

4 Solução Proposta

4.1 Escopo Proposto

Neste trabalho propomos um arcabouço genérico que permite implementar leques de serviços baseados na Simulação de Monte Carlo, utilizando o ambiente de computação na nuvem para prover resultados com tempo e custo ótimos.

Em particular, o arcabouço se concentra no atendimento da gama de domínios com as seguintes características:

- Picos de processamento (curta duração, sazonal ou *on/off*), i.e., existem momentos onde a demanda por processamento é muito maior que a demanda média, a ponto de o custo de dimensionamento da infraestrutura pelo pico tornar a solução economicamente inviável;
- Grande demanda de processamento, i.e., lidam com solicitações onde a quantidade de informações a serem processadas (lidas, calculadas, criadas, etc.) para se obter o resultado é tão grande que demanda um tempo de processamento muito grande, ou até mesmo inviável, em uma abordagem tradicional;
- Imediatismo na obtenção dos resultados, i.e., ocorre situações extraordinárias e pontuais, onde a infraestrutura precisará escalar em muitas vezes para reduzir ao máximo o tempo de processamento necessário para obter-se o resultado desejado;
- Processamento intensivo (Simulação de Monte Carlo) desacoplado, i.e., o processamento intenso demandado por uma aplicação pode ser separada e encapsulada na nuvem, para ser ofertada como serviço a outras aplicações.

Preferencialmente o processamento deve apresentar pouco volume de entrada e saída de dados e volumes significativos de resultados parciais e cálculos, de modo que o tempo de transferência de dados entre o serviço e a aplicação seja o mínimo, a ponto de ser irrelevante frente ao tempo total de execução.

De posse deste arcabouço, e do presente documento, esperamos que um desenvolvedor com conhecimentos básicos possa implementar serviços de calcular (aproximar/simular utilizando o Método de Monte Carlo) para atendimento de domínios específicos, com otimização de custo e tempo, podendo,

ainda, acoplá-los as suas aplicações, sem precisar adquirir nenhum conhecimento profundo de computação na nuvem ou migrar suas soluções.

Dessa forma, este arcabouço se propõe a tornar o uso de ambientes de computação na nuvem uma realidade para usuários de Simulações de Monte Carlo, ampliando seu campo de aplicação.

O arcabouço proposto foi batizado de “McCloud Service Framework”, ou simplesmente “McCloud” (“Monte Carlo Cloud”). A identidade visual criada para o arcabouço é apresentada na Figura 5.



Figura 5 – Identidade visual do McCloud Service Framework

4.2 Arquitetura Conceitual

Para definição da arquitetura conceitual do McCloud optamos pela adoção da ontologia para computação na nuvem proposta por Youseff, Butrico e Silva [11], que estabelece cinco camadas: aplicativos, ambientes de software, infraestrutura de software, kernel (núcleo) de software e hardware. Reproduzimos a ontologia proposta pelos autores na Figura 6.

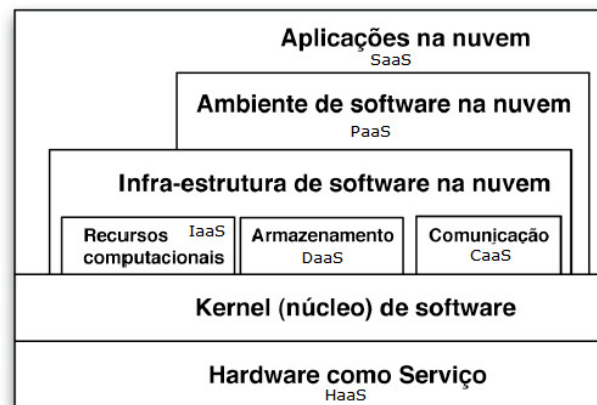


Figura 6 – Ontologia para computação na nuvem

As camadas desta ontologia, geralmente referenciadas pelos seus acrônimos em inglês, abstraem as camadas inferiores dos seus usuários, conforme apresentado a seguir.

- **SaaS (*Software as a Service*):** oferece como serviço softwares que executam em um ambiente de computação na nuvem;

- **PaaS (*Platform as a Service*):** oferece serviços específicos, tais como: banco de dados, troca de mensagem, armazenamento, processamento, dentre outros;

- **IaaS (*Infrastructure as a Service*):** oferece como serviço infraestrutura, normalmente servidores ou parte deles, para desenvolver, testar, implantar e executar aplicações próprias;

- **DaaS (*Data as a Service*):** refere-se exclusivamente ao serviço de dados;

- **CaaS (*Communication as a Service*):** refere-se exclusivamente ao serviço de comunicação;

- **Kernel:** refere-se ao *kernel* do sistema operacional e a máquinas virtuais;

- **HaaS (*Hardware as a Service*):** refere-se a oferta como serviços de equipamentos propriamente dito.

Na Tabela 2 [5], a seguir, relacionamos um conjunto de exemplos de ambientes de computação na nuvem comerciais classificados nesta ontologia.

