РАЗПРЕДЕЛЕНИ ПРИЛОЖЕНИЯ

Павел Кюркчиев
Ас. към ПУ „Паисий Хилендарски“
@pkyurkchiev

CLOUD DATABASES

Какво са облачно базираните бази данни?

- Cloud database е база данни работеща на облачно базирана платформа, достъпа до нея се осъществява чрез уеб услуги. База данни като услуга се грижи за мащабируемостта и достъпността до данните.

Типове на база модел на работа

- SQL база данни
- NoSQL база данни

Типове на база модел на предоставяне

- Вирутални машини
- База данни като услуга(Database-as-a-service (DBaaS))

Виртуална машина

- Облачните платформи позволяват на потребителите да купуват копия на виртуални машини за ограничен период от време, на които могат да се стартират база данни. Потребителите могат да избират между добавянето на собствени копия или използването на вече готови копия на системи с оптимизирани версии на база данни.

Примери

- SQL Server
- Oracle SQL
- MySQL
- MongoDB
- Cassandra database

База данни като услуга(Database-as-a-service (DBaaS))

- С база данни като услуга собствениците на приложения не са длъжни сами да инсталират и поддържат базата данни. Вместо това, доставчикът на услуги за база данни поема отговорността за инсталирането и поддръжката на базата данни, а собствениците на приложения се таксуват в зависимост от използването на услугата.

Примери

- Amazon Relational Database Service
- Clustrix Database as a Service
- EnterpriseDB Postgres Plus Cloud Database
- Microsoft Azure SQL Database
- **Amazon DynamoDB**
- Amazon SimpleDB
- Azure Cosmos DB
- Cloudant Data Layer
- **Firebase database**

Amazon DynamoDB

- Amazon DynamoDB е напълно управляема NoSQL база данни услуга, която се предлага от Amazon.com като част от Amazon Web Service portfolio;
- DynamoDB е NoSQL документно – ориентирана база данни от тип key – value;
- За пръв път Amazon DynamoDB е представена през 2012 година от Amazon CTO Werner Vogels;
- DynamoDB използва асинхронни репликации между голям брой дата центрове за висока издръжливост и достъпност;

Производителност и параметри

- Requests and throttling
- Errors: ConditionalCheckFailedRequests, UserErrors, SystemErrors
- Metrics related to Global Secondary Index creation

Езици и рамки за разработка подържани от DynamoDB

- Java, Node.js, Go, C# .NET, Perl, PHP, Python, Ruby, Haskell and Erlang.

```
using Amazon.DynamoDBv2;
using Amazon.DynamoDBv2.DocumentModel;
using Amazon.Runtime;

private static AmazonDynamoDBClient client = new
AmazonDynamoDBClient();
private static string tableName = "ProductCatalog";
```

Връзка с AWS SDK и създаване на обект DB

.Net

Създаване на таблица - Create

```
var request = new CreateTableRequest
{
  AttributeDefinitions = new List<AttributeDefinition>()
  {
    new AttributeDefinition
    {
      AttributeName = "Id",
      AttributeType = "N"
    },
    new AttributeDefinition
    {
      AttributeName = "ReplyDateTime",
      AttributeType = "N"
    }
  },
```

Деклариране на таблица

```
KeySchema = new List<KeySchemaElement>
{
  new KeySchemaElement
  {
    AttributeName = "Id",
    KeyType = "HASH" //Partition key
  },
  new KeySchemaElement
  {
    AttributeName = "ReplyDateTime",
    KeyType = "RANGE" //Sort key
  }
},
```

Деклариране на таблица

.Net

```
 ProvisionedThroughput = new ProvisionedThroughput
 {
   ReadCapacityUnits = 5,
   WriteCapacityUnits = 6
 },
 TableName = "ProductCatalog"
};

var response = client.CreateTable(request);
```

Създаване на таблица

.Net

Създаване на запис - Put

```
Table productCatalog = Table.LoadTable(client, tableName);
private static int sampleBookId = 555;
var book = new Document();
 book["Id"] = sampleBookId;
 book["Title"] = "Book " + sampleBookId;
 book["Price"] = 19.99;
 book["ISBN"] = "111-1111111111";
 book["Authors"] = new List<string>
                   { "Author 1", "Author 2", "Author 3" };
 book["PageCount"] = 500;
 book["Dimensions"] = "8.5x11x.5";
 book["InPublication"] = new DynamoDBBool(true);
 book["InStock"] = new DynamoDBBool(false);
 book["QuantityOnHand"] = 0;
productCatalog.PutItem(book);
```

