CouchDB, PHPillow & PHP

Kore Nordmann <kore@php.net>

November 18, 2009

About me

- Kore Nordmann, <kore@php.net>, <kn@ez.no>
- Long time PHP developer
- Regular speaker, author, etc.
- Studies computer science in Dortmund
- Active open source developer:
  - eZ Components (Graph, WebDav, Document), Arbit, PHPUnit, Torii, *PHPillow*, KaForkL, Image 3D, WCV, ...

Outline

Introduction

CouchDB

PHPillow

Views

QA

CouchDB is paradigmn shift

- No tables, document based database

CouchDB is paradigmn shift

- No tables, document based database
- No relational consistency

CouchDB is paradigmn shift

- No tables, document based database
- No relational consistency
  - Who uses foreign keys anyways?

CouchDB is paradigmn shift

- No tables, document based database
- No relational consistency
    - Who uses foreign keys anyways?
- “No” clustered consistency, *eventual* consistency

CouchDB is paradigmn shift

- No tables, document based database
- No relational consistency
  - Who uses foreign keys anyways?
- "No" clustered consistency, *eventual* consistency
- This talk tells you, why this is perfect for web applications

Wiki as an example application

- See what the paradigms mean for $application

Wiki as an example application

- See what the paradigms mean for $application
    - Wiki as an example

Wiki as an example application

- See what the paradigms mean for $application
  - Wiki as an example
  - Refactoring (table-less)

Wiki as an example application

- See what the paradigms mean for $application
  - Wiki as an example
  - Refactoring (table-less)
  - Handle concurrent edits (MVCC)

Wiki as an example application

- See what the paradigms mean for $application
  - Wiki as an example
  - Refactoring (table-less)
  - Handle concurrent edits (MVCC)
  - Clustered (eventual consistency)

Wiki as an example application

- See what the paradigms mean for $application
  - Wiki as an example
  - Refactoring (table-less)
  - Handle concurrent edits (MVCC)
  - Clustered (eventual consistency)
- Other examples
  - Classic ACL handling: `http://kore-nordmann.de/blog/couchdb_a_use_case.html`
  - Applications using CouchDB: `http://wiki.apache.org/couchdb/CouchDB_in_the_wild` (CRM, POI database, issue tracker, social networks, blogs, ...)

Outline

Introduction

CouchDB

PHPillow

Views

QA

CouchDB

- Accessed using R

CouchDB

- Apache top-level project
- Accessed using R

CouchDB

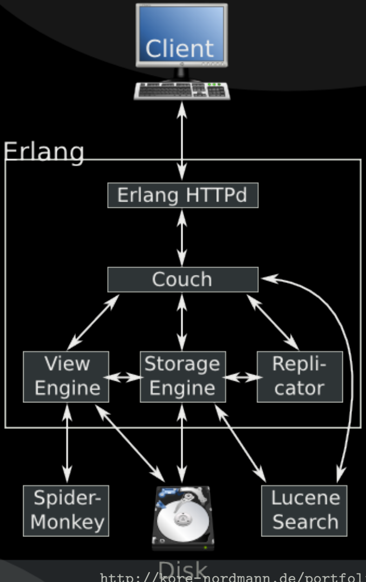
- Apache top-level project
- Queried and indexed using map-reduce
- Accessed using R

CouchDB

- Apache top-level project
- Queried and indexed using map-reduce
- Accessed using REST-ful HTTP

