

CouchDB - a document based database

Kore Nordmann
<kore@php.net>
@koredn

March 20, 2010

- ▶ Kore Nordmann, <kore@php.net>
- ▶ Long time PHP developer
- ▶ Studies computer science in Dortmund, currently writing thesis
- ▶ Active open source developer:
 - ▶ eZ Components (Graph, WebDav, Document), *Arbit*, PHPUnit, Torii, *PHPillow*, KaForkL, Image 3D, WCV, ...

Introduction

General

Structure

API

Views

Consistency

Applications

QA

- ▶ Structure

- ▶ Structure
- ▶ Consistency

- ▶ Structure
- ▶ Consistency
- ▶ API

- ▶ Structure
- ▶ Consistency
- ▶ API
- ▶ Applications

Introduction

General

Structure

API

Views

Consistency

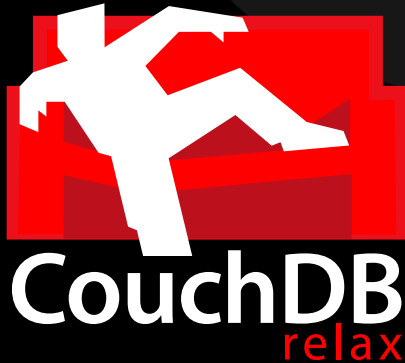
Applications

QA





- ▶ Apache top-level project



- ▶ Apache top-level project
- ▶ Build in Erlang / on Erlang/OTP

Introduction

General

Structure

API

Views

Consistency

Applications

QA

- ▶ Document based database

- ▶ Document based database
- ▶ No pre-defined structure (tables)

- ▶ Document based database
- ▶ No pre-defined structure (tables)
- ▶ Put in any JSON object you want

- ▶ Document based database
- ▶ No pre-defined structure (tables)
- ▶ Put in any JSON object you want
 - ▶ Even deep structures (arrays of objects)

- ▶ Document based database
- ▶ No pre-defined structure (tables)
- ▶ Put in any JSON object you want
 - ▶ Even deep structures (arrays of objects)
 - ▶ You may attach any number of files to documents

► Example wiki document

```
1 { "title": "Barcamp 2010",
2   "text": "Welcome to the Barcamp...",
3   "creator": "user-bar",
4   "edited": 2935678239,
5   "revisions": [
6     { "title": "Barcamp 2009",
7       "text": "Welcome to the Barcamp...",
8       "creator": "user-foo",
9       "edited": 2935678183,
10    }
11  ],
12  ...
13 ]
14 }
```

► Example wiki document

```
1 { "title":      "Barcamp 2010",
2   "creator":   "user-bar",
3   "text":      "<h1>Welcome to the Barcamp</h1>"
4
5               
7               "...",
8   "_attachments": {
9     "logo.png": {
10      "content_type": "image\png",
11      "stub":         true,
12      "length":       42,
13    }
14 }
```

- ▶ Change document structure at any time

- ▶ Change document structure at any time
- ▶ No need for non-transaction-safe Data Definition Language (DDL)

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

- ▶ Change document structure at any time
- ▶ No need for non-transaction-safe Data Definition Language (DDL)
- ▶ Fits rapid development approaches with common customer requested changes to the data structure
 - ▶ You need to handle this in your application properly, of course:
 - ▶ Incrementally update structure on modification
 - ▶ Liberal validation on read

Introduction

General

Structure

API

Views

Consistency

Applications

QA

- ▶ RESTful HTTP access

- ▶ RESTful HTTP access
- ▶ HTTP is available on “all” platforms natively
 - ▶ No PHP extension required
 - ▶ Just use PHPs HTTP stream wrapper, pecl/http or curl

- ▶ RESTful HTTP access
- ▶ HTTP is available on “all” platforms natively
 - ▶ No PHP extension required
 - ▶ Just use PHPs HTTP stream wrapper, pecl/http or curl
- ▶ You can use all your known HTTP middleware
 - ▶ Reverse proxies for scaling reads (Varnish, Squid)
 - ▶ Simple custom proxy configuration for direct “AJAX” access

