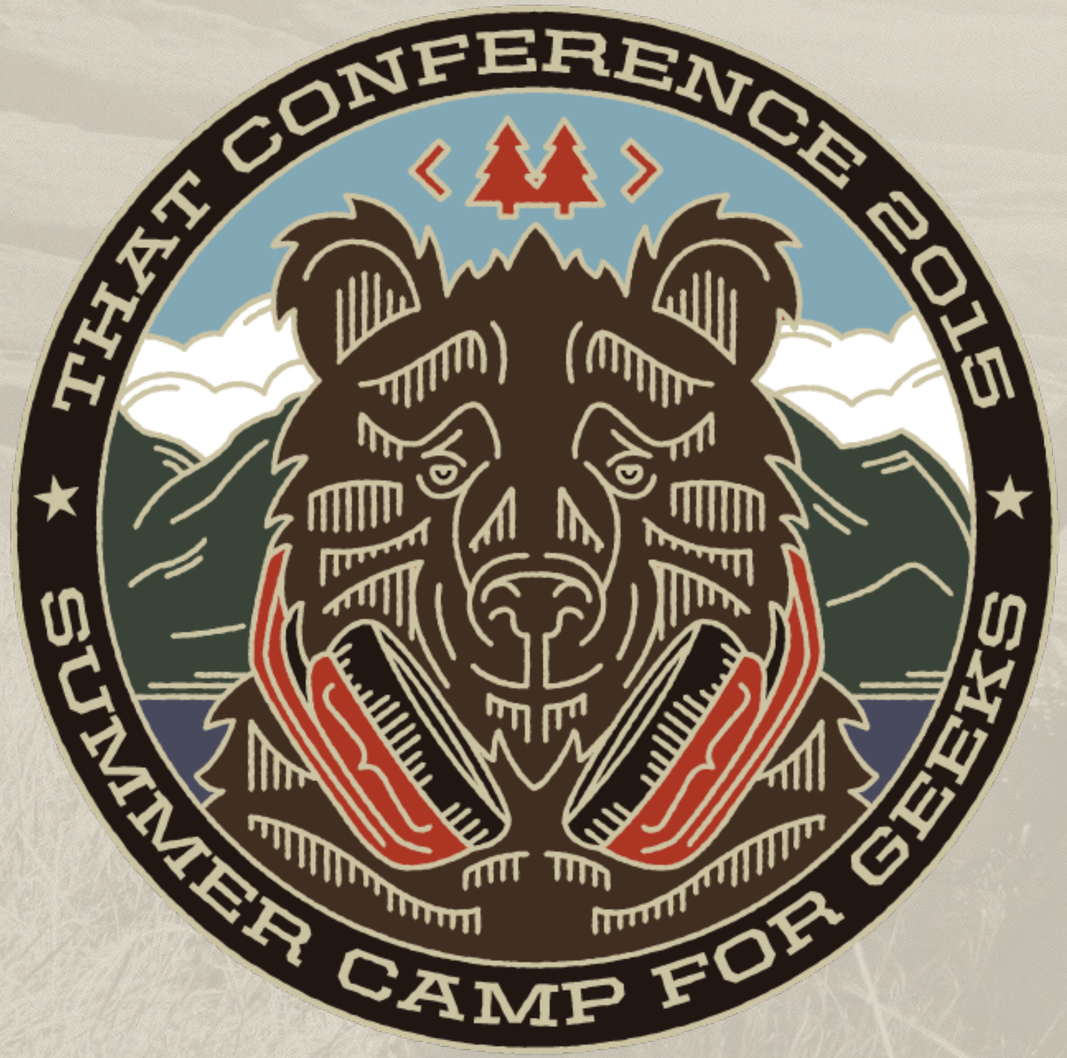
Amazon SimpleDB

By: Eli White

CTO & Founding Partner:
musketeers.me

Managing Editor & Conference Chair:
php[architect] - www.phparch.com

eliw.com - @eliw


THAT CONFERENCE 2015
SUMMER CAMP FOR GEEKS


Microsoft
REDARROW
A DOHMEN COMPANY
inrule
TECHNOLOGY®
IBM®
COLLABORATIVE
CDSC
COLLABORATIVE DOMESTIC SOLUTION CENTER
SafeNet
Consulting That Moves You
API®
HEALTHCARE
A GE Healthcare Company
RENAISSANCE
LEARNING
Beacon
Lighting a clear path for your success
WEST BEND®
runzheimer
INTERNATIONAL
COMPLY365
Concurrency
tegile

What is NoSQL?

And what's wrong with SQL anyway?

