

CouchDB

Michał Nowikowski



Agenda

- Wprowadzenie do CouchDB
- Mój przypadek
- Wyniki i wnioski



Dokumenty CouchDB

- Format **JSON**
 - Pary nazwa – wartość
 - Możliwe tablice i struktury
- **Załączniki**
 - Brak limitów na liczbę i rozmiar
 - Mogą zawierać dowolną treść: text, html, wideo, etc.



Przykład dokumentu

```
{  
  _id: "2DA92AAA628CB4156134F36927CF48",  
  _rev: "75AA3DA9",  
  typ: "książka",  
  tytuł: "Podróże Guliwera",  
  autor: {  
    imię: "Jonathan",  
    nazwisko: "Swift"  
  }  
}
```



Funkcjonalność CouchDB

- Baza danych dokumentów
- API HTTP
- Zapytania w języku Javascript
 - Poprzez podejście **map/reduce**
- Replikacja
- Itp



Dlaczego CouchDB

- Brak z góry obranego schematu
- Łatwy interfejs HTTP
 - Istnieje wiele bibliotek w wielu językach
- Przyjazny format dokumentów - JSON



Typy aplikacji

- Takie które są zorientowane na dokumenty
- Email, TODOs, notatki, wiki, forum, CMS, CRM, katalogi, zarządzanie wydatkami, zarządzanie zasobami, zarządzanie projektami, śledzenie błędów
- Czyli spora część aplikacji webowych



Widoki

- Koncepcja podobna do widoków SQL
- Tylko do odczytu, istnieją niezależnie od danych
- Statyczne zapytania dla dynamicznych danych
 - Model map/reduce
 - Index uaktualniany inkrementalnie
- Operacje na widoku:
 - Wybranie po kluczu, wybranie po zakresie, limit



Przykładowy widok

```
function (doc) {  
  if (doc.type == "książka")  
    emit(doc.autor.nazwisko, doc.tytuł);  
}
```

- Kluczem jest nazwisko, wartością tytuł
- Funkcje widoku przechowywane są w dokumentach projektowych ("design document")
- Zawartość widoku pobierana poprzez HTTP GET



Przykładowy dokument projektowy

```
{
  _id: "_design/książki",
  _rev: "4-94e22927f021d464a4f2573ae",
  views: {
    po_autorze: {
      map: "function (doc) {
if (doc.type == 'książka')
  emit(doc.autor.nazwisko, doc.tytuł);}"
    }
  }
}
```



Wyciąganie danych z widoku

```
GET /db/_design/ksiazki/_view/po_autorze?key='Swift'
```

- Wynik

```
{
  total_rows: 1,
  rows: [{
    id: "2DA92AAA628CB4156134F36927",
    key: "Swift",
    value: "Przygody Guliwera"}]
}
```



Porównanie z SQL

SQL

Predefiniowany schemat

Normalizacja: dane rozproszone po tabelach

Schemat musi być znany aby wydać zapytanie

Dynamiczne zapytania na statycznym schemacie

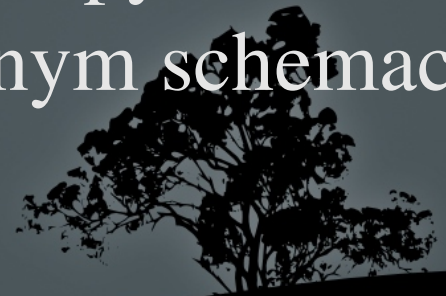
CouchDB

Dynamiczny schemat

Denormalizacja: dane zagnieżdżone, duplikacje

Musi być znana tylko nazwa dokumentu

Statyczne zapytania na dynamicznym schemacie



Dynamiczny SQL

- Tablice klucz – wartość
 - Traktowane jako anty-wzorzec
 - Ciężko budowania zapytania; nie skaluje się
- Przechowywanie struktury jako blob
 - Zserializowany obiekt Javy albo Pythona
 - Nie można wykonać na takiej kolumnie zapytań SQL



