

Dynamodb Demo

yupeng

05/23/14

Contents

- 1 Introduce
- 2 Deploy
- 3 Demo
- 4 More details

Main Sections

1 Introduce

2 Deploy

3 Demo

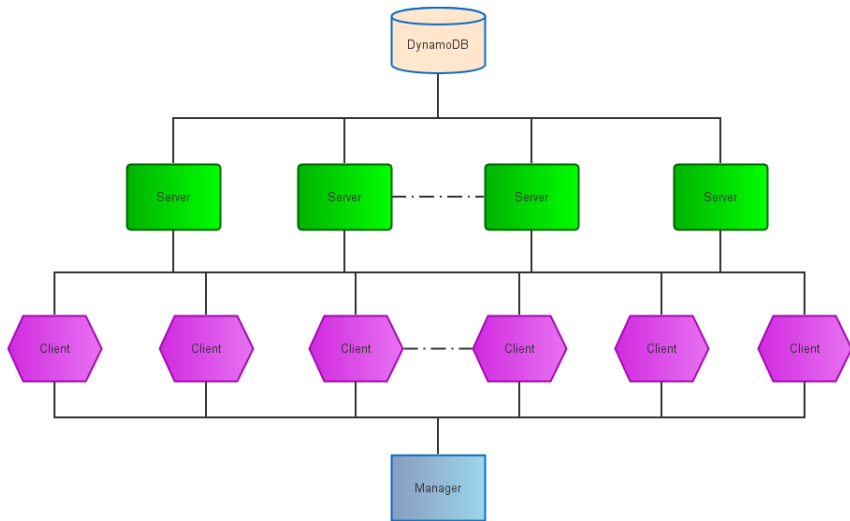
4 More details

Architecture summary

The demo has 4 components.

- A dynamodb, store 10M lines data. The data schema is “name(string) date(string) score(int)”. It simulates a gaming database. Every user has a name, and many gaming scores ranged by date.
- Servers, simulate a gaming web server, provide a web interface to access the dynamodb
- Clients, using http_load generate workload to servers, http_load is an opensource web benchmark tool, we use it examine the servers' respond latency.
- Manager, provide an interface to user, send command to clients, let clients generate workload by using http_load, and get the http_load result.

Architecture Picture



Main Sections

1 Introduce

2 Deploy

3 Demo

4 More details

Cloudformation template

The demo environment can be deployed from a cloudformation template, the template address is: [dynamodb_demo_link](#)

Launch template

Launch the template in the cloudformation console:

Select Template

Specify Parameters

Options

Review

Select Template

Specify a stack name and then select the template that describes the stack that you want to create.

Stack

An **AWS CloudFormation stack** is a collection of related resources that you provision and update as a single unit.

Name

Template

A template is a JSON-formatted text file that describes your stack's resources and their properties. AWS CloudFormation stores the stack's template in an Amazon S3 bucket. [Learn more](#).

Source ☒ Select a sample template

☐ Upload a template to Amazon S3

浏览...

 未选择文件。

☒ Specify an Amazon S3 template URL

Template parameters

The template has several parameters. Most of them have default values. Only several parameters need to be set:

- KeyName, the keypair used for launch ec2 instance
- ReadCapacityUnits and WriteCapacityUnits, the dynamodb read/write capacity
- ServerInstanceNumber and ClientInstanceNumber, if the ReadCapacityUnits and WriteCapacityUnits are 2000, set ServerInstanceNumber to 4, set ClientInstanceNumber to 8, if the ReadCapacityUnits and WriteCapacityUnits are 4000, set ServerInstanceNumber to 8, set ClientInstanceNumber to 16

Parameter graph

Select Template

Specify Parameters

Options

Review

Specify Parameters

Specify values or use the default values for the parameters that are associated with your AWS CloudFormation template.

Parameters

ClientInstanceNumber	<input type="text" value="8"/>	client instance number
ClientInstanceType	<input type="text" value="m3.large"/>	client instance type
ConcurrentNumber	<input type="text" value="500"/>	concurrent per client
CountPerUser	<input type="text" value="10"/>	every user name has how many items
KeyName	<input type="text"/>	key pair for all ec2 instances
ManagerInstanceType	<input type="text" value="m1.small"/>	manager instance type
NameDB	<input type="text" value="name_1M.db"/>	sqlite db store name
ReadCapacityUnits	<input type="text" value="2000"/>	dynamodb read capacity
ResourceLink	<input type="text" value="https://s3-us-west-2.amazonaws.com"/>	the zip file for all source code and data
ServerInstanceNumber	<input type="text" value="4"/>	server instance number
ServerInstanceType	<input type="text" value="c3.2xlarge"/>	server instance type
UriNumber	<input type="text" value="2000"/>	how many uri in uri list for http_load
WriteCapacityUnits	<input type="text" value="2000"/>	dynamodb write capacity

Cancel

Previous

Next

Check cloudformation Outputs

After the cloudformation launched complete, check the manager instance IP address in its Outputs

Create Stack Update Stack Delete Stack

Filter: Active ▾

By Name:

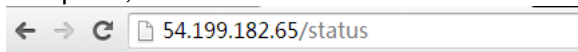
	Stack Name	Created Time	Status	Description
<input checked="" type="checkbox"/>	testdemo	2014-05-22 18:53:46 UTC+0800	CREATE_COMPLETE	create dynamodb demo environ

Overview Outputs Resources Events Template Parameters Tags Stack Policy

Key	Value	Description
ManagerIP	54.199.182.65	The manager ip address

Check <http://ManagerIP/status>

After the cloudformation launched, the manager instance will start to insert data to dynamodb, the <http://ManagerIP/status> indicate how many items are inserted. Seeing 'done' in that page indicate the insert complete, and we can start to demo.