Let's define it

Wikipedia says:

NoSQL is a class of database management system identified by its non-adherence to the widely used relational database management system (RDBMS) model.

Wait, what's that mean?

Differences between RDBMS:

Does not use SQL as its query language.

Probably

May not give full ACID guarantees.

Usually

Uses distributed, fault-tolerant architecture.

Sometimes

I'm confused!

Easier definition -
Not one of these:

Me too!

Oracle

MySQL

SQLite

DB2

Sybase

Postgres

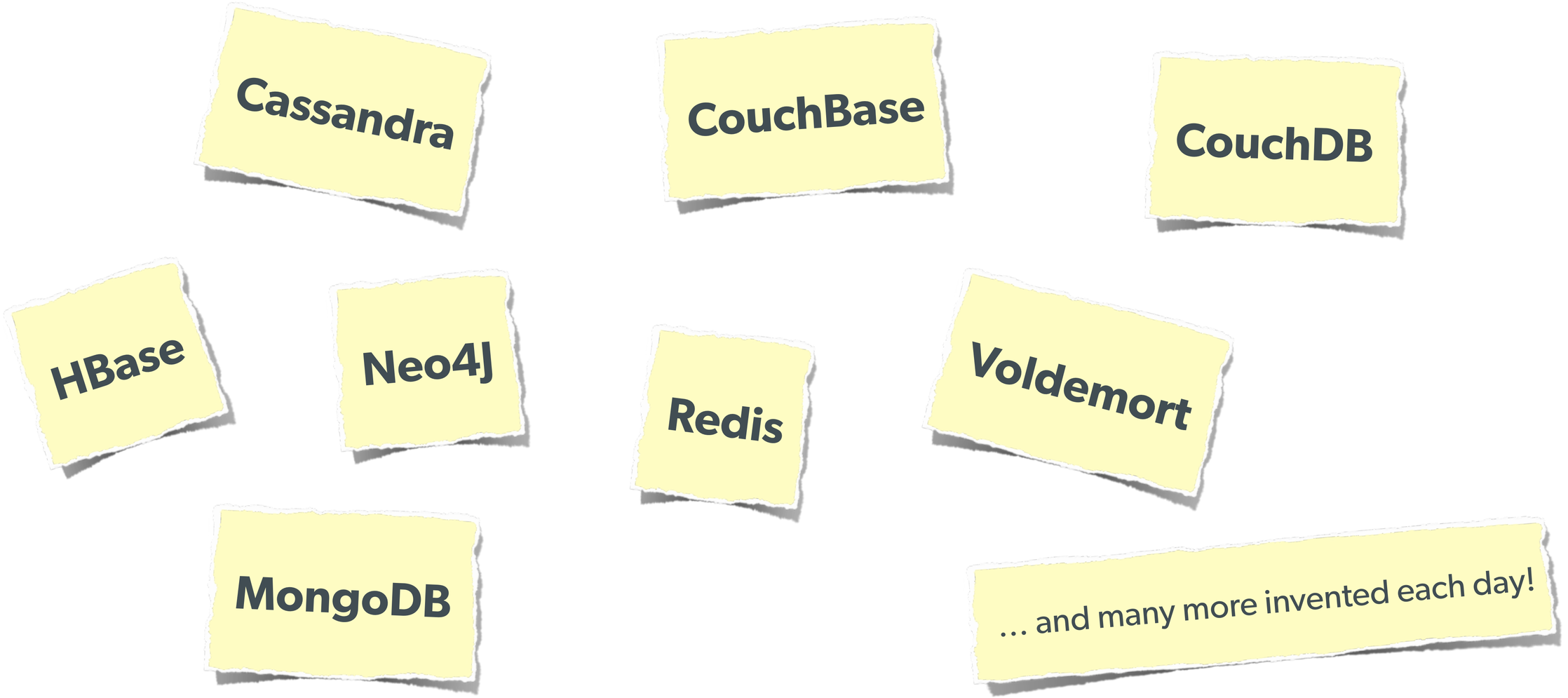
SQLServer

... and others like them.

Other NoSQL Options

They could have been contenders ...

To be fair, there are other options…



Why so many?

