

## 11 Conclusão

### 11.1 Descobertas

Nesse trabalho apresentamos o McCloud Service Framework, um arcabouço para implementação de serviços baseados na Simulação de Monte Carlo na nuvem, disponibilizamos duas implantações genéricas para uso científico utilizando algoritmos de simulação nas linguagens C#.Net e MATLAB. Realizamos experimentos expressivos com as implementações propostas, e.g., 100 instâncias simultâneas; amostra com um trilhão de realizações; 300 vezes mais rápido que na computação tradicional; divisão do processamento em 1.024 partes.

O McCloud permite aos pesquisadores incrementarem suas experiências utilizando ambientes de computação na nuvem, com mínimas alterações nos seus algoritmos de simulação, mantendo a interoperabilidade com suas aplicações originais, independente da tecnologia em que foram desenvolvidas.

Ao fazer uso da nuvem o arcabouço revoluciona a aplicação do Método de Monte Carlo, possibilitando simulações com um número muito maior de realizações, o que resulta em aproximações mais precisas. Além disso, torna o custo de simulação proporcional aos seus requisitos de execução, sem ociosidades ou escassez de recursos computacionais, viabilizando diversos experimentos, antes estacionados pela necessidade de grandes investimentos em infraestrutura, ou morosidade excessiva das simulações.

Esse trabalho representa também um alicerce de outras evoluções a serem propostas fazendo uso do advento da computação na nuvem e seus benefícios, pois propõe uma arquitetura de referência, sistemáticas e ferramentas que podem ser adaptadas para outros domínios diferentes que requerem a utilização de Monte Carlo.

Por último, nossa expectativa é que o McCloud efetivamente colabore no cenário nacional com diversas pesquisas, que fazem uso da Simulação de Monte Carlo para obter aproximações.

## 11.2 Lições Aprendidas

Diante das dificuldades enfrentadas na realização deste trabalho, tornou-se evidente a necessidade de dominar alguns recursos do Azure antes do desenvolvimento da solução de um problema específico, i.g., manipulação (inserção, alteração, remoção e consulta) dos armazenamentos (fila, blob e tabela); configuração de certificados e acesso remoto às instâncias (*RDP – Remote Desktop Protocol*); implantação de serviços (*WCF - Windows Communication Foundation*); tarefas de inicialização (*startup task*), instruções em linha de comando e orquestração de *downloads* e instalações de arquivos (*Bootstrapper*); técnicas de instrumentação; configuração e uso do armazenamento local das instâncias.

Algumas observações são de extrema relevância para próximos projetos:

1) Observar o tempo máximo de bloqueio de uma mensagem na fila (*timeout*). Isto porque, quando o tempo se expira, sem que a mensagem tenha finalizado, esta volta para o início da fila, causando seu duplo processamento e, conseqüentemente, falha no paralelismo.

2) Não é permitido salvar arquivos na instância, a não ser na pasta local configurada no arquivo de definições, ou seja, ao tentar salvar em outro diretório o programa trava.

3) As instâncias, a priori, não possuem usuários. Logo, as instalações devem considerar essa limitação ou criar seus usuários. Erros desse gênero aparecem nos Eventos do Windows de cada instância. Veja o abordado em nosso Estudo de Caso.

4) Como no Azure estamos sujeitos a leis Americanas, recomenda-se que a simulação seja “cega”, ou seja, rodem apenas massas de números, sem identificação dos mesmos, pelo fato de o governo americano poder acessar tais informações.

Por fim, destacamos algumas curiosidades:

1) Erros de programação que impedem as instâncias de subir, normalmente, fazem com que esse processo entre em *loop*. É essencial observar o tempo médio deste processo. Sem instalações adicionais, o mesmo levou em torno de 7 minutos.

2) Aparentemente o processo de subir um instância é muito mais lento do que em concorrentes, chegando a média de 7 minutos em casos simples, contra 30 segundos dos concorrentes com sistema operacional Linux.

3) Para executar com mais de 20 instâncias é preciso solicitar com antecedência a liberação deste limite na sua conta.

### 11.3 Limitações

A solução proposta e o trabalho realizado apresentam as seguintes limitações:

- A versão atual do arcabouço, apesar de calcular o número ótimo de instâncias para processamento, e o tempo máximo de espera de uma tarefa no ponto de extensão *optimization*, não altera essas configurações automaticamente. Esse ponto precisa ser aperfeiçoado para sua maior independência operacional.
- Esse trabalho não contemplou um viés de tolerância à falha, extremamente necessário se desejarmos tornar este arcabouço um produto. Por exemplo, o travamento de uma instância não possui qualquer ação corretiva, assim como, a não finalização de uma simulação, na hipótese de problemas na troca de mensagem, não possui uma ação preventiva.
- Não existe uma verificação se os requisitos necessários estão instalados, por exemplo, as dependências para a execução do MATLAB analisadas no estudo de caso. Portanto, qualquer automatização no levantamento de novas instâncias depende desta verificação, para evitar que ocorram problemas no processamento.
- Ao concatenar os arquivos gerados na etapa *process* podemos gerar um arquivo extremamente volumoso. Logo, a implementação do ponto de extensão *finish* deve considerar técnicas de leitura em pedaços deste arquivo, caso contrário o carregamento destes dados em memória pode tornar a etapa demasiadamente lenta ou mesmo exceder os limites de memória da instância. O mesmo problema pode ocorrer com os tipos escolhidos para contadores, somadores e outras variáveis acumulativas, pois o tamanho destes pode exceder o permitido aos mesmos. Também

é necessário de atenção ao limite de dados locais em uma instância do Azure (~200GB). Portanto, é preciso de atenção na implementação, pois as situações citadas são imperceptíveis em poucas realizações.

