

from **MySQL**

to



mongoDB

潘凡(*nightsailer*)

March 3, 2011 / Mongo Beijing

About.me

潘 凡

@nightsailer

nightsailer # gmail.com

<http://nightsailer.com/>

<http://github.com/nightsailer>

Mysql data type

char / varchar

Int/BigInt

Float/Double/Decimal

Blob/Text

....

存储和传输

BSON [bee · sahn]

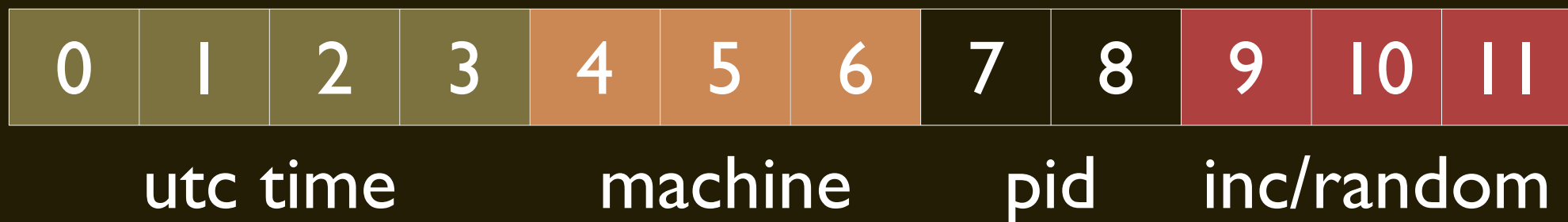
Byte
Int32/Int64
Double

Object Id
Array
Boolean
Javascript code
Embed document
.....

<http://bsonspec.org/>

BSON

Object Id



Enforce collection unique: `_id`
Should global/cluster unique

BSON

Date / Timestamp

Perl: DateTime / Mongo::Timestamp

PHP: MongoDB / MongoDBDatetime

* 无特殊需求的Date可考虑以Int64类型存储

BSON

Binary Data

Perl: string 引用

```
my $default_avatar = "\xFF\xFE\xFF";  
$collection->insert({"avatar_file" => \ $string});
```

PHP:MongoBinData

* 4Mb/8Mb(1.8+) 限制

灵活和自由!

Create Database

No!

Create Table

No!

Alter Table

No!

轻松面对不同数据源，随时调整变化的需求

Table => Collection

Row => Document

Embed document

Embed document : Array/List/Hash

_id

Collection 唯一
可以是任意简单类型
不能是Array/List

插入或保存时空缺则自动生成

Object Id => perl: MongoDB::OID php: MongoId

Insert/Save

perl:

```
$db->user->create({name=>'ns'});  
$db->user->save({_id=>5,name=>'ns'});
```

SQL Injection ? 哥很淡定.

Update

```
>db.user.update({_id:5},{name:'ns','email':'xxx'});
```

和mysql不同，这里是替换完整记录

Upsert

有匹配记录更新否则插入新记录

简单省事

```
>db.user.update({_id:5},{_id:5,name:'ns',email:'xxx'},true)
```

注意 自定义_id,

In-place Updates

\$inc \$push \$pushAll \$pull \$pullAll \$addToSet
\$set

....

Atomic

这些modifier可以彼此混搭，但不能和普通Array/Hash混用（需要\$set）

Upsert & modifier

\$modifier 不可引用 _id

```
>db.art_track.update({ art_id:25,d:20110303 },  
{ $addToSet:{follower:234},$inc:{views:1}},true);
```

```
> db.art_track.find()  
{ "_id" : ObjectId("4d6df20cb7fc9b3c1329c917"), "art_id" : 25, "d" : 20110303, "follower" :  
[ 234 ], "views" : 2 }
```

Select => Find

```
>db.user.find({city:'beijing'})  
      .sort({created_on:-1})  
      .limit(20);
```


Cursor

when find & run command

`cursor->next`

OP_QUERY —————> OP_GETMORE

返回结果集的都隐式创建游标
服务端会清理关闭cursor

Order by => Sort

Sort和Limit在Cursor执行前都可以更改

Joins ? No!