CouchDB w produkcji

- DesktopCouch / Ubuntu One

<https://launchpad.net/desktopcouch>

<http://www.freedesktop.org/wiki/Specifications/desktopcouch>

- CouchDB in the Wild

http://wiki.apache.org/couchdb/CouchDB_in_the_wild

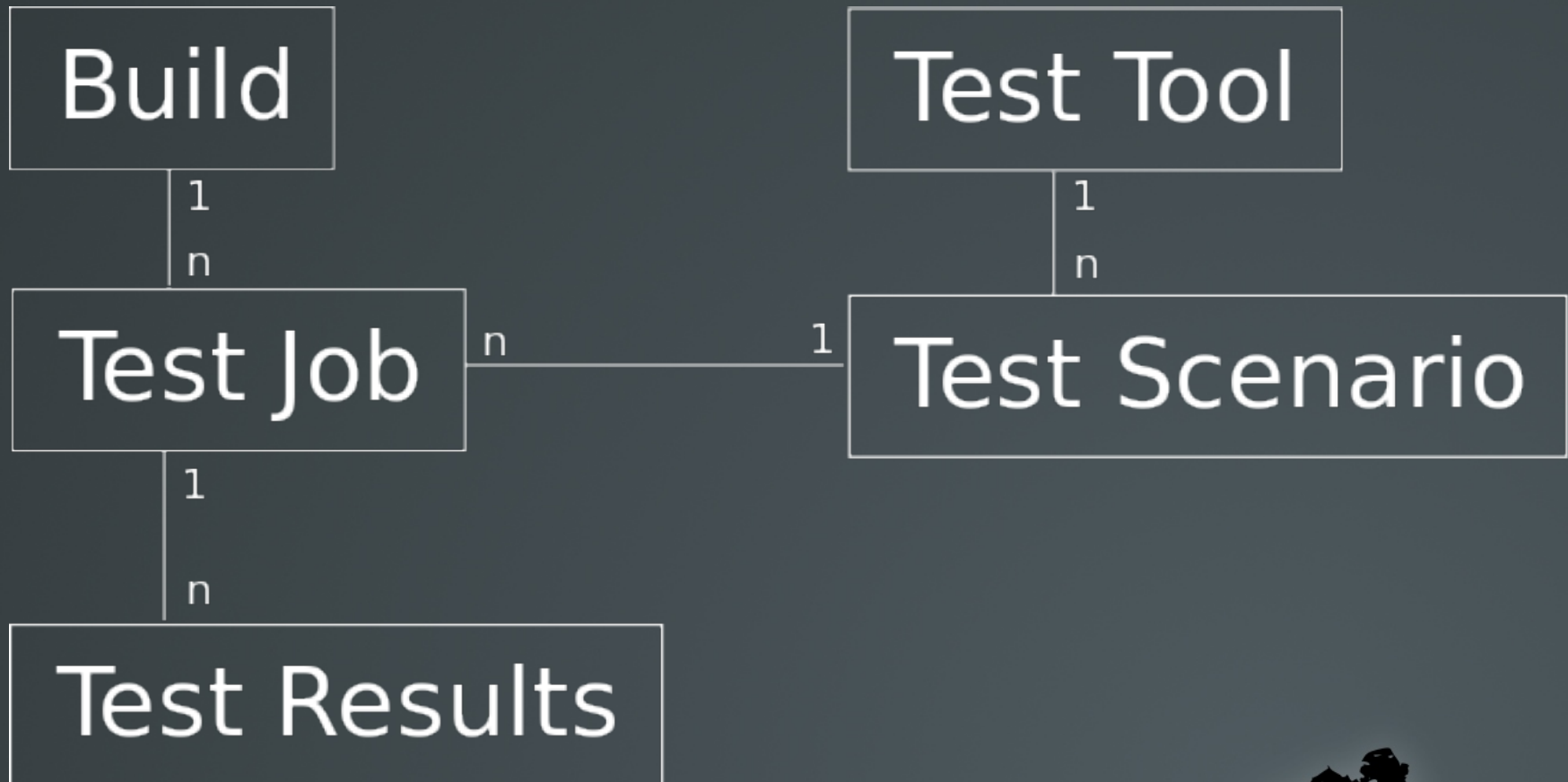


Mój przypadek

- Raportowanie wyników testów
- Mnóstwo testów (ok 10.000)
- Jeszcze więcej wyników per dzień (2-5x)