- ▶ GET / POST / PUT / DELETE
- ▶ <METHOD> <http://<host>/<database>/<document>>

```
1 $ curl -i -X PUT http://localhost:5984/Barcamp_wiki
2
3 HTTP/1.1 201 Created
4 Server: CouchDB/0.10.0 (Erlang OTP/R13B)
5 Location: http://localhost:5984/Barcamp_wiki
6 Date: Fri, 13 Jan 2009 14:07:57 GMT
7 Content-Type: text/plain; charset=utf-8
8 Content-Length: 12
9 Cache-Control: must-revalidate
10
11 {"ok": true}
```

```
1 $ curl -i -X PUT http://localhost:5984/Barcamp_wiki/  
   Start --data '{"name": "Start", "text": "Hello  
   World!"}'  
2  
3 HTTP/1.1 201 Created  
4 Server: CouchDB/0.10.0 (Erlang OTP/R13B)  
5 Location: http://localhost:5984/Barcamp_wiki/Start  
6 Etag: "1-6bfd4885b6c62bb5169a19d5a81927e3"  
7 Date: Fri, 13 Jan 2009 14:14:55 GMT  
8 Content-Type: text/plain; charset=utf-8  
9 Content-Length: 68  
10 Cache-Control: must-revalidate  
11  
12 {"ok":true,"id":"Start","rev":"1-6  
   bfd4885b6c62bb5169a19d5a81927e3"}
```

```
1 $ curl -i -X GET http://localhost:5984/Barcamp_wiki/  
   Start  
2  
3 HTTP/1.1 200 OK  
4 Server: CouchDB/0.10.0 (Erlang OTP/R13B)  
5 Etag: "1-6bfd4885b6c62bb5169a19d5a81927e3"  
6 Date: Fri, 13 Jan 2009 14:15:48 GMT  
7 Content-Type: text/plain; charset=utf-8  
8 Content-Length: 97  
9 Cache-Control: must-revalidate  
10  
11 {"_id": "Start", "_rev": "1-6  
    bfd4885b6c62bb5169a19d5a81927e3", "name": "Start",  
    "text": "Hello World!"}
```


- ▶ Simple database based access restrictions

- ▶ Simple database based access restrictions
- ▶ Using HTTP plain auth

- ▶ Simple database based access restrictions
- ▶ Using HTTP plain auth
- ▶ More fine grained access control will be in next release
 - ▶ Define functions which decide if a request from a user will be accepted.

Introduction

General

Structure

API

Views

Consistency

Applications

QA

- ▶ How to query such a mess?

- ▶ How to query such a mess?
 - ▶ Views are small scripts, run for all documents in a database

- ▶ How to query such a mess?
 - ▶ Views are small scripts, run for all documents in a database
 - ▶ Views are built iteratively, results stored in BTrees
 - ▶ Once built, they are *fast*

- ▶ How to query such a mess?
 - ▶ Views are small scripts, run for all documents in a database
 - ▶ Views are built iteratively, results stored in BTrees
 - ▶ Once built, they are *fast*
 - ▶ Mostly JavaScript, but also PHP, Ruby, Perl, Erlang, ...

- ▶ How to query such a mess?
 - ▶ Views are small scripts, run for all documents in a database
 - ▶ Views are built iteratively, results stored in BTrees
 - ▶ Once built, they are *fast*
 - ▶ Mostly JavaScript, but also PHP, Ruby, Perl, Erlang, ...
 - ▶ A view may emit any number of key-value pairs for each document

- ▶ How to query such a mess?
 - ▶ Views are small scripts, run for all documents in a database
 - ▶ Views are built iteratively, results stored in BTrees
 - ▶ Once built, they are *fast*
 - ▶ Mostly JavaScript, but also PHP, Ruby, Perl, Erlang, ...
 - ▶ A view may emit any number of key-value pairs for each document
 - ▶ Key and value may be any JSON structure

► Index all wiki documents by their title

```
1 function( doc )
2 {
3     if ( doc.title && doc.text )
4     {
5         emit( doc.title , doc._id );
6     }
7 }
```