嵌入子文档

建立外键链接

听了前面的讲座大家应该都懂了...

Query Modifier

查询条件修饰符:

\$ne, \$in, \$nin, \$mod, \$all, \$size, \$exists, \$type, ..

\$lt, \$lte, \$gt, \$gte, \$ne,

...

Command

大多数非CRUD操作都是command:
count,map/reduce, group,...

```
>db.$cmd.findOne(cmd_query_obj);
```

```
db.$cmd.findOne('user') == db.user.count()
```

FindAndModify

```
db.runCommand( { findAndModify : <collection>,  
                query: { filter },  
                sort : { },  
                update: {},  
                field: {},  
                new/remove: true/false  
              });
```

FindAndModify

#perl 从队列中获取一个任务

```
my $job = db_find_and_modify {  
    query => { state => 0 },  
    update => { '$set' => { state => 1, ts =>  
time } },  
    sort => { _id => 1 }  
};
```

#模拟Mysql Auto increment

<?php

```
$db->command(array(  
    'findandmodify'=>'sequences',  
    'query' => array('_id'=>'user_id'),  
    'update' => array('$inc'=>array('val'=>1)),  
    'new' => true,  
    'upsert' => true,  
));  
?>
```

*update/sort,注意索引!

List commands

```
> db.runCommand({listCommands:1})
```

Quick reference card

<http://www.10gen.com/reference>

Index

```
db.foo.ensureIndex({a:1})
```

```
db.foo.ensureIndex({a:1,b:-1})
```

单个collection 最多有64个index

单个query只会选择1个index

Multikeys

对Array/Object进行索引

```
>db.art.ensureIndex({tags:1});  
>db.art.find({tags:'插画'}).sort({created_on:-1}).limit(10);  
  
>db.user.ensureIndex({'profile.age':1});  
>db.user.find({"profile.age":{"$gte:18}});
```

Special index / 1.8+

Sparse Indexes

```
>db.user.ensureIndex({sina_account:1},{sparse:true});  
>db.user.findOne({sina_account:'xxx'});
```

Covered Indexes

```
>db.user.ensureIndex({_id:1,passport:1,state:1});  
>db.user.findOne({passport:'xxx'},{_id:0,passport:  
1,state:1});
```

Like %mongoDB% ?

```
> db.count({title:/mongodb/i});
```

Regex => index的问题

简单的全文检索

使用分词库分词

存入一个关键词的倒排表collection

```
db.fulltext.save({  
  index_name: [ word1,word2]  
  {希望辅助查询的其他属性}  
  ...  
})
```

检索结果

对查询短语分词

查找index_name中匹配记录

```
db.fulltext.find({index_name:{$all:  
    [w1,w2,w3]}})  
    .sort({updated_on:-1})
```

Sum/Group by => ?

Group command: 局限性+阻塞db

Map/Reduce: good, 非sharding 单线程

Alternate approach

Live: counter field =>\$inc

Backend service: 读取数据后汇总计算

=> Gearman: workers

ORM,easy.

PHP: Zend/Symfony/CI ...

Ruby:Mongoid

Perl: Mongoose MongoDBx

```

<?php
//定义一个model
class Lgk_Model_Art extends Lgk_Core_Model_Base {
    protected $collection = "art";
    protected $schema = array(
        'category_id' => 0,
        'tags' => array(),
        'fav_tags' => array(),
        ....
    );
    protected $required_fields = array('user_id','name');
    protected $int_fields = array('user_id','category_id','deleted','published','approved','private');
    protected $created_timestamp_fields = array('created_on','updated_on');
    protected $joins = array(
        'user' => array('user_id' => 'Lgk_Core_Model_User'),
        'assets' => array('assets' => 'Lgk_Core_Model_Asset'),
        'category' => array('category_id' => 'Lgk_Core_Model_Category'),
        'thumbnail_asset' => array('thumb_asset_id' => 'Lgk_Core_Model_Asset'),
    );
    protected function extra_extend_model_row(&$row) {
    }
?>
// 加载内联数据
public function load_joins($row) {
    ...
    foreach ($this->joins as $attribute => $definition) {
        list($pk_name,$model_class) = each($definition);
        if (isset($row[$pk_name])) {
            if (is_array($row[$pk_name])) {
                $row[$attribute] = &DoggyX_Model_Mapper::load_model_list($row[$pk_name],$model_class);
            }
            else {
                $row[$attribute] = &DoggyX_Model_Mapper::load_model($row[$pk_name],$model_class);
            }
        }
    }
    return $row;
}
}

