What is Accumulo?

How can I use Accumulo?

Who is involved in the Accumulo community?

Where is Accumulo going?
Design Drivers

Analysis of big data is central to our customers’ requirements, in which the strongest drivers are:

- **Scalability**: The ability to do twice the work at only (about) twice the cost.
- **Adaptability**: The ability to rapidly evolve the analytical tools available in an operational environment, building upon and enhancing existing capabilities.

From these directives we can derive the following requirements:

- Simplicity in the overall architecture to encourage collaboration and ameliorate learning curve.
- Generic design patterns to store and organize data whose format we don’t control.
- Generic discovery analytics to retrieve and visualize generic data.
- Solutions for common sub-problems, such as multi-level security and enforcement of legal restrictions, built into the infrastructure.
Optimization

... is a secondary concern, given:

- hundreds of evolving applications,
- hundreds of changing data sources,
- non-trivial data volumes,
- many complicated interactions.

Instead, we need a generic platform that is cheap, simple, scalable, secure, and adaptable, with pretty good performance.
Growth of Accumulo
Key/Value Structure

An Accumulo Key is a 5-tuple, including:

- **Row**: controls *Atomicity*
- **Column Family**: controls *Locality*
- **Column Qualifier**: controls *Uniqueness*
- **Visibility**: controls *Access* (unique to Accumulo)
- **Timestamp**: controls *Versioning*

### Sample Entries

<table>
<thead>
<tr>
<th>Row</th>
<th>Col. Fam.</th>
<th>Col. Qual.</th>
<th>Visibility</th>
<th>Timestamp</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam</td>
<td>Favorites</td>
<td>Food</td>
<td>(Public)</td>
<td>20090801</td>
<td>Sushi</td>
</tr>
<tr>
<td>Adam</td>
<td>Favorites</td>
<td>Programming Language</td>
<td>(Private)</td>
<td>20090830</td>
<td>Java</td>
</tr>
<tr>
<td>Adam</td>
<td>Favorites</td>
<td>Programming Language</td>
<td>(Private)</td>
<td>20070725</td>
<td>C++</td>
</tr>
<tr>
<td>Adam</td>
<td>Friends</td>
<td>Bob</td>
<td>(Public)</td>
<td>20110601</td>
<td></td>
</tr>
<tr>
<td>Adam</td>
<td>Friends</td>
<td>Joe</td>
<td>(Private)</td>
<td>20110601</td>
<td></td>
</tr>
</tbody>
</table>
Visibility Label Syntax and Semantics

**Document Labels**

- $Doc_1$: (Federation)
- $Doc_2$: (Klingon|Vulcan)
- $Doc_3$: (Federation&Human&Vulcan)
- $Doc_4$: (Federation&(Human|Vulcan))

**User Authorization Sets**

- $CptKirk$: \{Federation, Human\}
- $MrSpock$: \{Federation, Human, Vulcan\}

**Syntax**

- \( \text{WORD} \Rightarrow [a-zA-Z0-9-]+ \)
- \( \text{CLAUSE} \Rightarrow \text{AND} \)
- \( \Rightarrow \text{OR} \)
- \( \text{AND} \Rightarrow \text{AND} \& \text{AND} \)
- \( \Rightarrow (\text{CLAUSE}) \)
- \( \Rightarrow \text{WORD} \)
- \( \text{OR} \Rightarrow \text{OR} \mid \text{OR} \)
- \( \Rightarrow (\text{CLAUSE}) \)
- \( \Rightarrow \text{WORD} \)

**Semantics**

\[
\frac{(T \Rightarrow \tau) \land (\tau \in A)}{(T, A) \models \text{true}} \quad \text{term}
\]
\[
\frac{(T \Rightarrow T_1 \& T_2) \land ((T_1, A) \models \text{true}) \land ((T_2, A) \models \text{true})}{(T, A) \models \text{true}} \quad \text{and}
\]
\[
\frac{(T \Rightarrow T_1 \mid T_2) \land (((T_1, A) \models \text{true}) \lor ((T_2, A) \models \text{true}))}{(T, A) \models \text{true}} \quad \text{or}
\]
\[
\frac{(T \Rightarrow (T1)) \land (T1 \models \text{true})}{(T, A) \models \text{true}} \quad \text{paren}
\]
Tables

- Collections of key/value pairs form Tables
- Tables are partitioned into Tablets
- Metadata tablets hold info about other tablets, forming a three-level hierarchy
- A Tablet is a unit of work for a Tablet Server
Quick and loose definitions:

Table: A map of keys to values with one global sort order among keys.
Tablet: A row range within a Table.
Tablet Server: The mechanism that hosts Tablets, providing the primary functionality of Bigtable or Accumulo.