1000000
done

Main Sections

1 Introduce

2 Deploy

3 Demo

4 More details

Table information

See the table information in the dynamodb console, examine the item count:

The screenshot displays the Amazon DynamoDB console interface. At the top, the 'Amazon DynamoDB Tables' header is visible, followed by a 'Filter:' input field and several action buttons: 'Explore Table', 'Create Table', 'Modify Throughput', 'Delete Table', 'Export / Import', and 'Access Control'. Below this is a table listing the available tables. The table has three columns: 'Name', 'Status', and 'Hash Key'. A single table, 'testdemo-DynamoDB-HUX432WA0WS4', is listed with a status of 'ACTIVE' and a hash key of 'name'.

Below the table list, the 'Table Items' section is expanded, showing the 'Details' tab. This section provides comprehensive information about the selected table, 'game_alpha'. The details are organized into a key-value format with dotted lines separating the labels from the values.

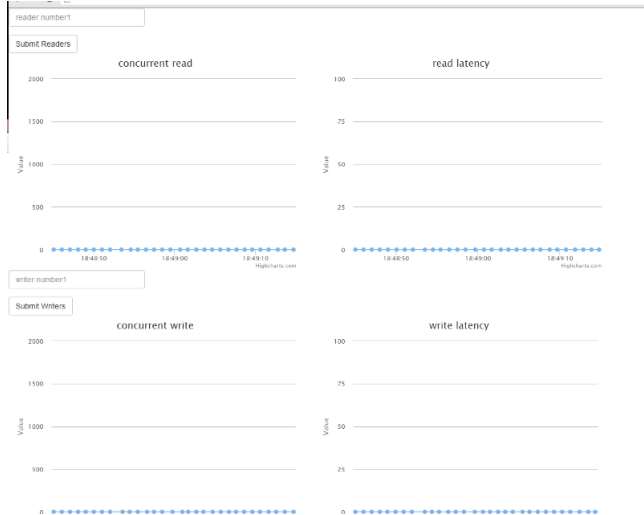
Name	Status	Hash Key
testdemo-DynamoDB-HUX432WA0WS4	ACTIVE	name

Table Items	
Details	Indexes
Table Name:	game_alpha
Primary Hash Key:	name (String)
Primary Range Key:	date (String)
Table Status:	ACTIVE
Creation Date:	Sun May 04 16:40:12 GMT+800 2014
Provisioned Read Capacity Units:	2000
Provisioned Write Capacity Units:	2000
Region:	Asia Pacific (Tokyo)
Amazon Resource Name (ARN):	arn:aws:dynamodb:ap-northeast-1:504215341699:table/game_alpha
Last Decrease Time:	
Last Increase Time:	
Storage Size (in bytes)*:	431703465
Item Count*:	10000000

* Storage size and Item count are not updated in real-time. Instead, they are updated periodically, roughly every

Manager Interface

Input the manager instance's public in the web browser, you can get such an interface:



Generate read workload

Input 500, 1000, 1500, 2000 to the first form, and type “Submit Readers” can generate 500, 1000, 1500, 2000 concurrent reads to the dynamodb. The smallest grant size is 500, you can't set to other value. The graph in the left top corner shows the concurrent read amount, and the graph in the right top corner shows the average read latency from the client perspective. Input 0 and submit the form will stop the workload.

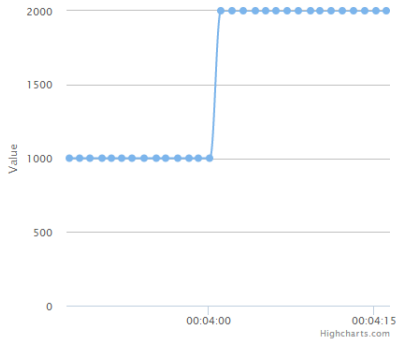
Read workload screenshot

← → ↻ 54.199.182.65

2000

Submit Readers

concurrent read



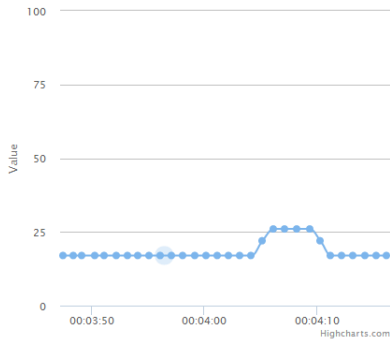
0

Submit Writers

concurrent write

2000

read latency



write latency

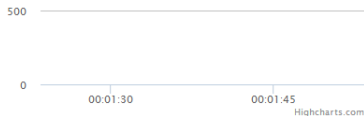
100

Generate write workload

Input 500, 1000, 1500, 2000 to the second form, and type “Submit Writers” can generate 500, 1000, 1500, 2000 concurrent write to the dynamodb. The smallest grant size is 500, you can't set to other value. The graph in the left bottom corner show the concurrent write amount, and the graph in the right bottom corner show the average write latency from the client perspective.