```

GridFS

Just specification.

Not best but good.

维护简单 + 水平延展性

空间为代价

Write-once, read many

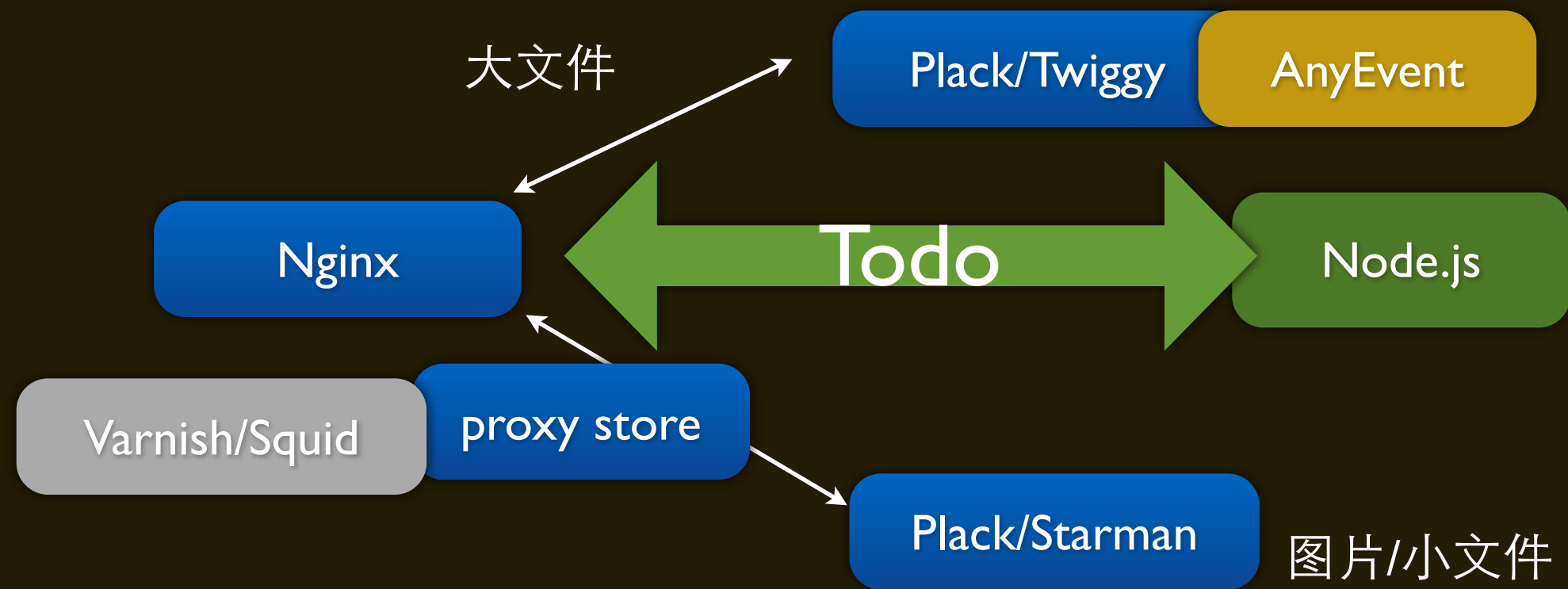
```
db.assets:{  
  _id:ObjectId,  
  file_id:ObjectId,  
  thumb_id:ObjectId,  
  ...  
}
```

```
db.fs.file: {  
  _id: ObjectId,  
  r: (ref counter)  
  hash:'xxxxxx'  
}
```