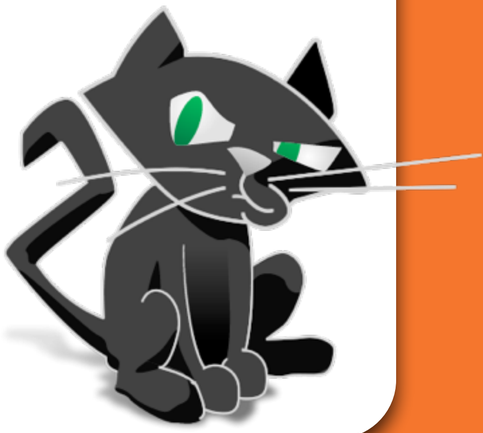
Every single one is different, very much so

- Document Stores
- Object Stores
- Column Stores
- Key-Value Stores
- Graph Storage

- Availability
- Distributed
- Fault Tolerance
- Consistency

Amazon SimpleDB Basics

Who is this ‘Amazon’ person you keep talking about?



php[architect]

Usage

Cloud Hosted on Amazon Web Services

Access from anywhere

Though connections from EC2 are best …

Pricing

Example rates per month for US East

They charge you 3 ways

Machine Utilization
- 1st 25 machine hours are free
- $0.140 per hour afterwards

Data Transfer Out
- 1 GB is free
- Up to 10 TB = $0.090 per GB
- Next 40 TB = $0.085 per GB
- Next 100 TB = $0.070 per GB
- (and it keeps going down from there)

Data Storage
- 1 GB for free
- $0.250 per GB afterwards

php[architect]

Greatest Strengths

Speaking as a PHP programmer …

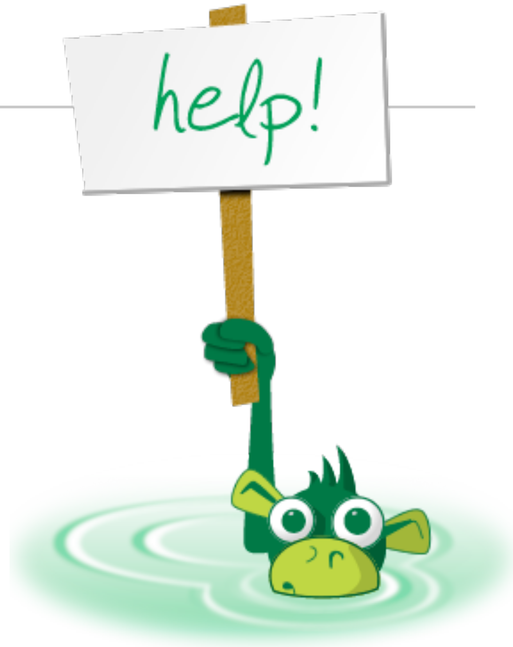
No operations overhead

Automatically scales

Amazingly simple PHP SDK

SimpleDB Details

So how does Amazon do ‘NoSQL’ ...



The Data Model

Domain --> Item --> Attributes

Domains
Think of a 'domain' as a 'table'

Items
These are the equivalent of rows

Attributes
Similar to columns, but unlimited

Example

Any number of attributes, all optional

ID	Type	Name	Gender	Color	Size	Inseam	Width
1	shirt	Ringer T	Unisex	blue, red, gray	S, M, L, XL		
2	shirt	Women's Cut T	Women	white, green	M, L, XL, 2X		
3	shoe	Track Runner	Male	white	8, 9, 10, 11, 11.5		D, EE
4	shoe	Classic Leather	Male	black, brown	7, 7.5, 7, 8.5		D
5	pants	Blue Jeans	Male		34, 36, 38, 40	28, 29, 30	
6	pants	Boot Cut Jeans	Female	blue, black	6, 8, 10, 12, 14		
7	belt	Leather Belt	male	brown	32-36, 34-38		

Things to keep in mind

No concept of relations at all

Everything is a text field

Requests are ‘eventually consistent’, unless specified otherwise

Everything is a text field?

Yes!

Highly optimized for text searches

Bit of a pain for numerical data (sorting):

- Zero pad your integers (0065, 0067, 0076)
- Negative offset (add a set number to all, like 100000)
- Dates: use ISO 8601 which supports string sorting - YYYY-MM-DDThh:mm:ss.sTZD

Selecting Data

Uses a query language looking suspiciously like SQL

Basic Format:

```
select <attribute_list>
from <domain>
[where expression]
[sort_instructions]
[limit limit]
```

Simple Queries

A basic query against a single attribute is very simple:

Examples:

```
select * from Products where size = ‘M’
select * from Products where size > ‘38’
select * from Products where name like ‘%boot%’
```

Range Queries

You can give multiple queries on the same attribute:

Special every() keyword makes an exclusive match.

