

5 Modelo de Programação

5.1 Visão Geral

O modelo de programação do McCloud foi inspirado no Map/Reduce, originalmente proposto em 2004 por Jeff Dean e Sanjay Ghemawat [16], para suportar o processamento de grandes coleções de dados de forma paralela e distribuída, utilizando um grande número de computadores (nós), coletivamente referidos como um *cluster*.

Esse modelo envolve dois passos: o mapeamento e a redução. No passo de mapeamento, o nó mestre divide os dados de entrada em subproblemas e os distribui para resolução pelos *mappers*. Basicamente, um *mapper* processa o subproblema e envia a resposta ao nó mestre. Este, ao receber as respostas de todos os subproblemas distribuídos aos *mappers* passa à etapa de redução, onde as respostas são enviadas aos *reducers*, que processam estas combinações de respostas a fim de obter uma nova resposta e as enviam de volta ao nó mestre, que obterá, então, a resposta do problema original.

A Figura 11 ilustra este fluxo de dados na resolução de um problema neste modelo de programação.

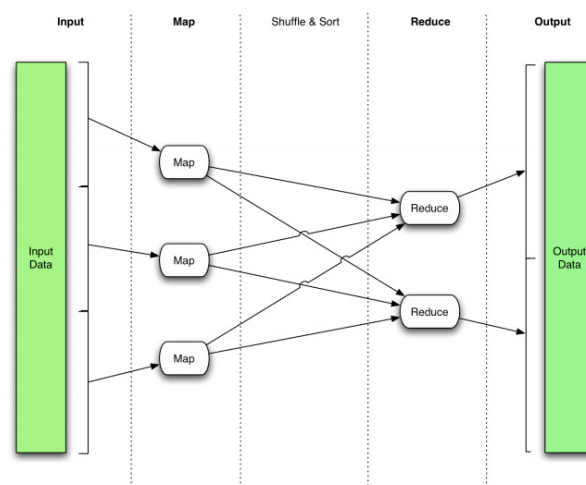


Figura 11 – Fluxo de dados do modelo Map/Reduce [16]

O modelo proposto, visando a melhor utilização das características do ambiente de computação na nuvem, é realizado em três etapas, conforme ilustrado na Figura 12, inspirada em Pereira et. at. [17].

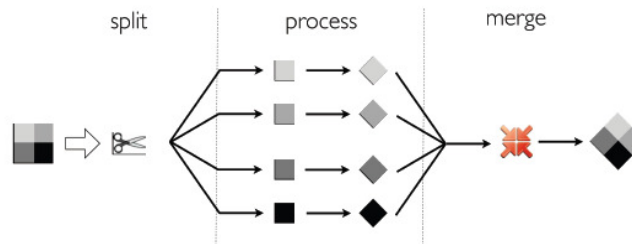


Figura 12 – Simplificação do modelo de programação do McCloud [17]

Em resumo, na etapa de *split* a solicitação de simulação enviada pelo cliente, referente à n realizações (amostra), é dividida em tarefas contendo um subconjunto destas realizações. Na etapa de *process* cada nó que estiver ocioso executa uma destas tarefas e salva o resultado da mesma, assim, as tarefas são executadas em paralelo pelos nós disponíveis. O nó que termina a execução da última tarefa que faltava da simulação inicia a etapa de *merge* desta, onde o conjunto de resultados das tarefas desta simulação são concatenados e processados para obtenção do resultado final da simulação, que é salvo e fica disponível para o cliente.

As principais características de cada etapa e seu fluxo são apresentadas a seguir e, após aprofundarmos o entendimento do arcabouço proposto apresentamos, na seção 6.6, um maior detalhamento da ilustração acima.

5.2 Etapa Split

Características

- Desempenhada pelo papel **WCF Role** e pode ser exercido por um único nó, no entanto, visando maior garantia de disponibilidade, é possível executar em vários nós;
- Ponto de entrada do leque de serviços, provendo o serviço de iniciar (executar) uma simulação implementada (o método em questão, *Run*, será tratado mais adiante);
- Recebe os pedidos de simulação e os subdivide em tarefas, ou seja, realiza a divisão das n repetições dos passos B e C do Método de Monte Carlo, descritos na seção 3.3, em conjuntos de repetições, visando paralelizar a execução destas tarefas e, conseqüentemente, da simulação;
- Coloca as tarefas criadas na fila para serem executadas na etapa *process*;

- Redimensiona os recursos computacionais e define a melhor estratégia de divisão para que a simulação ocorra na melhor relação de tempo e custo na visão dos requisitos do negócio;
- Armazena as informações da simulação para futuro controle do seu andamento;
- O nó que desempenha esse papel também disponibiliza outros métodos ao cliente, conforme será explicado na seção 6.3.