Създаване на запис

Извличане на запис - Get

```
Table productCatalog = Table.LoadTable(client, tableName);
GetItemOperationConfig config = new GetItemOperationConfig
{
  AttributesToGet = new List<string>
          { "Id", "ISBN", "Title", "Authors", "Price" },
  ConsistentRead = true
};
Document document = productCatalog.GetItem(sampleBookId,
config);
Console.WriteLine("RetrieveBook: Printing book retrieved...");
```

Извличане

Промяна на запис - Update

```
Table productCatalog = Table.LoadTable(client, tableName);
int partitionKey = sampleBookId;
var book = new Document();
 book["Id"] = partitionKey;
 book["Price"] = 29.99;

// For conditional price update, creating a condition
expression.
Expression expr = new Expression();
expr.ExpressionStatement = "Price = :val";
expr.ExpressionAttributeValues[":val"] = 19.00;
```

Декларация

.Net

```
// Optional parameters.
UpdateItemOperationConfig config = new
UpdateItemOperationConfig
 {
   ConditionalExpression = expr,
   ReturnValues = ReturnValues.AllNewAttributes
 };
Document updatedBook = productCatalog.UpdateItem(book,
config);
```

Промяна и извличане

Премахване на запис - Delete

```
Table productCatalog = Table.LoadTable(client, tableName);
// Optional configuration.
DeleteItemOperationConfig config = new
DeleteItemOperationConfig
 {
   // Return the deleted item.
   ReturnValues = ReturnValues.AllOldAttributes
 };
Document document = productCatalog.DeleteItem(sampleBookId,
config);
```

Изтриване

.Net

Търсене – Query and Scan

```
Table table = Table.LoadTable(client, "Reply");
DateTime twoWeeksAgoDate = DateTime.UtcNow -
TimeSpan.FromDays(15);
QueryOperationConfig config = new QueryOperationConfig()
{
  HashKey = "DynamoDB Thread 2", //Partition key
  AttributesToGet = new List<string>
    { "Subject", "ReplyDateTime", "PostedBy" },
  ConsistentRead = true,
  Filter = new RangeFilter(QueryOperator.GreaterThan,
twoWeeksAgoDate)
};

Search search = table.Query(config);
```

Извличане на резултати - Query

```
string tableName = "Thread";
Table ThreadTable = Table.LoadTable(client, tableName);

ScanFilter scanFilter = new ScanFilter();
scanFilter.AddCondition("ForumId", ScanOperator.Equal, 101);
scanFilter.AddCondition("Tags", ScanOperator.Contains,
"sortkey");

Search search = ThreadTable.Scan(scanFilter);
```

Извличане на резултати - Scan

Много моделни бази данни (**Multi-model database**)

- Много моделните бази данни са бази от данни, които могат да съхраняват, индексират и извличат информация от повече от един модел на данни. До преди няколко години базите от данни можеха да работят само с един модел на данни: relational database, document-oriented database, graph database or triplestore. Бази от данни, които могат да комбинират няколко от тези модели се наричат много модели.

Много моделни бази данни

- **ArangoDB** – document (JSON), graph, key-value
- **Cosmos DB** – document (JSON), key-value, SQL
- **Couchbase** – document (JSON), key-value, N1QL
- **Oracle Database** – relational, document (JSON and XML), graph triplestore, property graph, key-value, objects
- **Redis** – key-value, document (JSON), property graph, streaming, time-series

Microsoft Azure Cosmos DB

- Базата данни е част от Azure;
- Azure Cosmos DB представлява много моделна база данни;
- Базата данни не разполага с определена схема;
- Azure Cosmos DB разполага с автоматичен механизъм за индексиране на информацията;
- Базата данни представлява backend as a service;

Azure Cosmos DB – обединява следните NoSQL и SQL APIs

- SQL APIs
- Cassandra APIs
- MongoDB APIs
- Gremlin APIs
- Azure Tables Storage APIs
- Etcd APIs

DEMO AZURE COSMOS DB

examples/AzureCosmosDB.DatabaseManagement

Google Firebase и Firebase database

- Firebase е мобилна и уеб платформа за разработка на приложения собственост на Google;
- Firebase database е NoSQL база данни от тип key – value;
- Представлява база данни като услуга, част от Google cloud;
- Предоставят се два типа бази данни realtime database and backend as a service;

C# sdk

- FireSharp
  - *Install-Package FireSharp*
- FirebaseSharp
  - *Install-Package FirebaseSharp*
- FirebaseDatabase.net
  - *Install-Package FirebaseDatabase.net*

DEMO GOOGLE FIREBASE DATABASE

examples/DinosaurCatalog

DEMO GOOGLE FIREBASE DATABASE

examples/ConsoleChat

ВЪПРОСИ ?