PHPillow & CouchDB

Kore Nordmann <kore@php.net>

September 11, 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

Real world examples

QA

Databases

- Who uses an RDBMS (relational database management system)?

Databases

- Who uses an RDBMS (relational database management system)?
- Who uses a hash based “database”? (MemcacheDB, ...)

Databases

- Who uses an RDBMS (relational database management system)?
- Who uses a hash based "database"? (MemcacheDB, ...)
- Who uses a "document" based database?
  - Amazon SimpleDB
  - StrokeDB (Ruby)
  - FeatherDB (Java port of CouchDB)
  - CouchDB

Outline

Introduction

CouchDB

PHPillow

Views

Real world examples

QA

PHPillow

- A PHP based object oriented client for CouchDB

more comfortable

PHPillow

- A PHP based object oriented client for CouchDB
- The Pillow to make using the couch more comfortable

CouchDB vs. RDBMS

- RDBMS: Static data with dynamic views
- CouchDB: Dynamic data with “static” views

CouchDB

- Document based database

CouchDB

- Document based database
  - Arbitrary JSON objects, attachments

CouchDB

- Document based database
  - Arbitrary JSON objects, attachments
  - Store anything you want, even deep structures

CouchDB

- Document based database
  - Arbitrary JSON objects, attachments
  - Store anything you want, even deep structures
  - Schema-less

CouchDB

- Document based database
  - Arbitrary JSON objects, attachments
  - Store anything you want, even deep structures
  - Schema-less
  - No inter-document consistency

CouchDB

- Document based database
  - Arbitrary JSON objects, attachments
  - Store anything you want, even deep structures
  - Schema-less
  - No inter-document consistency
- Apache Project

CouchDB

- Document based database
  - Arbitrary JSON objects, attachments
  - Store anything you want, even deep structures
  - Schema-less
  - No inter-document consistency
- Apache Project
- Communicates with clients using HTTP

CouchDB

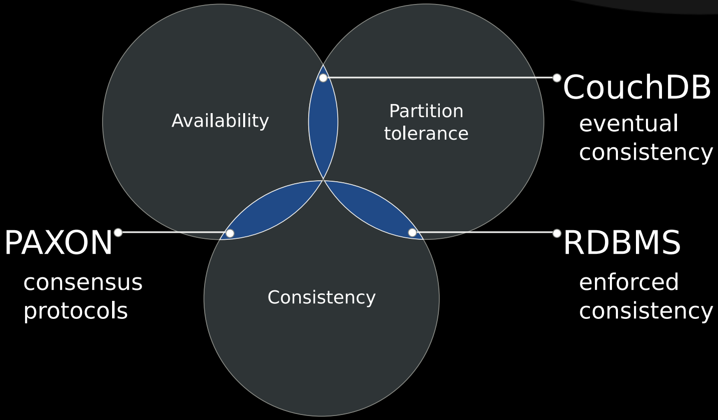
- Document based database
  - Arbitrary JSON objects, attachments
  - Store anything you want, even deep structures
  - Schema-less
  - No inter-document consistency
- Apache Project
- Communicates with clients using HTTP
- Trivial replication [Apa09]

CouchDB

- Document based database
  - Arbitrary JSON objects, attachments
  - Store anything you want, even deep structures
  - Schema-less
  - No inter-document consistency
- Apache Project
- Communicates with clients using HTTP
- Trivial replication [Apa09]
- Local consistency, weak clustered consitency (ACID, MVCC)

The CAP theorem

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



Architecture

- Written in/for Erlang/OTP

Architecture

- Written in/for Erlang/OTP
- Scales nearly linearly with amount of processors

Architecture

- Written in/for Erlang/OTP
- Scales nearly linearly with amount of processors
- Highly fault tolerant (9 nines @Ericsson)
  - “Live” updates of application code

Architecture

- Written in/for Erlang/OTP
- Scales nearly linearly with amount of processors
- Highly fault tolerant (9 nines @Ericsson)
  - “Live” updates of application code
- VM Developed by Ericsson for telephony systems (OTP)
  - Focus on stability and scalability

Communicates with HTTP (1/3)

```
$ curl -I -X PUT http://localhost:5984/unconf
HTTP/1.1 201 Created
 Server: CouchDB/0.10.0a773833 (Erlang OTP/R12B)
 Content-Type: text/plain;charset=utf-8

 {"ok":true}
```

Communicates with HTTP (2/3)

```
$ curl -I -X GET http://localhost:5984/unconf
HTTP/1.1 200 OK
Server: CouchDB/0.10.0a773833 (Erlang OTP/R12B)
Content-Type: text/plain;charset=utf-8

{
  "db_name": "unconf",
  "doc_count": 0,
  "doc_del_count": 0,
  "update_seq": 0,
  "purge_seq": 0,
  "compact_running": false,
  "disk_size": 4096,
  "instance_start_time": "1243673303593969",
  "disk_format_version": 2
}
```