Camada	Computação Tradicional
SaaS	Google Apps, Salesforce CRM, Facebook, Exchange Online, Microsoft SharePoint, Google Docs, SAP Business ByDesign
PaaS	Plataformas de Computação na Nuvem fornecidas pela Amazon, Microsoft (Azure), Google, Salesforce, dentre outras
IaaS	EC2 da Amazon, Enomalism Elastic Cloud e Microsoft (Azure)
DaaS	S3 da Amazon, EMC Storage Managed Service, Microsoft SQL Azure
CaaS	Microsoft Connected Service Framework (CSF)
Kernel	Globus e Condor
HaaS	Acordo de <i>lease</i> de hardware entre a IBM e o Morgan-Stanley, projeto Kittyhawk da IBM

Tabela 2 – Exemplos de ambientes de computação na nuvem [5]

Na literatura encontramos com maior frequência as camadas inferiores (DaaS, CaaS, Kernel e HaaS) sendo representadas unicamente pela camada IaaS, reduzindo, assim, o modelo para apenas 3 camadas. Preferimos, no entanto, apresentar uma ontologia com mais detalhes para melhor compreensão da arquitetura do arcabouço.

Diante da ontologia acima apresentada e da proposta do McCloud podemos perceber que as implantações realizadas com este arcabouço e executadas em

ambiente de computação na nuvem constituem-se em software como serviço (SaaS).

O arcabouço proposto, como definido, proverá um leque de serviços baseados na Simulação de Monte Carlo e será consumido por aplicações clientes. Optamos por adicionar uma camada referente a essas aplicações na arquitetura da solução.

Por outro lado, podemos considerar que as outras camadas dessa ontologia serão abstraídas pela camada PaaS, à medida que optaremos por um serviço de plataforma de computação na nuvem para suportar a implantação.

Dessa forma, a arquitetura conceitual da solução proposta se apresenta conforme a Figura 7 apresentada a seguir.



Figura 7 – Arquitetura conceitual da solução

4.3 Plataforma

Diante da arquitetura conceitual apresentada na seção anterior, claro se torna o papel de uma plataforma de computação na nuvem para suportar a solução proposta.

O objetivo de proporcionar à Simulação de Monte Carlo novas perspectivas através da elasticidade de recursos e do modelo de pagamento evidencia o fato de ser necessário escolhermos um fornecedor de nuvem pública. Como maiores fornecedores do mercado podemos citar Microsoft Windows Azure [12], Amazon Web Services [39] e Google App Engine [40].

Um dos objetivos deste trabalho é torná-lo acessível para desenvolvedores e pesquisadores de diferentes áreas, razão pela qual optamos pelo Microsoft

Windows Azure [12][13][14][15], conforme ilustrado anteriormente. Os principais motivos que conduziram a essa escolha são resumidos a seguir.

1) Abstração do sistema operacional, pois, a priori, não há necessidade de instalar nada nos servidores, o que não ocorre em outros fornecedores;

2) A IDE (*Integrated Development Environment*) exigida para desenvolvimento de aplicativos é grátis (*Microsoft Visual Studio 2010 Express*) e fornece uma série de facilidades para o desenvolvedor (autocompletar, documentação dos métodos, etc.);

3) Fácil instalação dos requisitos e kits para desenvolvimento e testes em máquina local, reduzindo o tempo e custo de desenvolvimento;

4) Existe disponível uma grande variedade de artigos, exemplos, livros e materiais de formação básicos sobre a plataforma;

5) A escolha da linguagem de programação C#.Net aproveita o conhecimento da sintaxe do C/C++, que muitos desenvolvedores possuem [13][14][30][31];

6) Interoperabilidade com outras tecnologias e linguagens de programação, através do WCF (*Windows Communication Foundation*).

O Windows Azure é a proposta da Microsoft para serviços de computação na nuvem. O serviço consiste em uma plataforma para execução de aplicações (PaaS), onde o programador encontra um conjunto de recursos que facilitam o desenvolvimento de aplicativos de software.

A plataforma é composta, essencialmente, por três grandes componentes, formadores do núcleo do serviço, quais sejam: unidades de computação (*compute*); espaço para armazenamento (*storage*); e fábrica (*fabric*).

As unidades de computação, também chamadas de nós ou instâncias, podem ser máquinas virtuais ou físicas e são caracterizadas pelo seguinte conjunto de atributos: velocidade do processador, quantidade de memória e tamanho do disco local. Nesse trabalho será sempre considerada a instância apelidada no Azure de “pequena”.

O espaço para armazenamento é composto de estruturas de dados criadas a partir de abstrações já conhecidas, e.g., tabelas, arquivos e filas.

A fábrica, por sua vez, é responsável pelo gerenciamento e monitoramento do *datacenter* (servidores, IPs, balanceadores de carga, switches, roteadores,

dentre outros) e, também, pelo gerenciamento do ciclo de vida da aplicação e principalmente pela garantia dos seus níveis de serviço.

Cada nó pode exercer um papel (role), ou seja, uma função. Existem quatro tipos básicos de papéis:

- **Web Role:** é utilizado como interface pública da aplicação. Trata-se de um *website* tradicional, desenvolvido com a tecnologia ASP.Net;

- **FastCGI Web Role:** é semelhante ao anterior, diferenciando-se pela tecnologia utilizada, pois adiciona mecanismos (*handlers*) ao servidor web permitindo a execução de outras linguagens, como PHP, Python, Ruby, dentre outras.