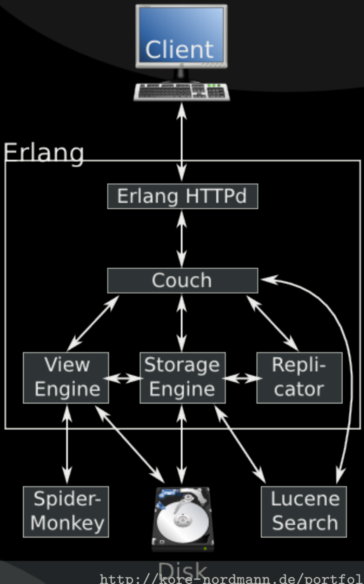
CouchDB

- Erlang/OTP virtual machine, developed by Ericsson



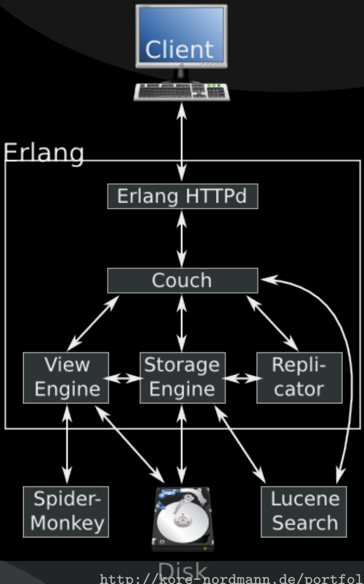
CouchDB

- Erlang/OTP virtual machine, developed by Ericsson
- Highly concurrent



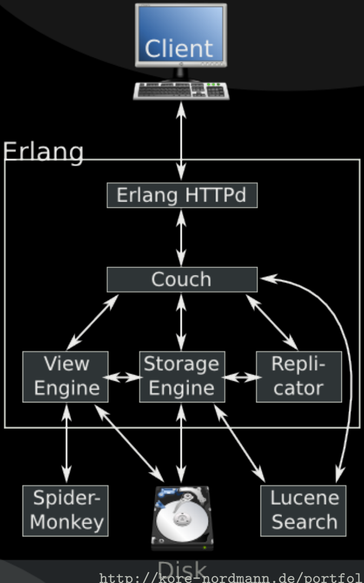
CouchDB

- Erlang/OTP virtual machine, developed by Ericsson
- Highly concurrent
- Scales nearly linearly with the amount of CPUs



CouchDB

- Erlang/OTP virtual machine, developed by Ericsson
- Highly concurrent
- Scales nearly linearly with the amount of CPUs
- High reliability (nine nines)



Create a database

```
$ curl -i -X PUT http://localhost:5984/ipc_wiki

HTTP/1.1 201 Created
 Server: CouchDB/0.10.0 (Erlang OTP/R13B)
 Location: http://localhost:5984/ipc_wiki
 Date: Fri, 13 Nov 2009 14:07:57 GMT
 Content-Type: text/plain;charset=utf-8
 Content-Length: 12
 Cache-Control: must-revalidate

{"ok":true}
```

Create a wiki document

```
$ curl -i -X PUT http://localhost:5984/ipc_wiki/Start
    --data '{"name": "Start", "text": "Hello World!"}'

HTTP/1.1 201 Created
 Server: CouchDB/0.10.0 (Erlang OTP/R13B)
 Location: http://localhost:5984/ipc_wiki/Start
 Etag: "1-6bfd4885b6c62bb5169a19d5a81927e3"
 Date: Fri, 13 Nov 2009 14:14:55 GMT
 Content-Type: text/plain;charset=utf-8
 Content-Length: 68
 Cache-Control: must-revalidate

 {"ok":true,"id":"Start","rev":"1-6
    bfd4885b6c62bb5169a19d5a81927e3"}
```

Get a wiki document back

```
$ curl -i -X GET http://localhost:5984/ipc_wiki/Start

HTTP/1.1 200 OK
 Server: CouchDB/0.10.0 (Erlang OTP/R13B)
 Etag: "1-6bfd4885b6c62bb5169a19d5a81927e3"
 Date: Fri, 13 Nov 2009 14:15:48 GMT
 Content-Type: text/plain;charset=utf-8
 Content-Length: 97
 Cache-Control: must-revalidate

{"_id":"Start","_rev":"1-6
    bfd4885b6c62bb5169a19d5a81927e3","name":"Start","
    text":"Hello World!"}
```

Inter document links

- There is no ensured inter document consistency in CouchDB

documents:

ocument (n:m)

```
}
```

Inter document links

- There is no ensured inter document consistency in CouchDB
- Different possibilities of relating documents:
  - document (n:m)

```
}
```

Inter document links

- There is no ensured inter document consistency in CouchDB
- Different possibilities of relating documents:
  - List IDs of related documents in document (n:m)

```
}
```

Inter document links

- There is no ensured inter document consistency in CouchDB
- Different possibilities of relating documents:
    - List IDs of related documents in document (n:m)
    - … both directions are feasible

```
  }
```

Inter document links

- There is no ensured inter document consistency in CouchDB
- Different possibilities of relating documents:
  - List IDs of related documents in document (n:m)
  - ... both directions are feasible
  - Embed the whole related document (1:n)

```
}
```