Communicates with HTTP (3/3)

```
$ curl -I -X GET http://localhost:5984/unknown
HTTP/1.1 404 Object Not Found
Server: CouchDB/0.10.0a773833 (Erlang OTP/R12B)
Content-Type: text/plain;charset=utf-8

{"error":"not_found","reason":"Missing"}
```

Outline

Introduction

CouchDB

PHPillow

Views

Real world examples

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 is still “alpha”

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

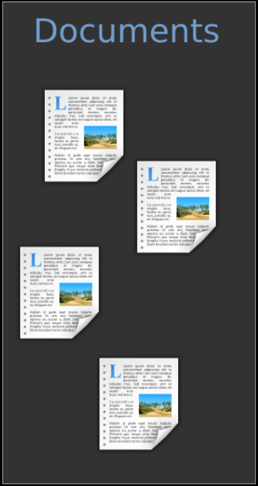
Real world examples

QA

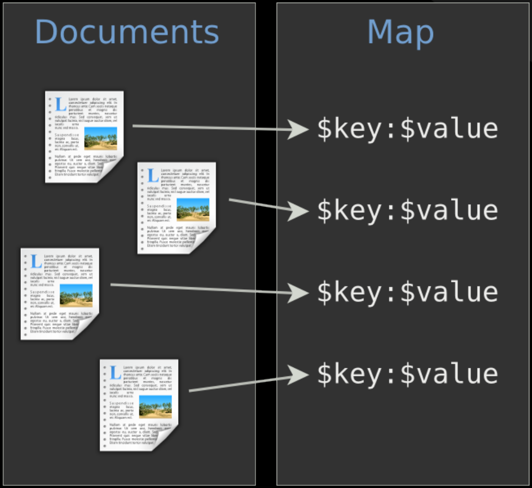
Map-reduce-views

- “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

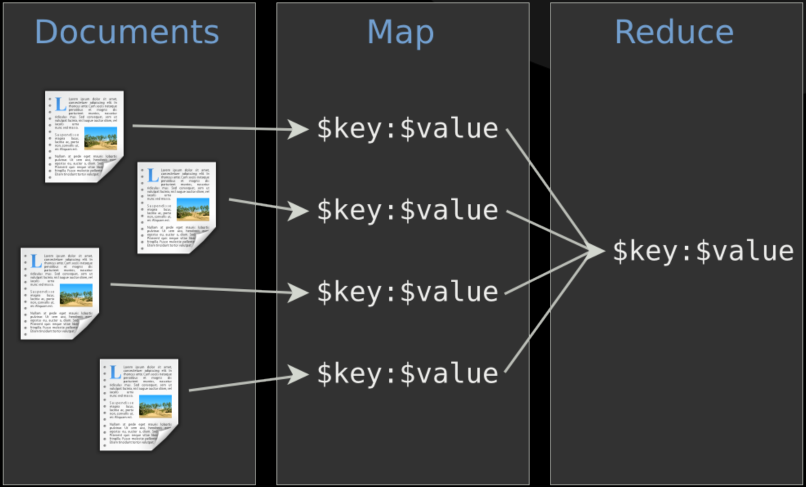
Map-Reduce



Map-Reduce



Map-Reduce



Documents

Map

Reduce

$key:$value

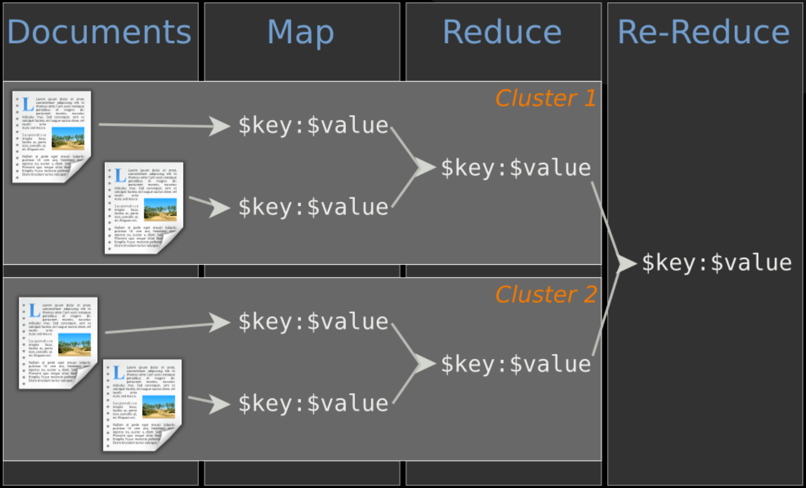
$key:$value

$key:$value

$key:$value

$key:$value

Map-Reduce



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

Map-Reduce example

- A single document

```
{   _id:          "tracker_issue-24",
    _rev:         "1-1015848757",
    revisions:    [ ... ],
    issue:        "24",
    author:       "user-kore",
    edited:       1249305176,                    // !
    title:        "Implement␣VCS␣module",
    text:         "...",
    issueType:    "enhancement",
    state:        "closed",
    priority:     "medium",
    resolution:   "implemented",
    scheduled:    "0.2-alpha",
    assigned:     [ "user-kore" ],
    type:         "tracker_issue"                // !
}
```