Fluxo da Etapa *Split*

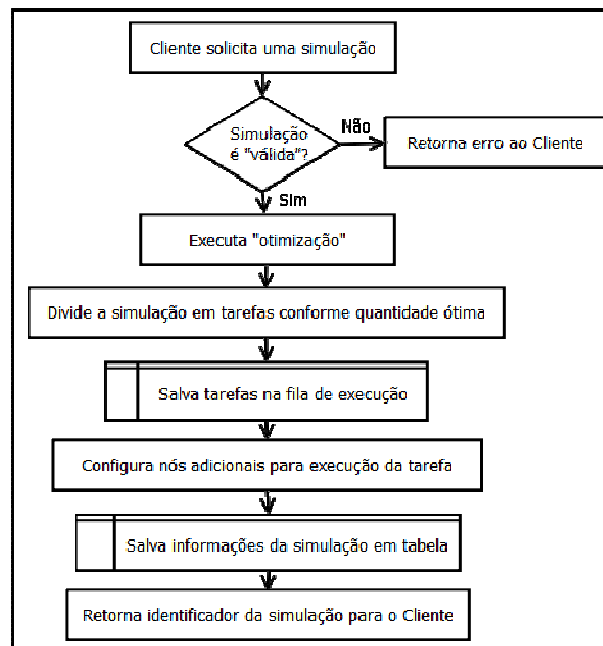


Figura 13 – Fluxo da etapa split

5.3 Etapa Process

Características

- Desempenhada pelo papel **Woker Role** e deve ser exercida pela quantidade de nós disponíveis para execução desta simulação;
- Monitora a fila de execução em busca de tarefas para executar;
- Captura a primeira tarefa disponível, quando existente, e a bloqueia;
- Executa a tarefa capturada, ou seja, executa uma parte da simulação;
- Salva o resultado da tarefa executada em um blob no container referente à simulação em questão, removendo a tarefa bloqueada da fila;

- Verifica se todas as tarefas foram executadas e, caso positivo, inicia etapa de *merge* para a simulação em questão, registrando na tabela a conclusão da etapa.

Fluxo da Etapa *Process*

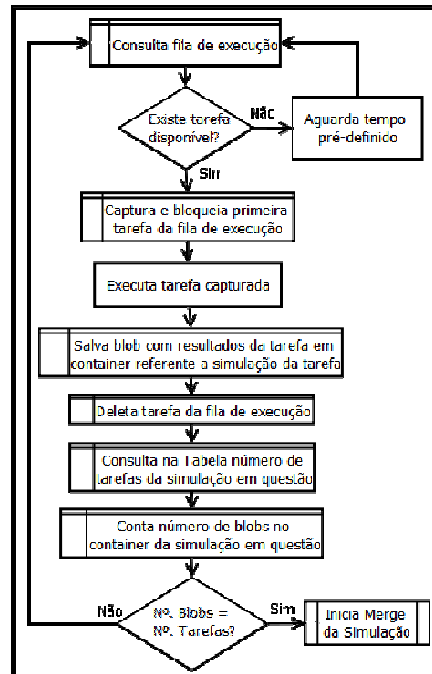


Figura 14 – Fluxo da etapa process

5.4 Etapa Merge

Características

- Executada dentro do nó que realizou a última tarefa;
- Recupera os resultados de todas as tarefas (salvas em blob na etapa *process*);
- Concatena as tarefas criando um arquivo local único na instância;
- Executa o processamento deste arquivo e gera o resultado final da simulação;
- Salva o resultado final da simulação em blob;
- Registra mudança de status da simulação para finalizada.

Fluxo da Etapa *Merge*

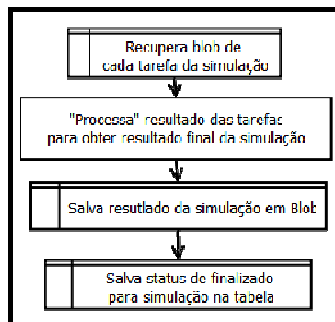


Figura 15 – Fluxo de trabalho da etapa *merge*