► Index all wiki documents by their title

```
1 function( doc )
2 {
3     if ( doc.type == "wiki" )
4     {
5         emit( doc.title , doc._id );
6     }
7 }
```

► Index all documents by their title

1	" BuildModuleDesign "	⇒	" wiki-buildmoduledesign "
2	" CodingGuidelines "	⇒	" wiki-codingguidelines "
3	" DiscussionProtocols "	⇒	" wiki-discussionprotocols "
4	" ModuleDesign "	⇒	" wiki-moduledesign "
5	" Protocol_08_02_07 "	⇒	" wiki-protocol_08_02_07 "
6	" VCSModuleDesign "	⇒	" wiki-vcsmoduledesign "
7	...		

- ▶ Index all documents by their title

1	" BuildModuleDesign "	⇒	" wiki-buildmoduledesign "
2	" CodingGuidelines "	⇒	" wiki-codingguidelines "
3	" DiscussionProtocols "	⇒	" wiki-discussionprotocols "
4	" ModuleDesign "	⇒	" wiki-moduledesign "
5	" Protocol_08_02_07 "	⇒	" wiki-protocol_08_02_07 "
6	" VCSModuleDesign "	⇒	" wiki-vcsmoduledesign "
7	...		

- ▶ Custom deterministic IDs can ensure uniqueness of documents
 - ▶ Just set the `_id` property on insert.

- ▶ Index all documents by their title

1	" BuildModuleDesign "	⇒	" wiki-buildmoduledesign "
2	" CodingGuidelines "	⇒	" wiki-codingguidelines "
3	" DiscussionProtocols "	⇒	" wiki-discussionprotocols "
4	" ModuleDesign "	⇒	" wiki-moduledesign "
5	" Protocol_08_02_07 "	⇒	" wiki-protocol_08_02_07 "
6	" VCSModuleDesign "	⇒	" wiki-vcsmoduledesign "
7	...		

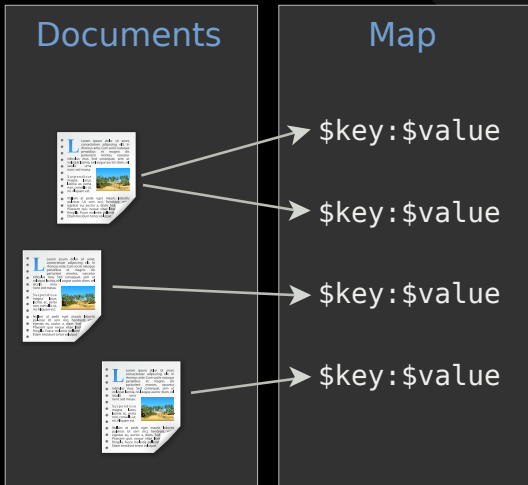
- ▶ Custom deterministic IDs can ensure uniqueness of documents
 - ▶ Just set the `_id` property on insert.
- ▶ CouchDB can also generate IDs for you

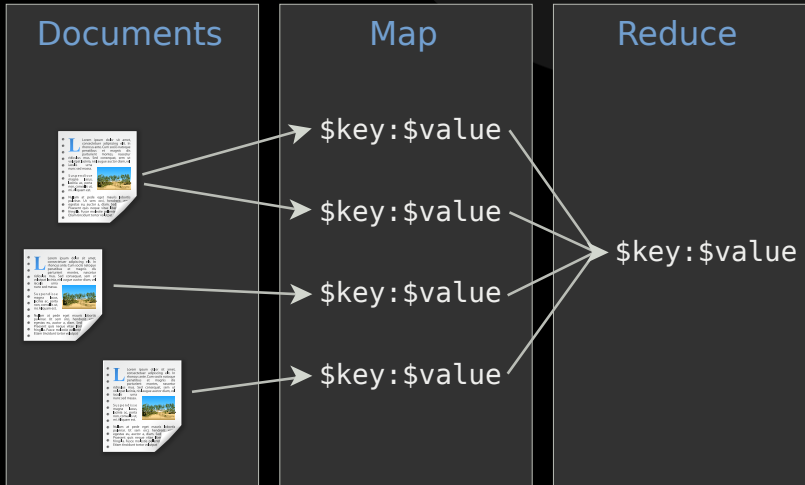
- ▶ “MapReduce is a software framework introduced by Google to support distributed computing on large data sets on clusters of computers.” [Wik09]
- ▶ Used by CouchDB to implement views