Inter document links

- There is no ensured inter document consistency in CouchDB
- Different possibilities of relating documents:
  - List IDs of related documents in document (n:m)
  - ... both directions are feasible
  - Embed the whole related document (1:n)
- Solution depends on update-ratio

```
{ "type": "wiki",
  "name": "Start",
  "text": "...",
}

{ "type": "discussion",
  "wiki": "wiki-Start",
  "text": "...",
  "comments": [
    { "comment": "..." },
  ],
}
```

Migration / Refactoring

- Change document structure at any time

Migration / Refactoring

- Change document structure at any time
- No need for non-transaction-safe DDL

Migration / Refactoring

- Change document structure at any time
- No need for non-transaction-safe DDL
- Fits rapid development approaches with common customer requested changes to the data structure

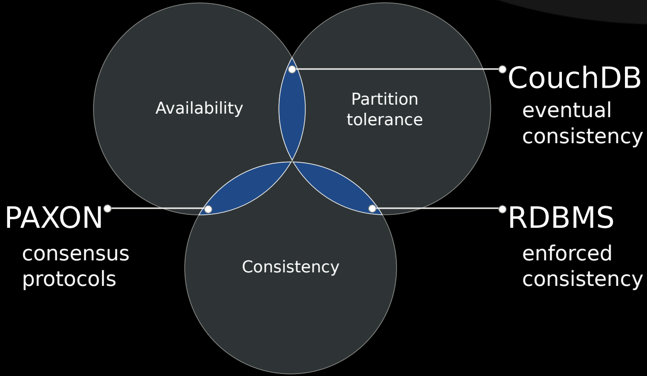
Local conflicts

- Updates and deletes need to use the revision ID

```
$ curl -X GET http://localhost:5984/ipc_wiki/Start

{"_id":"Start","_rev":"1-6
    bfd4885b6c62bb5169a19d5a81927e3","name":"Start
    ","text":"Hello World!"}

$ curl -i -X DELETE http://localhost:5984/ipc_wiki/
    Start?rev=3-2357834573

HTTP/1.1 409 Conflict
{"error":"conflict","reason":"Document update
    conflict."}

kore@kore-hp2140 couchdb $ curl -i -X DELETE

{"ok":true,"id":"Start","rev":"2-9423
    c28e3d23bfa03a99994dff367c98"}
```

The CAP theorem

- The CAP theorem, read more in “CouchDB: The Definitive Guide” [JCA09]



Availability

Partition tolerance

Consistency

CouchDB
eventual consistency

PAXON
consensus protocols

RDBMS
enforced consistency

- CouchDB employs “Eventual Consistency”

Eventual consistency

- Offline replication, like Lotus Notes
- Delayed synchronization (push, pull)
  - Deterministic conflict resolution on replication on all nodes

Eventual consistency

- Offline replication, like Lotus Notes
- Delayed synchronization (push, pull)
    - Deterministic conflict resolution on replication on all nodes
- Scales well for seldom concurrent writes

Eventual consistency

- Offline replication, like Lotus Notes
- Delayed synchronization (push, pull)
  - Deterministic conflict resolution on replication on all nodes
- Scales well for seldom concurrent writes
  - Structure your documents accordingly

Eventual consistency

- Offline replication, like Lotus Notes
- Delayed synchronization (push, pull)
  - Deterministic conflict resolution on replication on all nodes
- Scales well for seldom concurrent writes
  - Structure your documents accordingly
- Ubuntu One uses this to synchronize files, contact data, etc.

Eventual consistency

- Offline replication, like Lotus Notes
- Delayed synchronization (push, pull)
  - Deterministic conflict resolution on replication on all nodes
- Scales well for seldom concurrent writes
  - Structure your documents accordingly
- Ubuntu One uses this to synchronize files, contact data, etc.
- Mozilla develops a JavaScript implementation of the CouchDB API [Moz09]

Attachments

- CouchDB allows you to attach files to documents

Attachments

- CouchDB allows you to attach files to documents
- Files are replicated, even incrementally since the next version

Attachments

- CouchDB allows you to attach files to documents
- Files are replicated, even incrementally since the next version
- You can server full Web-Applications from a CouchDB

Security

- Simple database based access restrictions

Security

- Simple database based access restrictions
- Using HTTP plain auth

Security

- Simple database based access restrictions
- Using HTTP plain auth
- More fine grained access control is under discussion