db.fs.files只保存文件1个副本
应用端删除只通知文件系统减少计数器r
应用端更新：删除，添加到files，更新file_id
定期清理垃圾文件

GridFS 部署

Nginx module ? 不够灵活，大文件有问题



启用ETag: file_id

Replication

```
local.system.replset  
local.oplog.rs => oplog/capped
```

* `local.usr.xxx` => 自定义你的本地collection

神马都是collection
直观，方便管理维护

从mySQL: mmm

ReplicaSet

1 Primary + 2 Secondary + 1
Arbiter

最小安全配置

* Master/Slave不再考虑

“SlaveOk”

```
<?php
$con =new Mongo('mongodb://s1:27017',array
('replicaSet'=>true);
$con->setSlaveOkay(true);
?>
```

```
#perl
my $con = MongoDB::Connection->new(host=>'mongodb://
localhost',w =>2,find_master => 1);
$MongoDB::Cursor::slave_okay = 1;
my $cursor = $con->user->find;
$cursor->slave_okay = 1;
```

Driver层实现读写分离

Auto Shard

之前:手动分片模式: 单独db存储配置

正在逐步迁移: mongos

shard_key: 谨慎选择, 不合适可能更糟
counting的小问题

WTF?

多数据中心/VPN

墙 => VPN的稳定性

延迟 & 墙 => 数据一致性差

尝试 自定义的oplog传输/replay,局限性

空间碎片

频繁的删除更新导致碎片

目前仍无法在线压缩 `db.repairDatabase`, 慢!

很土鳖的方法: 切换primary, 删除数据重新同步!

注意oplog预分配空间

Raid/Fs

RAID 10

XFS filesystem

kill -9 or 断电 = Crash!

Durability?

`--syncdelay = ? (60=>5)`

`'w'`

`db.runCommand({ getlasterror : 1 , w : 2 })`

确保ReplicaSet有2个secondary

`-dur upgrade to 1.8!`

Count is slow!

扫描记录过多

自定义counter属性

Type: string vs int

“2” != 2

```
<php? $user_id = (int) $user_id; ?>
```

OOM Killer

```
> db.serverStatus()
```

Swap 分区 > Memory

索引不起作用

Ordered hash?

Perl: `IxHash`

`({a=>1, b=>-1}) => IxHash->new`

Ruby (<1.9): `BSON::OrderedHash`

PHP: `Array` is OK

Cursor模式分页

```
        last_row_id = ObjectId('....');  
db.activity_stream->find({_id:{$lt: last_row_id },  
        user_id:20 } ).sort( {_id:-1} ).limit(10);
```

我们的生产环境

PHP/5.2.x: front-end

Perl 5.2.10+ / Plack / Gearman:

Daemon, large-file upload, data
process, job queue service ...

MongoDB 用于:

session store

business data

GridFS: Media files (user uploaded
files, thumbnails ..)