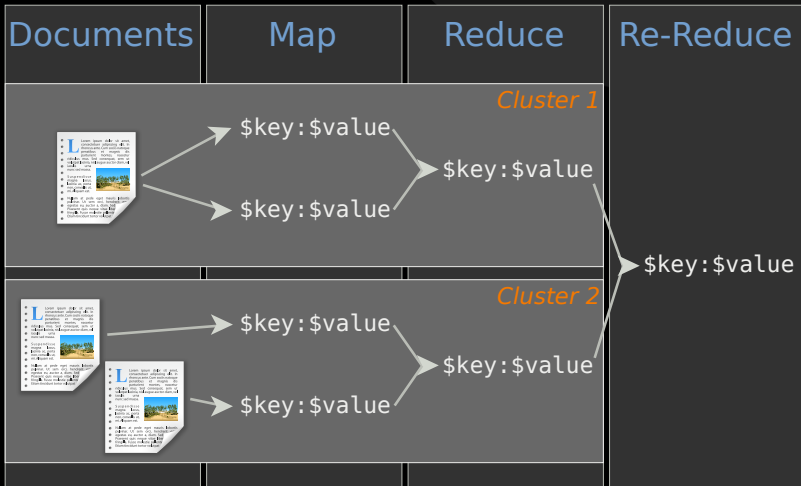
- ▶ “MapReduce is a software framework introduced by Google to support distributed computing on large data sets on clusters of computers.” [Wik09]
- ▶ Used by CouchDB to implement views
- ▶ Just a framework / pattern: You can implement “any” algorithm using map-reduce.

Documents









- ▶ Map and reduce functions are custom

- ▶ Map and reduce functions are custom
- ▶ Reduce is optional, plain view serves as a document index

- ▶ Map and reduce functions are custom
- ▶ Reduce is optional, plain view serves as a document index
- ▶ Reduce may be applied to subsets of the documents

- ▶ Map and reduce functions are custom
- ▶ Reduce is optional, plain view serves as a document index
- ▶ Reduce may be applied to subsets of the documents
- ▶ Reduce may be grouped

- ▶ The simplest reduce function is just `count()`
 - ▶ Often used for statistics

```
1 function( keys , values , combine )
2 {
3     return sum( values );
4 }
```

▶ The reduce result

1 `null` \Rightarrow 42

- ▶ What should be covered in more depth?
 - ▶ Map-reduce views in CouchDB
 - ▶ Scalability / consistency in CouchDB

► The map function

```
1 function( doc )
2 {
3     if ( doc.type == "wiki" )
4     {
5         date = new Date();
6         date.setTime( doc.edited * 1000 );
7         emit( [
8             date.getUTCFullYear(),
9             date.getUTCMonth() + 1,
10            date.getUTCDate(),
11            date.getUTCHours(),
12            date.getUTCMinutes(),
13            date.getUTCSeconds(),
14            ], 1 );
15        // You could also emit the whole doc as value
16    }
17 }
```

► The mapping result

```
1 [2008, 10, 11, 9, 11, 12] => 1
2 [2008, 10, 11, 9, 11, 12] => 1
3 [2008, 10, 11, 9, 11, 12] => 1
4 [2008, 10, 11, 9, 13, 8] => 1
5 [2008, 10, 11, 9, 13, 44] => 1
6 [2008, 10, 11, 9, 14, 2] => 1
7 [2008, 10, 12, 17, 46, 15] => 1
8 [2008, 10, 12, 17, 57, 52] => 1
9 [2008, 10, 12, 18, 0, 45] => 1
10 [2008, 10, 14, 8, 36, 29] => 1
11 [2008, 10, 14, 19, 33, 21] => 1
12 [2008, 10, 14, 19, 33, 35] => 1
```

