latest timestamp from server $X$ that server $R$ has seen

- When two servers connect, exchanging the version vectors allows them to identify the missing updates

- These updates are exchanged in the order of the logs, so that if the connection is dropped the crucial monotonicity property still holds
  - If a server $X$ has an update accepted by server $Y$, server $X$ has all previous updates accepted by that server
Bayou Writes

value1 : value2 : value3 denotes
Commit Sequence Number (CSN) : Local Timestamp : Node ID

W(X, 4) Client 1
Bayou Writes

value1 : value2 : value3 denotes
Commit Sequence Number (CSN) : Local Timestamp : Node ID
Bayou Writes

value1 : value2 : value3 denotes
Commit Sequence Number (CSN) : Local Timestamp : Node ID

Client 1

W(Y, 8)

Client 2

W(X, 3)
Bayou Writes

$value_1 : value_2 : value_3$ denotes Commit Sequence Number (CSN) : Local Timestamp : Node ID
Bayou Writes

value1 : value2 : value3 denotes Commit Sequence Number (CSN) : Local Timestamp : Node ID

A

versions

∞:7:P W(X,3)
∞:12:A W(Y,4)

versions

P: 0
A: 12
B: 0

Primary

versions

∞:1:P W(X,4)
∞:7:P W(Y,8)

versions

P: 7
A: 0
B: 0

B

versions

∞:5:B W(Z,8)

versions

P: 0
A: 0
B: 5
Bayou Anti-Entropy (Sync)

Anti-entropy Session
A & B

A

versions

P: 7
A: 0
B: 0

B

versions

P: 0
A: 12
B: 0

A

versions

P: 0
A: 0
B: 5

B

versions

P: 0
A: 0
B: 5

∞:7:A W(X,3)
∞:12:A W(Y,4)

∞:5:B W(Z,8)

∞:1:P W(X,4)
∞:7:P W(Y,8)

∞:5:B W(Z,8)
∞:7:A W(X,3)
∞:12:A W(Y,4)
Bayou Anti-Entropy (Sync)
Bayou Commit

Primary commits its entries
Bayou Write

Write after anti-entropy session
Write timestamp = max(clock, max(TS)+1)

Client 1
D(Y)
Bayou Anti-Entropy (Sync)

Anti-entropy Session
P & B

P

A: 0
B: 0

Versions

1:1:P  \text{w}(X,4)
2:7:P  \text{w}(Y,8)

\infty:5:B  \text{w}(Z,8)
\infty:7:A  \text{w}(X,3)
\infty:12:A  \text{w}(Y,4)
\infty:13:B  \text{D}(Y)

\text{A: 12}
\text{B: 13}

P

A: 0
B: 0

Versions

1:1:P  \text{w}(X,4)
2:7:P  \text{w}(Y,8)

\text{P: 0}
\text{A: 12}
\text{B: 13}

\infty:5:B  \text{w}(Z,8)
\infty:7:A  \text{w}(X,3)
\infty:12:A  \text{w}(Y,4)
\infty:13:B  \text{D}(Y)

\text{P: 0}
\text{A: 12}
\text{B: 13}

\infty:5:B  \text{w}(Z,8)
\infty:7:A  \text{w}(X,3)
\infty:12:A  \text{w}(Y,4)
\infty:13:B  \text{D}(Y)
Bayou Anti-Entropy

Anti-entropy Session
P & B
Primary respects causality

A
Versions
∞:5:B W(Z,8)
∞:7:A W(X,3)
∞:12:A W(Y,4)

P: 0
A: 12
B: 5

B
Versions
∞:5:B W(Z,8)
∞:7:A W(X,3)
∞:12:A W(Y,4)
∞:13:B D(Y)

P: 7
A: 12
B: 13
Bayou Commit

Primary commits Its entries
Bayou

After a number of commits and anti-entropy sessions (without further writes), all nodes converge on same state.
Bayou (review from class)

1. **Eventual consistency**: if updates stop, all replicas eventually the same view.
2. **Update functions** for automatic app-driven conflict resolution.
3. **Ordered update log** is the real truth, not the DB.
4. Use **Lamport clocks**: eventual consistency that respects causality.
Context for Chord: Key Value Stores

Amazon:
- Key: customerID
- Value: customer profile (e.g., buying history, credit card, ..)