- O paralelismo se torna irrelevante se o gerador de números pseudo aleatório, implementado no ponto de extensão *execute*, não garantir uma semente suficientemente espaçada entre os nós e um gerador com periodicidade adequada dado o número de realizações. No estudo de caso esse ponto foi tratado com cuidado, utilizando um gerador com boa periodicidade e controle da semente, mas na prova de conceito, para um número muito grande de realizações, o gerador não atende a esses requisitos, e pode ocorrer repetição de realizações, dado que a semente não é controlada.
- A otimização proposta precisa considerar que é possível realizar simulações concorrentes. Além disso, apesar de providências teóricas terem sido realizadas para permitir simulações concorrentes, não foram feitos testes com esse enfoque.

## 11.4 Trabalhos Futuros

Este trabalho apresenta uma arquitetura de referência para a implantação de métodos de Monte Carlo na nuvem. A partir dele muitos outros trabalhos e evoluções são possíveis. Discutimos algumas das possibilidades no que se segue.

- **Pensando em Paralelo: Estudo de Caso com MATLAB**

Um algoritmo de simulação para ser executada em paralelo precisa ser adaptado de forma que a união dos resultados parciais permita alcançar o resultado final estatístico correto. Nesse contexto, reunimos para viabilização do estudo de caso apresentado neste trabalho uma série de técnicas, que certamente são de interesse de pesquisadores da área de Matemática e Engenharias em geral, principalmente, ao serem exemplificados em uma das principais ferramentas do mercado, o MATLAB. Propõe-se estudar e compilar este conjunto de técnicas.

- **Um novo olhar estatístico com a Computação na Nuvem**

Ao alcançarmos um volume maior de realizações, em alguns problemas, é possível perceber regiões anteriormente “ocultas”. O uso

da computação na nuvem viabiliza enxergar o que antes era imperceptível à maioria dos orçamentos de pesquisa que adotam o Método de Monte Carlo. Estudar e exemplificar casos concretos deste novo olhar certamente quebrará paradigmas e motivará novas pesquisas de interesse para as áreas de Matemática, Estatísticas, Finanças e Engenharias em geral.

- **Fronteiras de Eficiência das Simulações na Nuvem**

Nossa abordagem sobre as fronteiras de eficiência na simulação na nuvem foi extremamente genérica e se limitou a um domínio simples e linear. Além disso, ocorreu de forma manual. Automatizar essa elasticidade e modelar casos mais complexos permitirá discutir com maior profundidade os ganhos com o paralelismo, seus limites, e custos. Esse estudo é do interesse de todas as soluções que utilizam o ambiente de computação na nuvem.

- **Estudo de Caso em R com McCloud**

A linguagem R [32] é atualmente uma das principais ferramentas gratuitas utilizadas no meio acadêmico para computação estatística e é totalmente gratuito. A implementação de um estudo de caso do McCloud em R pode motivar a adoção da ferramenta em diversas pesquisas, incrementando seus resultados e reafirmando seus benefícios.

- **Principais Estratégias de Tolerância a Falhas da Nuvem**

Um ambiente paralelo demanda uma série de cuidados, pois uma pequena falha pode comprometer o resultado de toda a solução. Implementar estratégias nesse ambiente, que é um assunto novo, demanda estudos adicionais.

- **Estudo de Caso de Aplicação Cliente do McCloud**

A escolha de uma aplicação existente que faça uso do Método de Monte Carlo com computação tradicional de forma a adaptá-la para o consumo deste serviço com o McCloud é uma experiência rica, que pode resultar em evoluções do arcabouço.

- **Kit de Geradores de Números**

Disponibilizar um conjunto de geradores de números pseudo-aleatórios ou mesmo aleatórios, aderentes a um universo amplo de distribuições, encapsulado no arcabouço, permite que o desenvolvedor de serviços faça uso destes recursos de forma rapidamente e sem estudar em profundidade o tema, para aprimorar seu serviço e garantir a aleatoriedade de cada realização.

- **Generalização do Arcabouço**

A arquitetura proposta pelo arcabouço pode ser adaptada para outros contextos de interesse, diferentes do Método de Monte Carlo. Escolher um contexto de relevância científica e mensurar os esforços para adaptação é extremamente interessante para que possamos medir a abrangência e adaptabilidade da solução aqui apresentada.