- **Worker Role:** é utilizado para rodar processos em segundo plano e não possui uma interface pública;

- **WCF Worker:** é semelhante ao anterior, mas possui interface pública baseada no *Windows Communication Foundation*, modelo unificado de programação da Microsoft para construção de aplicações distribuídas orientadas a serviço (SOA).

Cada papel pode ser executado em vários nós garantindo, assim, a escalabilidade das aplicações. Todos os papéis tem acesso à área de armazenamento de dados, portanto, a intercomunicação entre os mesmos pode ser feita através dessas estruturas, que são explicadas a seguir.

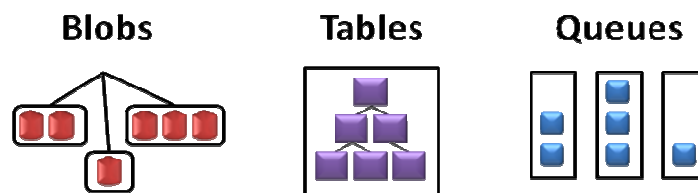


Figura 8 – Estruturas de armazenamento para tabelas no Azure

O espaço para armazenamento de dados (*storage*) possui três estruturas, como citado anteriormente, são elas: **Blobs** (arquivos), **Tables** (tabelas) e **Queues** (filas). Todos os dados criados no sistema de armazenamento são replicados automaticamente para fornecer alta disponibilidade e confiabilidade. A utilização destas estruturas se dá através de interfaces (APIs) fornecidas pelo serviço, que são chamadas pelo endereço, passando-se como parâmetros uma identificação única da conta e uma chave de acesso.

O **Blob** (*Binary Large Object*) é uma representação genérica para arquivos. Estes arquivos devem ser organizados em containers, ou seja, divisões lógicas semelhantes aos diretórios, onde se podem configurar permissões de acesso. Cada blob pode ter um tamanho máximo de 50GB e a carga de dados (*upload*) é feita em blocos, permitindo o envio paralelo.

A **Table** é uma abstração para tabela, com algumas diferenças para as tradicionais tabelas dos bancos de dados relacionais. A principal diferença é a falta de estrutura fixa, ou seja, cada linha da tabela, ou *Entity*, como foram nomeadas no Azure, pode conter um conjunto de propriedades (*Properties*) diferente. Além disso, não existem, por exemplo, chaves estrangeiras entre tabelas; junção entre tabelas no lado do servidor; criação de índices customizados; realização de contagem de linhas no lado do servidor.

Duas propriedades obrigatórias são utilizadas como identificador único dentro de uma tabela: *PartitionKey* e *RowKey*. A primeira possibilita escalabilidade, já que as tabelas podem ser divididas em partições e armazenadas em nós diferentes. Já a segunda funciona como um identificador dentro de uma partição. Além disso, possuem obrigatoriamente a propriedade *Timestamp*, com a data e hora da última modificação da entidade. Na Figura 9 apresentamos uma ilustração da estrutura básica de tabela do Azure.

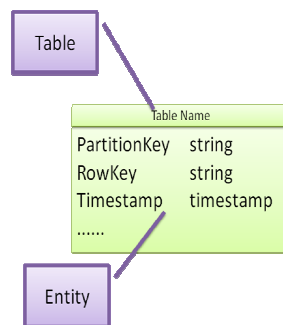


Figura 9 – Estrutura padrão da Table do Azure

Por fim, a **Queue** (fila) é um conjunto de mensagens ordenadas (*FIFO - First in First out*), com tamanho máximo de 8KB, seus nomes precisam ser em letra minúscula e sem caracteres especiais (*URL friendly*) e quantidade ilimitada. Tem como objetivo principal a comunicação entre nós, como, por exemplo, para a distribuição de trabalho assíncrono, conforme debateremos mais a frente. O funcionamento desta tecnologia é ilustrada na Figura 10.

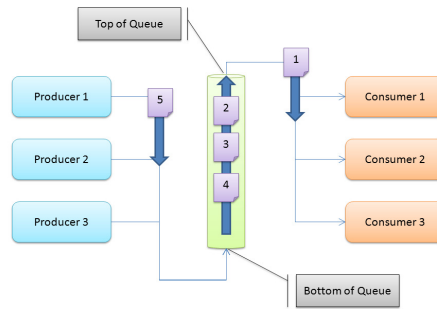


Figura 10 – Funcionamento da tecnologia de fila do Azure

O site do fornecedor define: *“O Microsoft Windows Azure é uma plataforma de computação na nuvem aberta e flexível que permite o rápido desenvolvimento, implantação e gestão de aplicativos globalmente conectados aos datacenters da empresa ao redor do mundo. Com ele é possível desenvolver aplicações em qualquer linguagem, ferramenta ou arcabouço.”*

Apesar dos conceitos discutidos neste trabalho serem independentes de plataforma o Windows Azure é o alicerce do McCloud.