Tablet servers have several primary functions:

1. Hosting RPCs (read, write, etc.)
2. Managing resources (RAM, CPU, File I/O, etc.)
3. Scheduling background tasks (compactions, caching, etc.)
4. Handling key/value pairs

Category 4 is almost entirely accomplished through the *Iterator framework*.
Accumulo

What is Accumulo?

How can I use Accumulo?

Who is involved in the Accumulo community?

Where is Accumulo going?

Tablet Server Data Flow

Iterator Uses

- File Reads
- Block Caching
- Merging
- Deletion
- Isolation
- Locality Groups
- Range Selection
- Column Selection
- Cell-level Security
- Versioning
- Filtering
- Aggregation
- Partitioned Joins
The Perils of Distributed Computing

Dealing with failures is hard!

- Operations like table creation are logically atomic, but consist of multiple operations on distributed systems.
- Resource locking (via mutex, semaphores, etc.) provides some sanity.
- Distributed systems have many complicated failure modes: clients, master, tablet servers, and dependent systems can all go offline periodically.
- *Who is responsible for unlocking locks when any component can fail?*
- *How do we know it’s safe to unlock a lock?*
Accumulo Testing Procedures

Testing Frameworks

- **Unit**: Verify correct functioning of each module separately
- **System**: Perform correctness and performance tests on a small running instance
- **Load/Scale**: Generate high loads at scale and measure performance and correctness
- **Random Walk**: Randomly, repeatedly, and concurrently execute a variety of test modules representative of user activity on an instance at scale
- **Simulation**: Evaluate the model to gauge expected performance

Other Considerations

- Scoping tests to include server-side code, client-side code, dependent processes, etc.
- Code coverage vs. path coverage
- Static vs. dynamic analysis
- Simulating failures of distributed components
- Strange failure modes (often hardware/physics-related)
Fault-Tolerant Executor

- If a process dies, previously submitted operations continue to execute on restart.
- FATE serializes every task in Zookeeper before execution.
- The Master process uses FATE to execute table operations and administrative actions.
- FATE eliminates the single point of failure.
Verified State Models

- State models used for many internal functions
- Explicit-state model checking proves correctness
Accumulo
Adam Fuchs

What is Accumulo?

How can I use Accumulo?

Who is involved in the Accumulo community?

Where is Accumulo going?

Hierarchical Decomposition

<table>
<thead>
<tr>
<th>Row ID</th>
<th>Col. Family</th>
<th>Col. Qual.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>attribute</td>
<td>age</td>
<td>&lt;age&gt;</td>
</tr>
<tr>
<td></td>
<td>purchases</td>
<td>discount</td>
<td>&lt;40%&gt;</td>
</tr>
<tr>
<td></td>
<td>returns</td>
<td>sneakers</td>
<td>&lt;cost&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hat</td>
<td>&lt;cost&gt;</td>
</tr>
</tbody>
</table>
Hierarchical Representation

<table>
<thead>
<tr>
<th>Row ID</th>
<th>bill</th>
<th>george</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. Family</td>
<td>attribute</td>
<td>purchases</td>
</tr>
<tr>
<td>Col. Qual.</td>
<td>age</td>
<td>discount</td>
</tr>
<tr>
<td>Value</td>
<td>49</td>
<td>40%</td>
</tr>
</tbody>
</table>
Event Table with Inverted Index

Table:  |  Event Table:  |  Inverted Index:
---|---|---
Row:  |  <UUID>  |  <Term>  
Column Family:  |  <Type>  |  <Type> + <Field>  
Column Qualifier:  |  <Field>  |  <UUID>  
Value:  |  <Term>  |  <Digest of Event>
Multidimensional Index