Facebook, Twitter:
- Key: UserID
- Value: user profile (e.g., posting history, photos, friends, ...)

iCloud/iTunes:
- Key: Movie/song name
- Value: Movie, Song

Distributed file systems
- Key: Block ID
- Value: Block

Credit: Ion Stoica’s slide deck
Context for Chord: Key Value Stores

Key Value Store

Also called a Distributed Hash Table (DHT)
Main idea: partition set of key-values across many machines

Credit: Ion Stoica’s slide deck
Chord

- Chord: “a distributed lookup protocol” for a peer-to-peer distributed hash table [Stoica ’01]
- Consistent hashing for partitioning key space + lookup
Identifiers in Chord

- Key identifier = SHA1(key) mod $2^m$
- Node identifier = SHA1(IP address) mod $2^m$
- Both are uniformly distributed in the same identifier space
- The identifier length, $m$, must be large enough to make the probability of two nodes or keys hashing to the same identifier negligible (e.g. $m = 160$)

How do we map key IDs to node IDs?
Consistent Hashing

● A node owns the preceding key range, including its own identifier.
● Key k is stored at its successor node, the first node whose identifiers is equal to or greater than the identifier of key k.
Basic Lookups

- Each node only remembers its successor node in the circle.
- Lookups in clockwise direction.
- Assume N nodes and K keys.
- m is the number of bits in the node/key identifier.

What is the lookup time?

O(N) hops
Finger Table Notations

- Each node maintains additional routing information (e.g., finger tables) to accelerate lookups.
- A finger table contains m entries

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$finger[k].start$</td>
<td>$(n + 2^{k-1}) \mod 2^m$, $1 \leq k \leq m$</td>
</tr>
<tr>
<td>$interval$</td>
<td>$[finger[k].start, finger[k + 1].start)$</td>
</tr>
<tr>
<td>$node$</td>
<td>first node $\geq n.finger[k].start$</td>
</tr>
</tbody>
</table>

$n$ is the identifier of the node

$m = 7$

<table>
<thead>
<tr>
<th>$k$</th>
<th>$start$</th>
<th>interval</th>
<th>node</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>91</td>
<td>[91, 92)</td>
<td>105</td>
</tr>
<tr>
<td>2</td>
<td>92</td>
<td>[92, 94)</td>
<td>105</td>
</tr>
<tr>
<td>3</td>
<td>94</td>
<td>[94, 98)</td>
<td>105</td>
</tr>
<tr>
<td>4</td>
<td>98</td>
<td>[98, 106)</td>
<td>105</td>
</tr>
<tr>
<td>5</td>
<td>106</td>
<td>[106, 122)</td>
<td>120</td>
</tr>
<tr>
<td>6</td>
<td>122</td>
<td>[122, 26)</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>26</td>
<td>[26, 91)</td>
<td>32</td>
</tr>
</tbody>
</table>
Finger Table of Node 10

- A finger table contains \( m \) entries

\[
\begin{array}{|c|c|c|}
\hline
\text{start} & \text{interval} & \text{node} \\
91 & [91, 92) & 105 \\
92 & [92, 94) & 105 \\
94 & [94, 98) & 105 \\
98 & [98, 106) & 105 \\
106 & [106, 122) & 120 \\
122 & [122, 26) & 10 \\
26 & [26, 91) & 32 \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
\text{start} & \text{interval} & \text{node} \\
11 & [11, 12) & 32 \\
12 & [12, 14) & 32 \\
14 & [14, 18) & 32 \\
18 & [18, 26) & 32 \\
26 & [26, 42) & 60 \\
42 & [42, 74) & 60 \\
74 & [74, 11) & 90 \\
\hline
\end{array}
\]