Write workload screenshot

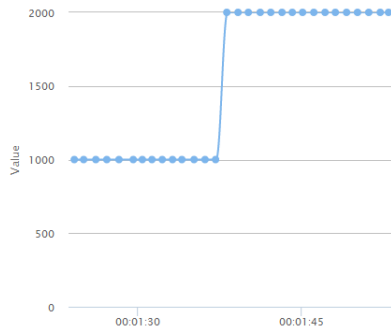
← → ↻ 54.199.182.65



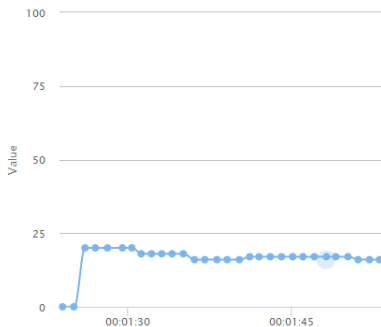
2000

Submit Writers

concurrent write



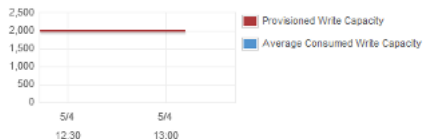
write latency



The dynamodb latency

Check the dynamodb latency in cloudwatch

Write Capacity Units/Second - 5 Minute Average



Throttled Write Requests Count



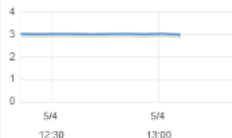
Get Latency Milliseconds



Put Latency Milliseconds



Query Latency Milliseconds



Scan Latency Milliseconds

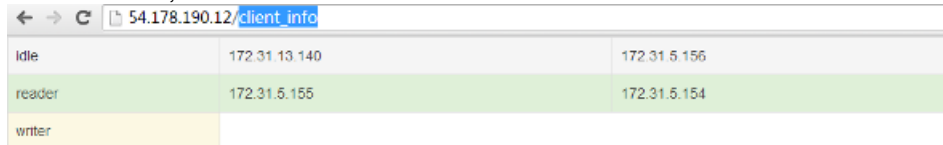


User Errors Count



The http_load raw result

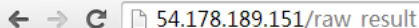
Check the server_ip/client_info page, we can see the ip address of the reader client, writer client and idle client




← → ↻	54.178.190.12/client_info	
idle	172.31.13.140	172.31.5.156
reader	172.31.5.155	172.31.5.154
writer		

Check http_load raw result from client

From the ip address in server_ip/client_info page, we can choose a client, access the client_ip/raw_result page:

A screenshot of a web browser's address bar. It features navigation icons (back, forward, refresh) on the left and a text input field containing the URL "54.178.189.151/raw_result".

← → ↻  54.178.189.151/raw_result

2490 fetches, 23 max parallel, 87714 bytes, in 5.00002 seconds

35.2265 mean bytes/connection

497.998 fetches/sec, 17542.7 bytes/sec

msecs/connect: 0.383643 mean, 12.709 max, 0.255 min

msecs/first-response: 18.1026 mean, 76.067 max, 5.378 min

HTTP response codes:

code 200 -- 2490

Main Sections

1 Introduce

2 Deploy

3 Demo

4 More details

The demo source code

You can get the source code from github:

https://github.com/yupeng820921/dynamodb_demo.git

I also upload it to the S3:

<https://s3-us-west->

[2.amazonaws.com/yupengpublic/dynamodb_demo.zip](https://s3-us-west-2.amazonaws.com/yupengpublic/dynamodb_demo.zip)

The cloudformation has a parameter “ResourceLink”, it indicate where the server/client/manager instance should get the source code. You can modify the source code and upload to another place, and set the “ResourceLink” to another value.

Manage name

In the source code directory, we have a `name_1M.db` and `name_20K.db`. The `name_1M.db` have 1M random name, and `name_20K.db` have 20K random name. By default, manager will use `name_1M.db`, upload 10 “name(string) date(string) score(int)” items per name. If you want to change the name count, you can use the `generate_name.py` generate a new name db, store it to the source code directory, upload the new source code directory to S3, specific the filename of the “NameDB” parameter in cloudformation template. By default, the manager will upload 10 items per name, if you want to change it, you can assign a new value to “CountPerUser” parameter.

http_load parameters

We call http_load by such parameters:

```
http_load -p concurrent_number -r concurrent_number -s  
interval url_list.txt
```

ConcurrentNumber and Interval parameters in the template will be pass to the above command, and the UrlNumber parameter will be used to specific how many urls will be placed in a url_list.txt file.

Th