1. Why record time series
2. Time series database is everywhere
3. Time series database requirements
4. Time series database architecture
5. Time series compression
6. Time Series Similarity Retrieval
Why record Time Series?
<table>
<thead>
<tr>
<th>Hour</th>
<th>Knots</th>
<th>Fathoms</th>
<th>Courses steered</th>
<th>Winds</th>
<th>Direction</th>
<th>Force</th>
<th>Pressure (inches)</th>
<th>Temperature (°F)</th>
<th>State of the Weather</th>
<th>Forms of Clouds</th>
<th>State of the Sea</th>
<th>Record of the sail the vessel is under at end of watch</th>
</tr>
</thead>
<tbody>
<tr>
<td>A M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>29.66</td>
<td>48</td>
<td>48</td>
<td></td>
<td></td>
<td>Steam Alone</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>29.60</td>
<td>48</td>
<td>48</td>
<td></td>
<td></td>
<td>0 S</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>29.56</td>
<td>45</td>
<td>45</td>
<td></td>
<td></td>
<td>0 S</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>5</td>
<td></td>
<td></td>
<td>West</td>
<td>1</td>
<td>29.52</td>
<td>45</td>
<td>45</td>
<td></td>
<td></td>
<td>0 S</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
<td>S. W.</td>
<td>1</td>
<td>29.61</td>
<td>46</td>
<td>46</td>
<td></td>
<td></td>
<td>0 S</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
<td>S. W.</td>
<td>1</td>
<td>29.62</td>
<td>46</td>
<td>46</td>
<td></td>
<td></td>
<td>0 S</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
<td>S. W.</td>
<td>1</td>
<td>29.61</td>
<td>45</td>
<td>45</td>
<td></td>
<td></td>
<td>0 S</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
<td>S. W.</td>
<td>1</td>
<td>29.62</td>
<td>46</td>
<td>46</td>
<td></td>
<td></td>
<td>0 S</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>4</td>
<td>E. S. E.</td>
<td></td>
<td>E.</td>
<td>2</td>
<td>29.62</td>
<td>46</td>
<td>46</td>
<td></td>
<td></td>
<td>0 S</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>29.63</td>
<td>47</td>
<td>47</td>
<td></td>
<td></td>
<td>0 S</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>29.65</td>
<td>47</td>
<td>48</td>
<td></td>
<td></td>
<td>0 S</td>
</tr>
<tr>
<td>Noon</td>
<td></td>
<td>25</td>
<td>E. S. E.</td>
<td></td>
<td>Calm</td>
<td>6</td>
<td>29.68</td>
<td>48</td>
<td>48</td>
<td></td>
<td></td>
<td>0 S</td>
</tr>
</tbody>
</table>

**Making passage from New York to St. Johns A. F.**
Fun fact -- Logbook (ship's logs)

The term originally referred to a book for recording readings from the chip log.
High frequency GPS
Time Series Database is Everywhere
Tick Database

Capturing, managing, and processing market data
Operational historian

Record trends and historical information about industrial processes for future reference
Time-series Database

Data Center

KairosDB, graphite, OPENTSDDB, influxdb, riakTS, Facebook

Record CPU, memory, network utilization etc. Data Center Monitoring.
Time Series Database Requirements

Time series is a sequence of timestamp plus value.

Time series data entries are rarely amended, no updates, no delete.

Time series data is often retrieved by reading a contiguous sequence of samples, range query.
System Assumption

1. Read is sequential and ordered
2. Concurrent read workload
3. Data are immutable
4. Writes dominated workload
5. Writes are usually sequential appends
6. Highly compressible data
7. Deleting usually across large time period
As of Spring 2015, Facebook’s monitoring systems generate more than 2 billion unique time series of counters, with about 12 million data points added per second. This represents over 1 trillion points per day.

How to store?
RDBMS Schema

- Time series ID
- Time
- Value

<table>
<thead>
<tr>
<th>Time</th>
<th>Time series ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>15:51:00</td>
<td>101</td>
<td>0.01</td>
</tr>
<tr>
<td>15:51:03</td>
<td>102</td>
<td>1.16</td>
</tr>
<tr>
<td>15:52:07</td>
<td>101</td>
<td>0.04</td>
</tr>
<tr>
<td>15:52:11</td>
<td>101</td>
<td>0.08</td>
</tr>
<tr>
<td>15:53:17</td>
<td>103</td>
<td>4.18</td>
</tr>
</tbody>
</table>
LSM Tree -- Write
LSM Tree -- Compaction

Start compaction

Merge data

Evict tombstones
Remove deletions

End compaction

Consolidate

Available disk space
LSM Tree -- Read
Row Oriented Database
Columnar Database
Columnar Database -- Why is it fast?

- **Reduces the amount of data** you need to load from disk, only the column you need.
- **Minimize seek time**, only require large sequential read.
- **Compression**, column data is of uniform type, significantly reduce information entropy
BTrDB
Berkeley Tree Database

Berkeley Tree Database (BTrDB) K-ary Tree
Berkeley Tree Database (BTrDB) K-ary Tree
Time Series Compression
Frame of Reference Compression (FOR)


Subtract 107

0, 1, 3, 8, 13, 18, 25, 25, 24, 28.
Delta Encoding

We can sometimes get better compression if the difference between the values is small.


1, 2, 5, 5, 5, 7, 0, -1, 4.
Bitshuffle Encoding
Variable-length encodings

- **Varint encoding:**
  - 7 bits per byte with continuation bit
  - **Con:** Decoding requires lots of branches/shifts/masks

- **Idea:** Encode byte length using 2 bits
  - **Better:** fewer branches, shifts, and masks
  - **Con:** Limited to 30-bit values, still some shifting to decode
Time Series Similarity Retrieval
Time Series Similarity Measures

![Graph showing DTW and Euclidean similarity measures]

- DTW
- Euclidean
Time Series Similarity Retrieval

Range Query

Nearest Neighbor query

Discrete Fourier transform (DFT) + R-Tree = (F-Index)

Euclidean distance in time domain preserved in frequency domain.

Thank You!