Outline

Introduction

CouchDB

PHPillow

Views

QA

PHPillow

- Object-oriented client for CouchDB
- PHP >= 5.2 since last release (5.3 only before)
- >96% test coverage

PHPillow

- Object-oriented client for CouchDB
- PHP >= 5.2 since last release (5.3 only before)
- >96% test coverage
- Still in alpha state

PHPillow

- Object-oriented client for CouchDB
- PHP >= 5.2 since last release (5.3 only before)
- >96% test coverage
- Still in alpha state
  - Since CouchDB was still “alpha” up to last month

Document examples

- Document creation example

```php
// Create a document
$doc = new phpillowUserDocument();
$doc->login = 'kore';
$doc->name  = 'Kore␣Nordmann';
$doc->data  = array(
    'mail' => "kore@php.net",
    // ...
);
$id = $doc->save();

// Fetch a document by ID
$doc = new phpillowUserDocument( $id );
```

Outline

Introduction

CouchDB

PHPillow

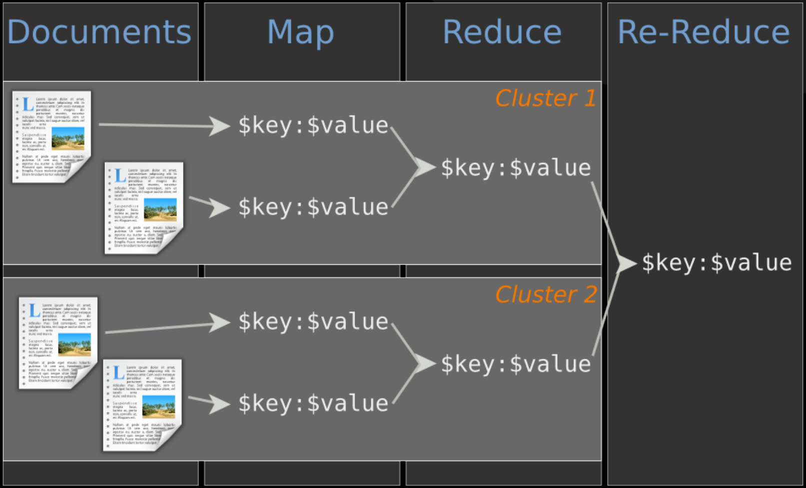
Views

QA

Map-reduce-views

- "MapReduce is a software framework introduced by Google to support distributed computing on large data sets on clusters of computers." [Wik09]
- Used by CouchDB to implement views

Map-Reduce


Documents
Map
Reduce
Re-Reduce
Cluster 1
$key:$value
$key:$value
$key:$value
$key:$value
Cluster 2
$key:$value
$key:$value
$key:$value

Map-Reduce

- Map and reduce functions are custom

Map-Reduce

- Map and reduce functions are custom
- Defined in any language
    - ECMAScript (Spidermonkey), PHP, Ruby, Python, Erlang, . . .

Map-Reduce

- Map and reduce functions are custom
- Defined in any language
  - ECMAScript (Spidermonkey), PHP, Ruby, Python, Erlang, . . .
- Multiple map-reduce-functions (per document)

Map-Reduce

- Map and reduce functions are custom
- Defined in any language
  - ECMAScript (Spidermonkey), PHP, Ruby, Python, Erlang, . . .
- Multiple map-reduce-functions (per document)
- Keys and values may be any JSON data structure

Map-Reduce

- Map and reduce functions are custom
- Defined in any language
  - ECMAScript (Spidermonkey), PHP, Ruby, Python, Erlang, . . .
- Multiple map-reduce-functions (per document)
- Keys and values may be any JSON data structure
- Reduce is optional, mapping serves as a document index

Map-Reduce

- Map and reduce functions are custom
- Defined in any language
  - ECMAScript (Spidermonkey), PHP, Ruby, Python, Erlang, . . .
- Multiple map-reduce-functions (per document)
- Keys and values may be any JSON data structure
- Reduce is optional, mapping serves as a document index
- Reduce may be applied to subsets of the documents

Map-Reduce

- Map and reduce functions are custom
- Defined in any language
  - ECMAScript (Spidermonkey), PHP, Ruby, Python, Erlang, . . .
- Multiple map-reduce-functions (per document)
- Keys and values may be any JSON data structure
- Reduce is optional, mapping serves as a document index
- Reduce may be applied to subsets of the documents
- Reduce may be grouped