\( n \) is the identifier of the node

\[
\begin{array}{|c|c|}
\hline
\text{Notation} & \text{Definition} \\
\hline
\text{finger}[k].start & (n + 2^{k-1}) \mod 2^m, 1 \leq k \leq m \\
\text{interval} & [\text{finger}[k].start, \text{finger}[k+1].start) \\
\text{node} & \text{first node } \geq n.\text{finger}[k].start \\
\hline
\end{array}
\]
Route $k = 15$

- A finger table contains $m$ entries

<table>
<thead>
<tr>
<th>start</th>
<th>interval</th>
<th>node</th>
</tr>
</thead>
<tbody>
<tr>
<td>91</td>
<td>[91, 92)</td>
<td>105</td>
</tr>
<tr>
<td>92</td>
<td>[92, 94)</td>
<td>105</td>
</tr>
<tr>
<td>94</td>
<td>[94, 98)</td>
<td>105</td>
</tr>
<tr>
<td>98</td>
<td>[98, 106)</td>
<td>105</td>
</tr>
<tr>
<td>106</td>
<td>[106, 122)</td>
<td>120</td>
</tr>
<tr>
<td>122</td>
<td>[122, 26)</td>
<td>10</td>
</tr>
<tr>
<td>26</td>
<td>[26, 91)</td>
<td>32</td>
</tr>
</tbody>
</table>

- $N = 6$
- $m = 7$

<table>
<thead>
<tr>
<th>start</th>
<th>interval</th>
<th>node</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>[11, 12)</td>
<td>32</td>
</tr>
<tr>
<td>12</td>
<td>[12, 14)</td>
<td>32</td>
</tr>
<tr>
<td>14</td>
<td>[14, 18)</td>
<td>32</td>
</tr>
<tr>
<td>18</td>
<td>[18, 26)</td>
<td>32</td>
</tr>
<tr>
<td>26</td>
<td>[26, 42)</td>
<td>60</td>
</tr>
<tr>
<td>42</td>
<td>[42, 74)</td>
<td>60</td>
</tr>
<tr>
<td>74</td>
<td>[74, 11)</td>
<td>90</td>
</tr>
</tbody>
</table>

$n$ is the identifier of the node

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{finger}[k].\text{start}$</td>
<td>$(n + 2^{k-1}) \mod 2^m, 1 \leq k \leq m$</td>
</tr>
<tr>
<td>$\text{interval}$</td>
<td>$[\text{finger}[k].\text{start}, \text{finger}[k+1].\text{start})$</td>
</tr>
<tr>
<td>$\text{node}$</td>
<td>first node $\geq n.\text{finger}[k].\text{start}$</td>
</tr>
</tbody>
</table>
Route $k = 15$

- A finger table contains $m$ entries

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{finger}[k].\text{start}$</td>
<td>$(n + 2^{k-1}) \mod 2^m$, $1 \leq k \leq m$</td>
</tr>
<tr>
<td>.interval</td>
<td>$[\text{finger}[k].\text{start}, \text{finger}[k+1].\text{start})$</td>
</tr>
<tr>
<td>.node</td>
<td>first node $\geq n$, $\text{finger}[k].\text{start}$</td>
</tr>
</tbody>
</table>

$n$ is the identifier of the node

<table>
<thead>
<tr>
<th>start</th>
<th>interval</th>
<th>node</th>
</tr>
</thead>
<tbody>
<tr>
<td>91</td>
<td>[91, 92)</td>
<td>105</td>
</tr>
<tr>
<td>92</td>
<td>[92, 94)</td>
<td>105</td>
</tr>
<tr>
<td>94</td>
<td>[94, 98)</td>
<td>105</td>
</tr>
<tr>
<td>98</td>
<td>[98, 106)</td>
<td>105</td>
</tr>
<tr>
<td>106</td>
<td>[106, 122)</td>
<td>120</td>
</tr>
<tr>
<td>122</td>
<td>[122, 26)</td>
<td>10</td>
</tr>
<tr>
<td>26</td>
<td>[26, 91)</td>
<td>32</td>
</tr>
</tbody>
</table>
Route $k = 15$