▶ The reduce function

```
1 function( keys , values , combine )
2 {
3     return sum( values );
4 }
```


▶ The reduce result

1 `null` \Rightarrow 12

► The grouped reduce result

```
1 [2008, 10, 11, 9, 11, 12] => 3
2 [2008, 10, 11, 9, 13, 8] => 1
3 [2008, 10, 11, 9, 13, 44] => 1
4 [2008, 10, 11, 9, 14, 2] => 1
5 [2008, 10, 12, 17, 46, 15] => 1
6 [2008, 10, 12, 17, 57, 52] => 1
7 [2008, 10, 12, 18, 0, 45] => 1
8 [2008, 10, 14, 8, 36, 29] => 1
9 [2008, 10, 14, 19, 33, 21] => 1
10 [2008, 10, 14, 19, 33, 35] => 1
```

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

```
1 [2008, 10, 11, 9, 11, 12] => 3
2 [2008, 10, 11, 9, 13, 8] => 1
3 [2008, 10, 11, 9, 13, 44] => 1
4 [2008, 10, 11, 9, 14, 2] => 1
```

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

1	[2008, 10, 11]	⇒	6
2	[2008, 10, 12]	⇒	3
3	[2008, 10, 14]	⇒	3

- ▶ Index all documents by all their words

```
1 function( doc ) {
2   if ( doc.type == "wiki" ) {
3     // Simple word indexing, does not respect overall
4     // occurrences of words,
5     // stopwords, different word separation characters,
6     // or word variations.
7     var text = doc.title.replace( /\s:.,!?-]+/g, "_" )
8     +
9     doc.text.replace( /\s:.,!?-]+/g, "_" )
10    ;
11    var words = text.split( "_" );
12    for ( var i = 0; i < words.length; ++i ) {
13      value = {};
14      value[doc._id] = 1;
15      emit( words[i].toLowerCase(), value );
16    }
17  }
18 }
```

► Index all documents by all their words

```
1 ...
2 "a"      => {wiki-8: 1}
3 "a"      => {wiki-8: 1}
4 "a"      => {wiki-8: 1}
5 "a"      => {wiki-8: 1}
6 "a"      => {wiki-81: 1}
7 "a"      => {wiki-83: 1}
8 "a"      => {wiki-83: 1}
9 "able"   => {wiki-39: 1}
10 "able"  => {wiki-56: 1}
11 "able"  => {wiki-73: 1}
12 "able"  => {wiki-80: 1}
13 "about" => {wiki-24: 1}
14 "about" => {wiki-43: 1}
15 "about" => {wiki-85: 1}
16 ...
```

► Reduce by word count

```
1 function( keys , values ) {
2     var count = {};
3     for ( var i in values ) {
4         for ( var id in values[i] ) {
5             if ( count[id] ) {
6                 count[id] = values[i][id] + count[id];
7             } else {
8                 count[id] = values[i][id];
9             }
10        }
11    }
12    return count;
13 }
```

► Index all documents by all their words

```
1  ...
2  "a"      => {
3          wiki-68: 6,
4          wiki-66: 6,
5          wiki-22: 4,
6          wiki-63: 3,
7          wiki-60: 2,
8          wiki-35: 2,
9          wiki-34: 1,
10         wiki-31: 1,
11         ...
12         }
13  "able"   => {wiki-86: 1, wiki-80: 1, wiki-73: 1, wiki
14             -56: 1, wiki-39: 1}
15  "about"  => {wiki-85: 1, wiki-43: 1, wiki-24: 1}
```


Introduction

General

Structure

API

Views

Consistency

Applications

QA

- ▶ Multi-Version Concurrency Control

- ▶ Multi-Version Concurrency Control
- ▶ All documents in the database are versioned

- ▶ Multi-Version Concurrency Control
- ▶ All documents in the database are versioned
 - ▶ Don't use it for application level document versioning

- ▶ Multi-Version Concurrency Control
- ▶ All documents in the database are versioned
 - ▶ Don't use it for application level document versioning
- ▶ Updates and deletes need to specify the revision ID