Map-Reduce example

- The map function

```
function( doc )
{
    if ( doc.type == "tracker_issue" )
    {
        date = new Date();
        date.setTime( doc.edited * 1000 );
        emit( [
                date.getUTCFullYear(),
                date.getUTCMonth() + 1,
                date.getUTCDate(),
                date.getUTCHours(),
                date.getUTCMinutes(),
                date.getUTCSeconds(),
            ], 1 );
        // You could also emit the whole doc as value
    }
}
```

Map-Reduce example

- The mapping result

```
[2008, 10, 11, 9, 11, 12]   => 1
[2008, 10, 11, 9, 11, 12]   => 1
[2008, 10, 11, 9, 11, 12]   => 1
[2008, 10, 11, 9, 13, 8]    => 1
[2008, 10, 11, 9, 13, 44]   => 1
[2008, 10, 11, 9, 14, 2]    => 1
[2008, 10, 12, 17, 46, 15]  => 1
[2008, 10, 12, 17, 57, 52]  => 1
[2008, 10, 12, 18, 0, 45]   => 1
[2008, 10, 14, 8, 36, 29]   => 1
[2008, 10, 14, 19, 33, 21]  => 1
[2008, 10, 14, 19, 33, 35]  => 1
```

Map-Reduce example

- The reduce function

```
function( keys, values, combine )
{
    if ( combine )
    {
        return sum( values );
    }
    else
    {
        return values.length;
    }
}
```

Map-Reduce example

- The reduce result

```
null                          => 12
```

Map-Reduce example

- The grouped reduce result

```
[2008, 10, 11, 9, 11, 12]   => 3
[2008, 10, 11, 9, 13, 8]    => 1
[2008, 10, 11, 9, 13, 44]   => 1
[2008, 10, 11, 9, 14, 2]    => 1
[2008, 10, 12, 17, 46, 15]  => 1
[2008, 10, 12, 17, 57, 52]  => 1
[2008, 10, 12, 18, 0, 45]   => 1
[2008, 10, 14, 8, 36, 29]   => 1
[2008, 10, 14, 19, 33, 21]  => 1
[2008, 10, 14, 19, 33, 35]  => 1
```

Map-Reduce example

- The filtered grouped reduce result
- startkey=[2008,10,11] endkey=[2008,10,12]

```
[2008, 10, 11, 9, 11, 12]   => 3
[2008, 10, 11, 9, 13, 8]    => 1
[2008, 10, 11, 9, 13, 44]   => 1
[2008, 10, 11, 9, 14, 2]    => 1
```

Map-Reduce example

- The grouped reduce result, with group level
- group-level=3

```
[2008, 10, 11]              => 6
[2008, 10, 12]              => 3
[2008, 10, 14]              => 3
```

Outline

Introduction

CouchDB

PHPillow

Views

Real world examples

QA

Applications

- Seems useful for everything, which follows the ”document” paradigm.
  - Wikis
  - Issue tracker
  - Content management

Applications

- Seems useful for everything, which follows the "document" paradigm.
  - Wikis
  - Issue tracker
  - Content management
- Not useful for data with hard consistency constraints

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"
...
```

Wiki

- Find last edits in one namespace

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

- No reduce function

Wiki

- Find last edits in one namespace

```
["development_wiki", 1223731672]     => "wiki_page-
    development_wiki_codingguidelines"
["development_wiki", 1223731854]     => "wiki_page-
    development_wiki_protocol_08_02_14"
["development_wiki", 1223731918]     => "wiki_page-
    development_wiki_protocol_08_02_07"
["development_wiki", 1223817097]     => "wiki_page-
    development_wiki_roadmap"
["development_wiki", 1223832559]     => "wiki_page-
    development_wiki_discussionprotocols"
["development_wiki", 1223833275]     => "wiki_page-
    development_wiki_moduledesign"
...
```

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}
...
```

Summary

- Document based database
- Views using any language, following the map-reduce-pattern.
- Fast, reliable, scalable

Outline

Introduction

CouchDB

PHPillow

Views

Real world examples

QA

URLs

- Apache CouchDB: `http://couchdb.org/`
- Free CouchDB book: `http://books.couchdb.org/relax/`
- PHPillow: `http://kore-nordmann.de/projects/phpillow/`
- CouchDB use case: `http://kore-nordmann.de/blog/phpillow_php_couchdb_wrapper.html`
  - Implementing a user permission system using CouchDB

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`
  - Thank me: `http://wishlist.kore-nordmann.de/`

Bibliography I

[Apa09] Apache, *Replication*, `http://wiki.apache.org/couchdb/Replication`, July 2009.

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

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