- A finger table contains $m$ entries

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{finger}[k].start$</td>
<td>$(n + 2^{k-1}) \mod 2^m$, $1 \leq k \leq m$</td>
</tr>
<tr>
<td>$\text{interval}$</td>
<td>$[\text{finger}[k].start, \text{finger}[k+1].start)$</td>
</tr>
<tr>
<td>$\text{node}$</td>
<td>first node $\geq n.\text{finger}[k].start$</td>
</tr>
</tbody>
</table>

$n$ is the identifier of the node

<table>
<thead>
<tr>
<th>start</th>
<th>interval</th>
<th>node</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>[11, 12)</td>
<td>32</td>
</tr>
<tr>
<td>12</td>
<td>[12, 14)</td>
<td>32</td>
</tr>
<tr>
<td>14</td>
<td>[14, 18)</td>
<td>32</td>
</tr>
<tr>
<td>18</td>
<td>[18, 26)</td>
<td>32</td>
</tr>
<tr>
<td>26</td>
<td>[26, 42)</td>
<td>60</td>
</tr>
<tr>
<td>42</td>
<td>[42, 74)</td>
<td>60</td>
</tr>
<tr>
<td>74</td>
<td>[74, 11)</td>
<td>90</td>
</tr>
</tbody>
</table>

$N = 6$

$m = 7$

<table>
<thead>
<tr>
<th>start</th>
<th>interval</th>
<th>node</th>
</tr>
</thead>
<tbody>
<tr>
<td>91</td>
<td>[91, 92)</td>
<td>105</td>
</tr>
<tr>
<td>92</td>
<td>[92, 94)</td>
<td>105</td>
</tr>
<tr>
<td>94</td>
<td>[94, 98)</td>
<td>105</td>
</tr>
<tr>
<td>98</td>
<td>[98, 106)</td>
<td>105</td>
</tr>
<tr>
<td>106</td>
<td>[106, 122)</td>
<td>120</td>
</tr>
<tr>
<td>122</td>
<td>[122, 26)</td>
<td>10</td>
</tr>
<tr>
<td>26</td>
<td>[26, 91)</td>
<td>32</td>
</tr>
</tbody>
</table>
Route $k = 15$

- A finger table contains $m$ entries

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{finger}[k].$start</td>
<td>$(n + 2^{k-1}) \mod 2^m, 1 \leq k \leq m$</td>
</tr>
<tr>
<td>.interval</td>
<td>$[\text{finger}[k].$start, $\text{finger}[k + 1].$start)</td>
</tr>
<tr>
<td>.node</td>
<td>first node $\geq n.\text{finger}[k].$start</td>
</tr>
</tbody>
</table>

$n$ is the identifier of the node

<table>
<thead>
<tr>
<th>start</th>
<th>interval</th>
<th>node</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>[11, 12)</td>
<td>32</td>
</tr>
<tr>
<td>12</td>
<td>[12, 14)</td>
<td>32</td>
</tr>
<tr>
<td>14</td>
<td>[14, 18)</td>
<td>32</td>
</tr>
<tr>
<td>18</td>
<td>[18, 26)</td>
<td>32</td>
</tr>
<tr>
<td>26</td>
<td>[26, 42)</td>
<td>60</td>
</tr>
<tr>
<td>42</td>
<td>[42, 74)</td>
<td>60</td>
</tr>
<tr>
<td>74</td>
<td>[74, 11)</td>
<td>90</td>
</tr>
</tbody>
</table>

$N = 6$
$m = 7$

Node 32 is your successor
Summary: Finger Table

- Each node maintains additional routing information (e.g., finger tables) to accelerate lookups. This information is not essential for correctness, as long as the successor information is correct.
- A finger table contains $m$ entries
- We tradeoff space for better lookup performance

What is the lookup time?

$O(\log N)$ hops

Credit: UC Berkeley CS 262a Fall 2023 slide deck.