Table: Geo Index

Row: <Morton Ordered Coordinate>

Column Family: <Event Type>

Column Qualifier: <UUID>

Value: <Digest of Event>

See also: http://en.wikipedia.org/wiki/Geohash
The “shard” Table

Table:

Row:

Column Family:

Column Qualifier (2-Tuples):

Value:

Indexed Event Table

<Partition ID>

“Event”

“Index”

“Geo”

<Event>

<Term>

<UUID>

<Field>

<UUID>

<Morton Coordinate>

<Event and Index Records Co-Partitioned!>
Committers, Contributors, and Community

<table>
<thead>
<tr>
<th>username</th>
<th>name</th>
<th>role</th>
</tr>
</thead>
<tbody>
<tr>
<td>acordova</td>
<td>Aaron Cordova</td>
<td>pmc, committer</td>
</tr>
<tr>
<td>afuchs</td>
<td>Adam Fuchs</td>
<td>pmc, committer</td>
</tr>
<tr>
<td>bimargulies</td>
<td>Benson Margulies</td>
<td>pmc, committer</td>
</tr>
<tr>
<td>billie</td>
<td>Billie Rinaldi</td>
<td>pmc, committer</td>
</tr>
<tr>
<td>cawaring</td>
<td>Chris Waring</td>
<td>pmc, committer</td>
</tr>
<tr>
<td>medined</td>
<td>David Medinets</td>
<td>pmc, committer</td>
</tr>
<tr>
<td>Dennis Patrone</td>
<td>contributor</td>
<td></td>
</tr>
<tr>
<td>drew</td>
<td>Drew Farris</td>
<td>contributor</td>
</tr>
<tr>
<td>edwardyoon</td>
<td>Edward Yoon</td>
<td>contributor</td>
</tr>
<tr>
<td>ecn</td>
<td>Eric Newton</td>
<td>pmc, committer</td>
</tr>
<tr>
<td>jtrost</td>
<td>Jason Trost</td>
<td>committer</td>
</tr>
<tr>
<td>Jesse Yates</td>
<td>contributor</td>
<td></td>
</tr>
<tr>
<td>Jim Klucar</td>
<td>contributor</td>
<td></td>
</tr>
<tr>
<td>vines</td>
<td>John Vines</td>
<td>pmc, committer</td>
</tr>
<tr>
<td>Josh Elser</td>
<td>contributor</td>
<td></td>
</tr>
<tr>
<td>kturner</td>
<td>Keith Turner</td>
<td>pmc, committer</td>
</tr>
<tr>
<td>Matthew Kirkley</td>
<td>contributor</td>
<td></td>
</tr>
<tr>
<td>Michael Wall</td>
<td>contributor</td>
<td></td>
</tr>
<tr>
<td>Sapah Shah</td>
<td>contributor</td>
<td></td>
</tr>
<tr>
<td>Scott Kuehn</td>
<td>contributor</td>
<td></td>
</tr>
<tr>
<td>Travis Pinney</td>
<td>contributor</td>
<td></td>
</tr>
<tr>
<td>William Slacum</td>
<td>contributor</td>
<td></td>
</tr>
</tbody>
</table>

Accumulo-Related Companies
- 42six
- Accumulo Data
- Berico
- Booz Allen Hamilton
- CyberPoint
- Data Tactics
- Eclectic Consulting
- Invertix
- KEYW
- PDI
- Peterson Technologies
- Potomac Fusion
- Praxis
- SAIC
- sqrrl
- SRA
- SW Complete
- Tetra Concepts
- TexelTek
- Your name here!
User Base

Accumulo
Adam Fuchs

What is Accumulo?
How can I use Accumulo?
Who is involved in the Accumulo community?
Where is Accumulo going?
Features in the Pipeline

- Block stats indexing
- Transient block indexing
- Pluggable Authentication and Authorization
- HDFS-based write-ahead log
- Multiple namenode/volume support
- Integration with cluster management systems
- Web-integrated shell
Theoretical Projects and Challenges

- Coprocessors
- Multi-row Transactions
- Improved Iterator Framework
- Statistics API
- Custom Sort Order
- Multi-Data Center Replication
- Other Suggestions?