MongoX - DSL语法糖

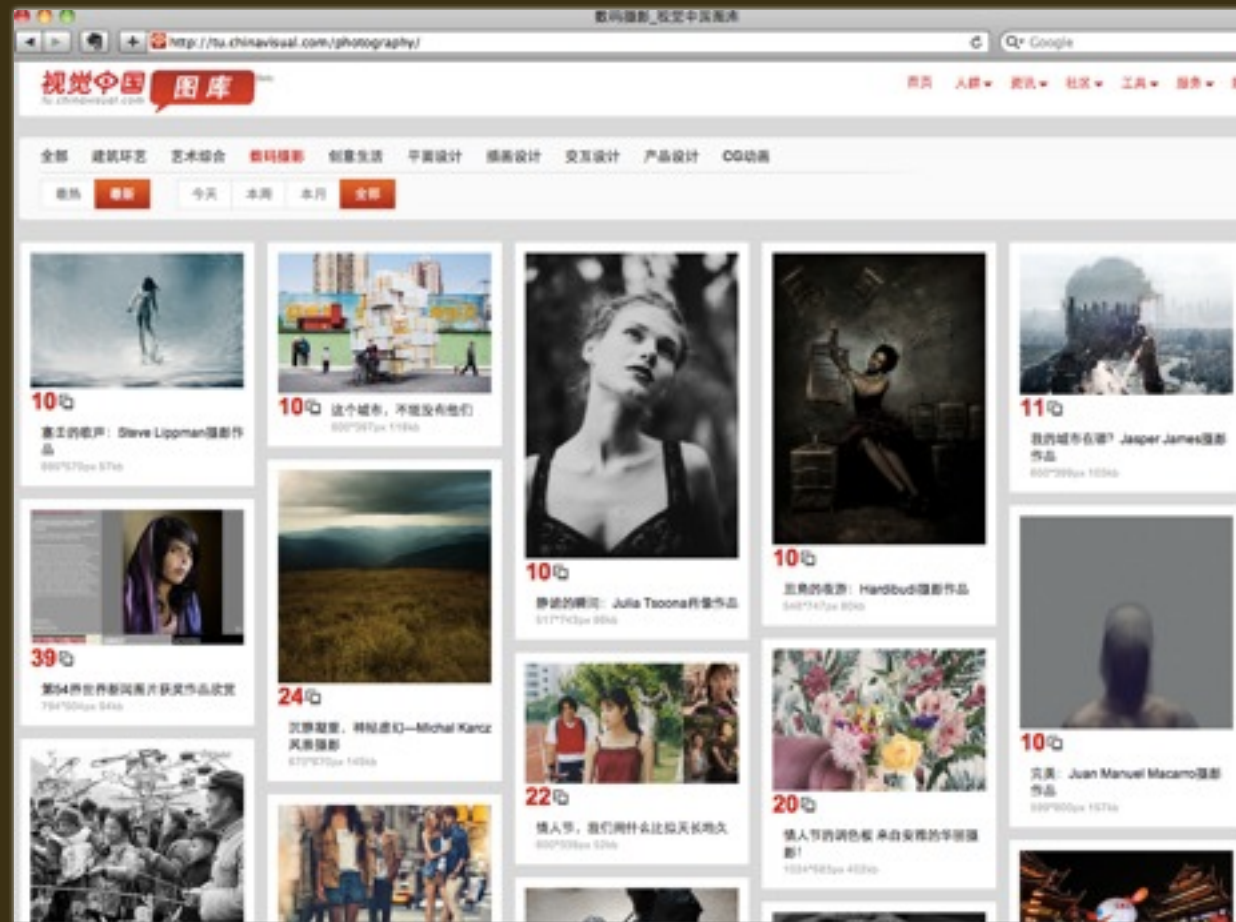
```
use MongoX (host=>'mongodb://127.0.0.1',db=>'test');
use_collection 'task_queue';
db_update { state => 1, queue => 'sina_tweet' }, {
    '$set' => { state => 0 },
    '$inc' => { tries => 1 },
    },{ multiple => 1 };
db_remove { state => 1, queue => 'sina_tweet', 'tries' =>
{ '$gt' => 3 } };
db_inc {}
```

```
//循环统计几个不同数据库
my $cnt = 0;
for_connections {
    for_dbs {
        for_collections {
            $cnt += db_count;
        } 'art'
    } 'db1','db2'
} 'c_1' , 'c_2', 'c_arch1', 'c_arch2';
```