- ▶ Multi-Version Concurrency Control
- ▶ All documents in the database are versioned
 - ▶ Don't use it for application level document versioning
- ▶ Updates and deletes need to specify the revision ID
- ▶ Changing outdated documents result in conflicts

- ▶ There is no ensured inter document consistency in CouchDB

ents:

- ▶ There is no ensured inter document consistency in CouchDB
- ▶ Different possibilities of relating documents:

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

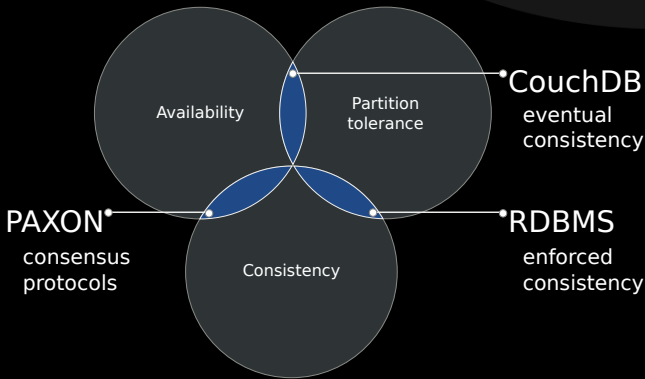
- ▶ There is no ensured inter document consistency in CouchDB
- ▶ Different possibilities of relating documents:
 - ▶ List IDs of related documents in document (n:m)
 - ▶ ... both directions are feasible

- ▶ 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)

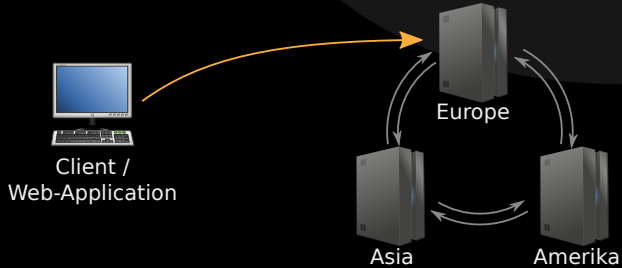
- ▶ There is no ensured inter document consistency in CouchDB
- ▶ Different possibilities of relating documents:
 - ▶ List IDs of related documents in document (n:m)
 - ▶ ... both directions are feasible
 - ▶ Embed the whole related document (1:n)
- ▶ Solution depends on update-ratio

```
1  { "type": "wiki",  
2    "title": "Hello world",  
3    "text": "...",  
4    "comments": [  
5      { "comment": "..."} ],  
6  ],  
7  "creator": "user-foo",  
8  }
```

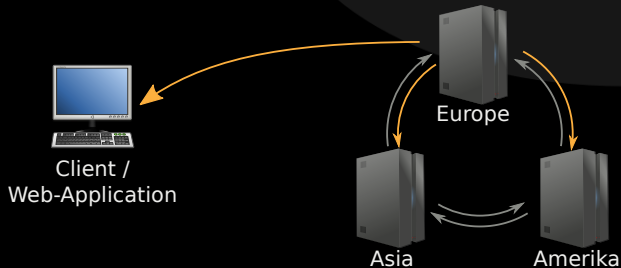
- ▶ The CAP theorem, read more in “CouchDB: The Definitive Guide” [JCA09]



- ▶ CouchDB employs “Eventual Consistency” [Vog09]