Examples:

```
select * from Products where size < ‘42’ and size > ‘30’
select * from Products where width = ‘D’ or width = ‘EE’
select * from Products where every(size) in (‘S’,‘M’,‘L’)
```

An ‘and’ query wouldn’t work as expected on the width example. Each value is compared individually, no one value can be 2 values. Use intersection sets instead.

Multiple Attribute Queries

Use intersection to query against multiple attributes:

Examples:

```
select * from Products where size = ‘42’ intersection gender = ‘Male’
select * from Products where width = ‘D’ intersection width = ‘EE’
select * from Products where width = ‘EE’ intersection size in (‘8’,‘9’,’10’)
```

Other bits

Your ID column is accessed by itemName()

Sorting uses standard SQL syntax

But fails if there are any NULL values

The only aggregate method is count(*)

Black Magic

Optimization via composite attributes

Possible because of highly optimized text searches

Combine two attributes into one, then query

Composite Example

If you store your size & gender data together in one field

Original:

```
select * from Products where size = ‘42’ intersection gender = ‘Male’
```

Composite:

```
select * from Products where gendersize = ‘Male|42’
select * from Products where gendersize like ‘Male|%’
```

API Usage

Otherwise known as “Using the PHP SDK” ...

Getting the SDK & Documentation

http://aws.amazon.com/sdk-for-php/

http://aws.amazon.com/documentation/sdk-for-php/

Basic Usage

After installing it, paste your Access Key & Secret Key into the configuration file

Getting Started:

```
// Initialize AWS
use Aws\Common\Aws;
$aws = Aws::factory('/path/to/my_config.file');

// Retrieve the SimpleDB client:
$client = $aws->get('SimpleDb');
```

Creating a Domain:

```
$client->createDomain(['DomainName' => 'thatconference']);
```

Putting data

You have the option (4th parameter) whether to replace, or add attributes, in case of the duplicate Attribute

Data storage syntax:

```
$client->putAttributes([
    'DomainName' => 'thatconference',
    'ItemName'   => 'geeklingshirts',
    'Attributes' => [
                    [ 'Name' => 'color', 'Value' => 'red', 'Replace' => true ],
                    [ 'Name' => 'type', 'Value' => 'tshirt', 'Replace' => true ],
                    [ 'Name' => 'size', 'Value' => 'YL' ],
                    [ 'Name' => 'size', 'Value' => 'YM' ],
                    [ 'Name' => 'size', 'Value' => 'YS' ],
                    ]
    ]);
```

Querying

Use getIterator('Select') for easy queries:

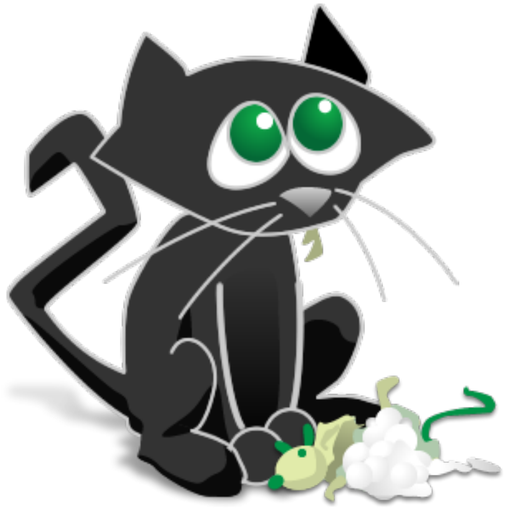
Query syntax:

```
$iterator = $client->getIterator('Select', array(
    'SelectExpression' => "select * from thatconference where color = 'red'"
    ));

foreach ($iterator as $item) {
    echo "\n\n", $item['Name'], ":\n";
    foreach ($item['Attributes'] as $attr) {
       echo "  {$attr['Name']}: {$attr['Value']}\n";
    }
}
```

Limits & Numbers

Just so you know …



Limits

A number of limitations to work within

Parameter	Restriction
Domain Size	10 GB, 1 billion attributes
Domains per account	250
Attribute name-value pairs per item	256
Attribute value length	1024 bytes
Attributes requested per select	256
Maximum items in response per select	2500
Maximum query time	5 seconds
Maximum number of comparisons per select	20
Maximum response size per select	1 MB

DynamoDB

Is that a gorilla standing in the corner?

DynamoDB comes into view

The newest/greatest from Amazon (3 years ago)

Very similar in structure to SimpleDB

Has a huge performance increase

With two large catch-22s:

- Has 'provisioned' scaling
- Requires queries by primary key and a single predefined range

Questions?

For this presentation & more:
http://eliw.com/

Twitter: @EliW

php[architect]: https://www.phparch.com/
musketeers: http://musketeers.me/