

Document based storage systems

- PHP Unconference Hamburg
- 27.04.2008

Speaker

- Nils Adermann
- Kore Nordmann

Who uses a RDBMS?

Who uses a document based storage system?

- Amazon S3
- CouchDB
- StrokeDB
 - Ruby
- FeatherDB
 - Java port of CouchDB

Forget RDBMS for now.

What is CouchDB?

- Store anything you want
 - Arbitrary JSON structures
 - Attachments

The project

- Apache project
 - Alpha state
- Main developer: Damian Katz
 - Paid by IBM to develop CouchDB
 - Ex-Lotus-Notes developer

Storage

- ACID compliant
 - Atomicity
 - Consistency
 - Isolation
 - Durability
- MVCC
 - Multiversion Concurrency Control
 - Transactions
- No relational integrity.

Example document.

- {
 _id: "098F6BCD4621D373CADE4E832627B4F6",
 _rev: 4134847,
 "BugId": "098F6BCD4621D373CADE4E832627B4F6",
 "Counter": 1,
 "Name": "test",
 "Type": "Project"
 "tags": ["1.0", "core", "severe"]
}

Example view function

- ```
function (doc) {
 for(var i in doc.tags)
 emit(doc.tags[i], 1);
}
```

# Example view

- ```
{
  _id: "_design/tags",
  _rev: 12345,
  "language": "text/javascript",
  "views":
  {
    "all": "<view function>",
    ...
  }
}
```

Example view result

- GET /db/_view/tags/all
- {"total_rows":42,
 "rows": [
 { "key": "1.0", "value":1},
 { "key": "core", "value":1},
 { "key": "severe", "value":1},
 { "key": "1.0", "value":1},
 ...
]
}

Example reduce

- ```
function (tag, counts) {
 int sum = 0;
 for (var i=0; i<counts.length; i++)
 sum += counts[tag];
 return sum;
}
```

# Result

- GET /db/\_view/tags/count

```
{ "total_rows": 3,
 "rows": [
 { "key": "1.0", "value": 2 },
 { "key": "core", "value": 5 },
 { "key": "severe", "value": 2 }
]
}
```

# Details on views:

- Calculated / updated on request
- Ad-hoc views

# Accessing CouchDB

- Simple HTTP (REST) API.
- URLs:
  - /database/document\_id
  - /database/\_view/name/function
  - /database/\_temp\_view
  - /database/\_design/name
- Methods:
  - POST / PUT / GET / HEAD



# Features

- Robust Data Storage
- REST API
- Replication
- User Authentication
- JavaScript Views
- Built on Erlang/OTP
- Fulltext Search

# Replication

- Will be cool! ;)
- Currently in discussion:
  - Will happen magically

# Revisions

- All document revisions are stored
  - Compaction in latest SVN

# Fulltext search

- Not entirely implemented yet
- Indexing: CouchDB
  - You get: Database and maybe updated documents/  
view results
  - Index with any backend
- Querying: REST-API
  - `/db/?query=unconf`
  - `/db/_view/name/?query=unconf`

# Speed

- Two seeks for each write
- High IO-Load for reads
  - No optimizations
- No caching
  - It's just HTTP – use varnish / squid
    - <http://varnish.projects.linpro.no/>

# Example: User permission management

- Browser...

# Questions

- Please ask!

# The end

- Thank you for listening
  - <http://couchdb.org>
  - <http://damienkatz.net>
  - <http://jan.prima.de>