Schemat rekordów



Przykładowe dane (1)

```
var test_tool = {  
  type: 'tool',  
  name: 'tool1'  
};
```

```
var test_scenario = {  
  type: 'scenario',  
  tool: <tool_id>,  
  name: 'scen 1'  
};
```



Przykładowe dane (2)

```
var build = {  
  type: 'build',  
  name: 'build_001'  
};  
var test_job = {  
  type: 'job',  
  build: <build_id>,  
  scenario: <scenario_id>,  
  results: {  
    tc_1: 1,  
    tc_2: 0  
  }  
};
```



Moje widoki

- Lista narzędzi i scenariuszów testowych
- Lista buildów
- Detaliczne rezultaty testów buildu
- Statystyka rezultatów testów dla scenariusza: liczba passed i total



Strony WWW

- Takie jak widoki, plus:
- Porównanie rezultatów dwóch wskazanych buildów
- Analiza statystyczna rezultatów scenariuszów dla zakresu buildów (średni wynik)



Przykład strony

1. Tools and Scenarios 2. Builds 3. Test results of build 4. Pass rate **5. Two builds comparison** 6. Results trends

Build 1: ▼

Build 2: ▼

Reload

Tool	Scenario	Test case	Build 1	Build 2
test tool 3	scenario 3-2	testcase_719	0	0
test tool 3	scenario 3-2	testcase_718	1	1
test tool 3	scenario 3-2	testcase_1308	0	0
test tool 3	scenario 3-2	testcase_1309	0	1
test tool 3	scenario 3-2	testcase_711	0	1
test tool 3	scenario 3-2	testcase_710	1	0
test tool 3	scenario 3-2	testcase_713	1	1
test tool 3	scenario 3-2	testcase_712	0	0
test tool 3	scenario 3-2	testcase_715	0	0
test tool 3	scenario 3-2	testcase_714	1	0
test tool 3	scenario 3-2	testcase_717	0	1

Page 1 of 900 Displaying results 1 - 20 of 18000

Implementacja

- 3 wieczory
- Inicjacja bazy, dokumenty projektowe, zapytania: **python-couchdb**
- Widoki i strona WWW: **CouchApp**
 - <http://github.com/couchapp/couchapp>
- Widżety na stronie WWW: **ExtJs**
 - <http://extjs.com>



Źródła i uruchomienie

- Requirements:
 - CouchDB (on Ubuntu 9.10 already present)
 - python-virtualenv
- `git clone http://github.com/godfryd/checking-couchdb`
- `make`
- visit `http://localhost:5984/tests-reporting/_design/tr/index.html`



Pomiary

- **9** scenariuszy, **10** buildów
- **18 000** przypadków testowych (dla 9 scenariuszy razem)
- **500kB** logów per każdy scenariusz
- ok. **50MB** zajętości bazy
- Czas załadowania widoków grubo poniżej **100ms**
- Czas przetwarzania po stronie przeglądarki zależny od rozmiaru danych: **do kilkudziesięciu sekund**



Wnioski

- Fajna baza jako back-end
- Wydajna
- Fajnie, że nie narzuca schematu
- Trochę pracy wymagają dokumenty projektowe
- Nie praktyczne jest tworzenie dynamicznych zapytania: bardzo wolno działa



Links

- Ta prezentacja
<http://godfryd.wordpress.com/2010/01/06/checking-couchdb/>
- Repo na GitHub
<http://github.com/godfryd/checking-couchdb>
- CouchDB wiki
<http://wiki.apache.org/couchdb>
- CouchDB book
<http://books.couchdb.org/relax/>
- CouchDB internals
<http://horicky.blogspot.com/2008/10/couchdb-implementation.html>



Links c.d.

- CouchApp

<http://github.com/couchapp/couchapp>

- Python-CouchDB

<http://code.google.com/p/couchdb-python/>