<http://github.com/nightsailer/mongo-x/>

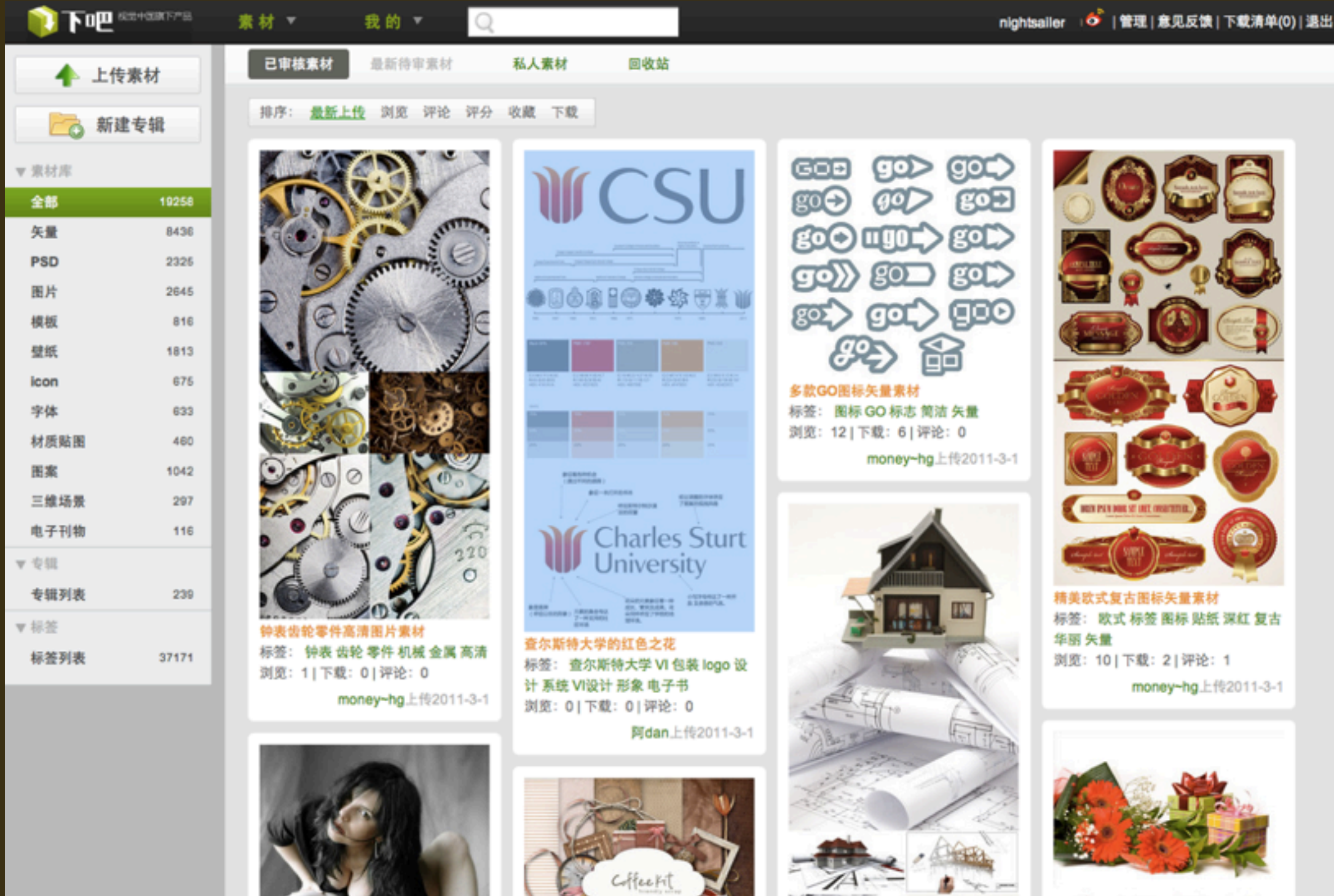


原创榜:
<http://czone.chinavisual.com/>
2009-6 rebuild on mongodb
< 1.0



短网址 <http://cvurl.cn/>

图库:
<http://tu.chinavisual.com/>

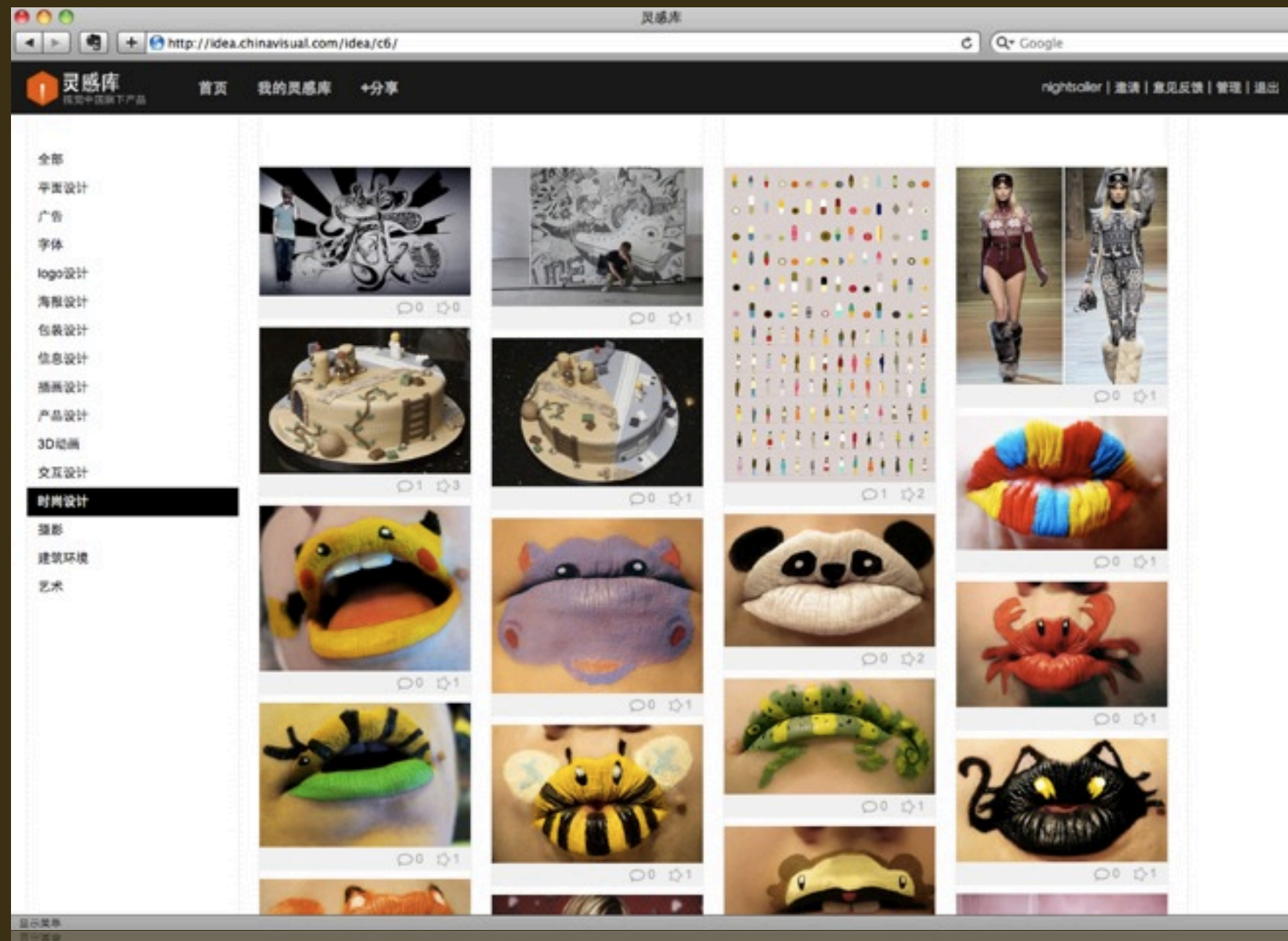


下吧:

<http://down.chinavisual.com/>

mongoDB:

GridFS: high resolution
images, photos/medium
size files(60mb~500mb)



灵感库：在线图片分享社区
<http://idea.chinavisual.com/>

我们在寻找

PHP 开发工程师

job-php@chinavisual.com

Modern Perl 开发工程师

job-perl@chinavisual.com

mongoDB / Git / Catalyst /
Ubuntu / Xiapian / Nginx / Plack /
Moose / Node.js / Titanium / Redis