Software Requirement Specification

CosmosDB RU Cost Calculation Improvement - Efficiency & Time

Version 2.0

January 19, 2019

Team Members: Joe Do, Ferdinand Tembo, Kevin Tran

https://docs.google.com/document/d/15cA7mmaJDEVWgycGrXLp_tpZaY8iRy20Ef

C5z1v4fEw/editBellevue College - Computer Science Department

Alfred Nehme

Revisions Page

Version	Primary Author	Description of Version	Date Completed
1.0	Joe Do Ferdinand Tembo Kevin Tran	First Draft - Interpretation	10/17/2018
2.0	Joe Do Ferdinand Tembo Kevin Tran	Second Draft	01/19/2019

Table of Contents

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to provide an overview of the entire RU Calculator that we are going to build with the purpose, scope, definitions, acronyms, abbreviations, references, overview, overall description , and the architecture of the system. It will explain the features of the system, what we are going to do or not, its constraints about how the system should work . This document can be used by both parties the developers and StockHolders.

1.2 Scope

Cosmos DB is a microsoft’s cloud that need improvement in terms of RU estimation. To estimate the RU , microsoft has it own calculator called ”capacityplanner”. The customer is asked to provide a JSON document and the number of CRUD operations. The scope is to calculate the RU based on these inputs.

1.3 Definitions and Acronyms

CRUD	Create, read, update, delete
Request Unit	Azure Cosmos DB resources are billed based on the provisioned throughput and storage. Azure Cosmos DB throughput is expressed in

	terms of **Request Units**
Store Procedure	It's a set of Structured Query Language (SQL) statements with an assigned name,which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs
User Defined Function (UDF)	It's a side effect free piece of application logic written in JavaScript. It allows developers to construct a query operator, thus extending the core of the Cosmos DB query language
Cosmos DB	Cosmos DB is Microsoft's proprietary globally-distributed, multi-model database service "for managing data at planet-scale"
JSON	(JavaScript Object Notation) is a lightweight data-interchange format
Indexing Policies	A structural rule in a database that strategizes in optimizing database commands and the database resources (CPU usage, I/O, ram) that they use.

1.4 References

[1] Markjbrown. "How to Call Stored Procedures, Triggers, and User-Defined Functions Using Azure Cosmos DB SDKs." *Microsoft Docs*, docs.microsoft.com/en-us/azure/cosmos-db/how-to-use-stored-procedures-triggers-udfs.

[2] Howell, Jason W, et al. "Indexing in Azure Cosmos DB." Microsoft Azure, Microsoft, 9 Nov. 2018, docs.microsoft.com/en-us/azure/cosmos-db/index-overview.

[3] CamSoper. “DocumentClient.ExecuteStoredProcedureAsync Method (Microsoft.Azure.Documents.Client) - Azure for .NET Developers.” *Microsoft Docs*, docs.microsoft.com/en-us/dotnet/api/microsoft.azure.documents.client.documentclient.executestoredprocedureasync?view=azure-dotnet#Microsoft_Azure_Documents_Client_DocumentClient_ExecuteStoredProcedureAsync__1_System_Uri_System_Object___.

1.5 Overview

The remainder of this document include three other chapters.The chapter two provides an overview of the functionality of the RU Calculator. It also mentions the system constraints. The chapter three provides the architecture of system.It also provides the description of the different system interfaces.

2.OVERALL DESCRIPTION

2.1 Product Perspective

The product will use JSON files as input sources , and n number of documents as well as CREATE / READ/ UPDATE/ DELETE operations to estimate the required RU needed for the customer. The product can estimate the RU of any query operations and manipulations that the customer will do with the JSON documents.

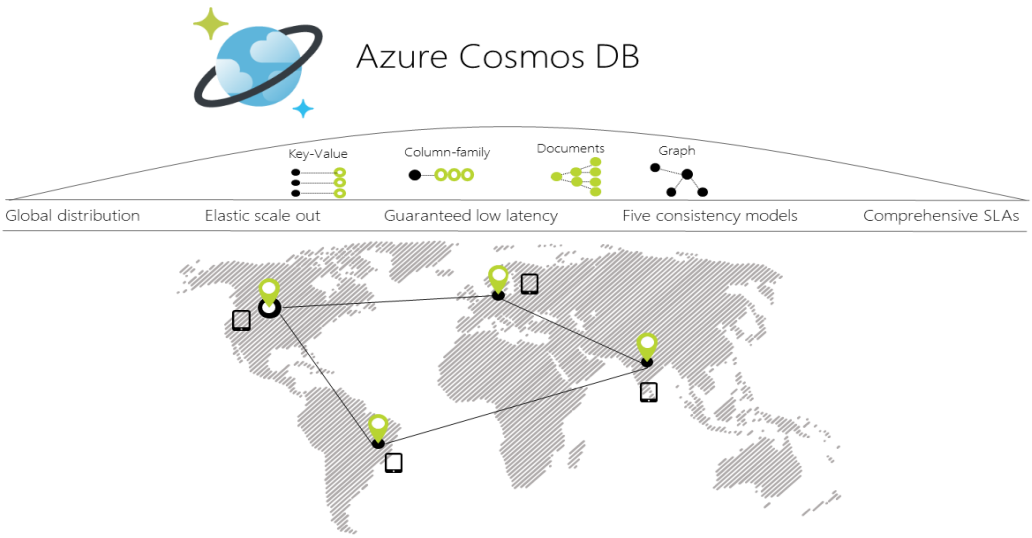
2.2 Product Functions

IResourceResponseBase interface represents the non-resource specific service response headers returned by any request in the Azure Cosmos DB service.

2.3 User Characteristics

The RU calculator tagerts everyone who users who want to build applications with elastic scale and highly responsive with the consistency of data. Also user who want to build planet-scale

applications with Azure Cosmos DB , and also the product also targets users who are thinking about switching from other cloud services like AWS and Google Cloud to Microsoft Azure



2.4 Constraints

- Browser constraints:
- Safari for Mac
- Google Chrome for Mac and Windows
- Firefox for Mac and Windows
- Microsoft Edge
- Internet Explorer

This product will interact with Azure Cosmos DB server, a virtual dedicated server hosted by Microsoft. The server operates on a certain number of RAM, CPU and allocated storage space.

2.5 Assumptions and Dependencies

- The user must upload a JSON document and the number of queries best represents the application usage.

- To Update , the user must have a updated JSON document file to use
- The last assumption about the product is that it will always be accessed with high performance internet connection
- The user must have at least one document , so that the calculator can calculate and give the RU result.

3. SYSTEM ARCHITECTURE

3.1 External Interfaces

Project will use the following external interfaces:

- .NET 4.7.2
- Azure - Cosmos DB SDK 2.1.2
- Azure Cosmos DB NuGet library

3.2 Functional Requirements

The product must be able to output an estimate of required RU's needed to run various queries. It must be able to match or beat the microsoft current RU calculator.

3.3 Performance Requirements

- Output an answer within four (4) seconds.
- Answer must be at least accurate within 5% of the web calculator

3.4 Logical Database Requirements

The client console program and engine must be able to connect to the cloud to create and manage the database in order for queries and data to add onto the database. The database must be hosted by Microsoft in order for this project to exist.

3.5 Design Constraints

- UDF : This has to be defined as a string

- Stored Procedure : This cannot be duplicated by its ID , also can only pass in one parameter on the server-side.

3.6 Software System Attributes

- Customizability: Engine’s ability to intake multiple and varying query conditions
- (Future) Accuracy: Of RU’s required to produced queries
- (Future) Adaptability: To be able to achieve all of the above in any situation while inflation/deflation of the RU may occur as well