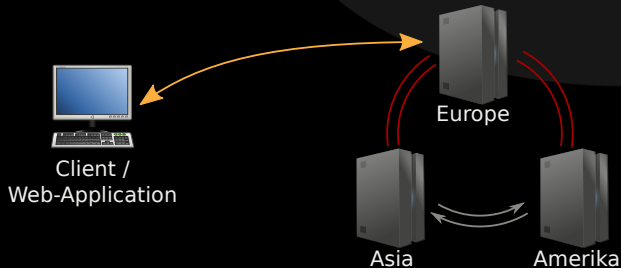


► Structure your



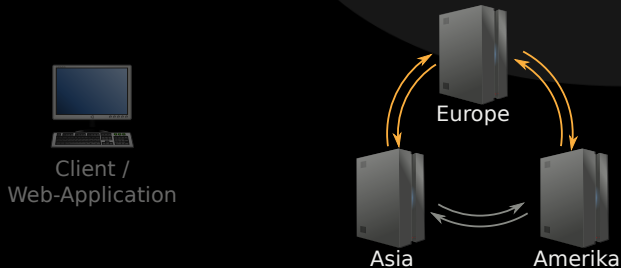
- ▶ Delayed, triggered synchronization (push, pull)
 - ▶ Deterministic (manual) conflict resolution on replication on all nodes

▶ Structure your...



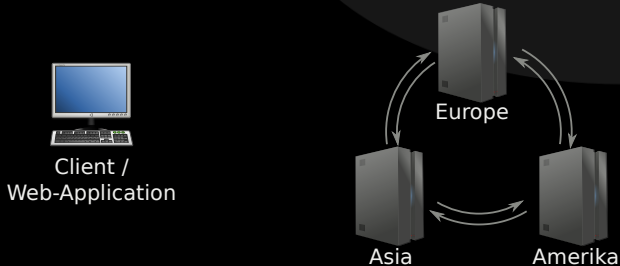
- ▶ Delayed, triggered synchronization (push, pull)
 - ▶ Deterministic (manual) conflict resolution on replication on all nodes

▶ Structure your...



- ▶ Delayed, triggered synchronization (push, pull)
 - ▶ Deterministic (manual) conflict resolution on replication on all nodes
- ▶ Scales well for seldom concurrent writes

▶ Structure your



- ▶ Delayed, triggered synchronization (push, pull)
 - ▶ Deterministic (manual) conflict resolution on replication on all nodes
- ▶ Scales well for seldom concurrent writes
 - ▶ Structure your documents accordingly

Introduction

General

Structure

API

Views

Consistency

Applications

QA

- ▶ CouchDB allows you to attach files to documents

- ▶ CouchDB allows you to attach files to documents
- ▶ Files are replicated

- ▶ CouchDB allows you to attach files to documents
- ▶ Files are replicated
- ▶ You can serve full Web-Applications from a CouchDB
 - ▶ See CouchApp
- ▶ Deploy using PUSH-replication

- ▶ Mirror database into userspace

- ▶ Mirror database into userspace
- ▶ Offline usage and synchronization of Browser applications

- ▶ Mirror database into userspace
- ▶ Offline usage and synchronization of Browser applications
- ▶ Mozilla develops a JavaScript implementation of the CouchDB API [Moz09]

- ▶ Ubuntu One uses CouchDB
- ▶ Synchronize contacts & date between nodes, or to a server

- ▶ Ubuntu One uses CouchDB
- ▶ Synchronize contacts & date between nodes, or to a server
- ▶ Yes, all Ubuntu Karmic users already have a CouchDB running

- ▶ Arbit uses CouchDB for issue tracking, wiki, FAQ and more
- ▶ Other applications: `http://wiki.apache.org/couchdb/CouchDB_in_the_wild`

- ▶ CouchDB is fast (enough)
- ▶ Document oriented approach allows new application development approaches
- ▶ CouchDB scales really well, horizontally and vertically
- ▶ CouchDB fits web applications really well
 - ▶ RDBMS are still better for single-cluster scalable applications with strong integrity requirements.

Introduction

General

Structure

API

Views

Consistency

Applications

QA

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

- ▶ 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>

- [JCA09] Noah Slater J. Chris Anderson, Jan Lehnardt, *Couchdb: The definitive guide*, O'Reilly Media, Inc., 2009.
- [Moz09] Mozilla, *Browsercouch documentation*, November 2009.
- [Vog09] Werner Vogels, *Eventually consistent - revisited*, http://www.allthingsdistributed.com/2008/12/eventually_consistent.html, December 2009.
- [Wik09] Wikipedia, *Mapreduce — wikipedia, the free encyclopedia*, 2009, [Online; accessed 27-August-2009].