Wiki

- Index all documents by their title

```
function( doc )
{
    if ( doc.type == "wiki_page" )
    {
        emit( [doc.namespace, doc.title], doc._id );
    }
}
```

- No reduce function

Wiki

- Index all documents by their title

```
["development_wiki", "BuildModuleDesign"]     => "wiki_page-development_wiki_buildmoduledesign"
["development_wiki", "CodingGuidelines"]      => "wiki_page-development_wiki_codingguidelines"
["development_wiki", "DiscussionProtocols"]   => "wiki_page-development_wiki_discussionprotocols"
["development_wiki", "ModuleDesign"]          => "wiki_page-development_wiki_moduledesign"
["development_wiki", "Protocol_08_02_07"]     => "wiki_page-development_wiki_protocol_08_02_07"
["development_wiki", "VCSModuleDesign"]       => "wiki_page-development_wiki_vcsmoduledesign"
...
```

Outline

Introduction

CouchDB

PHPillow

Views

QA

URLs

- Apache CouchDB: `http://couchdb.org/`
- Free CouchDB book: `http://books.couchdb.org/relax/`
- PHPillow: `http://arbitracker.org/phpillow.html`

The end

- Open questions?
- Further remarks?
- Contact
  - Mail: <kore@php.net>
  - Web: `http://kore-nordmann.de/` (Slides will be available here soonish)
  - Twitter: `http://twitter.com/koredn`

Bibliography I

[JCA09] Noah Slater J. Chris Anderson, Jan Lehnardt, *Couchdb: The definitive guide*, O'Reilly Media, Inc., 2009.

[Moz09] Mozilla, *Browsercouch documentation*, November 2009.

[Wik09] Wikipedia, *Mapreduce — wikipedia, the free encyclopedia*, 2009, [Online; accessed 27-August-2009].

Full-Text-Search

- Index all documents by all their words

```
function( doc ) {
  if ( doc.type == "tracker_issue" ) {
    // Simple word indexing, does not respect overall occurences of words,
    // stopwords, different word seperation characters, or word variations.
    var text  = doc.title.replace( /[\s:.,!?-]+/g, "␣" ) +
        doc.text.replace( /[\s:.,!?-]+/g, "␣" );
    var words = text.split( "␣" );
    for ( var i = 0; i < words.length; ++i ) {
      value = {};
      value[doc._id] = 1;
      emit( words[i].toLowerCase(), value );
    }
  }
}
```

Wiki

- Index all documents by all their words

```
...
"a"         => {tracker_issue-8: 1}
"a"         => {tracker_issue-8: 1}
"a"         => {tracker_issue-8: 1}
"a"         => {tracker_issue-8: 1}
"a"         => {tracker_issue-81: 1}
"a"         => {tracker_issue-83: 1}
"a"         => {tracker_issue-83: 1}
"able"      => {tracker_issue-39: 1}
"able"      => {tracker_issue-56: 1}
"able"      => {tracker_issue-73: 1}
"able"      => {tracker_issue-80: 1}
"about"     => {tracker_issue-24: 1}
"about"     => {tracker_issue-43: 1}
"about"     => {tracker_issue-85: 1}
...
```

Full-Text-Search

- Reduce by word count

```
function( keys, values ) {
  var count = {};
  for ( var i in values ) {
    for ( var id in values[i] ) {
      if ( count[id] ) {
        count[id] = values[i][id] + count[id];
      } else {
        count[id] = values[i][id];
      }
    }
  }
  return count;
}
```

Wiki

- Index all documents by all their words

```
...
"a"             => {
                     tracker_issue-68: 6,
                     tracker_issue-66: 6,
                     tracker_issue-22: 4,
                     tracker_issue-63: 3,
                     tracker_issue-60: 2,
                     tracker_issue-35: 2,
                     tracker_issue-34: 1,
                     tracker_issue-31: 1,
                     ...
                   }
"able"          => {tracker_issue-86: 1, tracker_issue-80:
      1, tracker_issue-73: 1, tracker_issue-56: 1,
      tracker_issue-39: 1}
"about"         => {tracker_issue-85: 1, tracker_issue-43:
      1, tracker_issue-24: 1}
...
```