



Otimizando o Hadoop MapReduce para Tarefas Pequenas: um estudo envolvendo simulações de cenários agrícolas

Alan Massaru Nakai¹

Introdução

O Hadoop (BORTHAKUR, 2007; LAM, 2011) é um arcabouço de software livre, mantido pela Apache, voltado para processamento distribuído de grandes quantidades de dados. O Hadoop permite escalas de uma única máquina a milhares delas. Além disso, implementa tolerância a falhas no nível da aplicação, provendo serviços altamente disponíveis. Dentre os diversos componentes do arcabouço, destaca-se a sua implementação do modelo de programação MapReduce (DEAN; GHEMAWAT, 2008), originalmente proposto pelo Google, para processamento e geração de grandes quantidades de dados.

No Laboratório de Modelagem Agroambiental (LMA) da Embrapa Informática Agropecuária, o Hadoop MapReduce é utilizado na simulação de cenários agrícolas futuros. Estas simulações são realizadas por meio de uma metodologia de zoneamento de risco climático, utilizando dados de projeções climáticas provenientes do Intergovernmental Panel on Climate Change (IPCC).

A metodologia de zoneamento utilizada é baseada no cálculo do balanço hídrico, que requer parâmetros como: coeficiente das culturas, capacidade de armazenamento de água do solo, evapotranspiração e séries

temporais de chuva. Utiliza-se um modelo baseado em balanço hídrico chamado Bipzon (ASSAD, 1983; FOSTER, 1984; VAKSMANN, 2000) para calcular o Índice de Satisfação da Necessidade de Água (Isna) para cada coordenada desejada. Este índice é utilizado pelos especialistas para determinar os cenários agrícolas com base no risco climático.

Atualmente, os cenários agrícolas futuros são simulados com base em dados históricos de 3313 estações climáticas. Estas séries históricas são perturbadas com as variações de temperatura e chuva simuladas nos modelos de projeção climática. Para cada simulação, o valor do Isna deve ser calculado para todas estas estações.

No contexto da computação de alto desempenho, no qual a quantidade de dados a serem processados frequentemente chegam a centenas de *gigabytes* ou *terabytes*, a quantidade de dados processados em cada simulação de cenário agrícola futuro é relativamente pequena. Apesar disto, a execução distribuída destas simulações pode ser justificada pela grande quantidade de cenários que podem ser gerados em um mesmo estudo. Existem dezenas de modelos de projeção climática que podem ser utilizados para perturbar as séries históricas e, eventualmente, novos modelos

¹ Doutor em Ciência da Computação, Analista da Embrapa Informática Agropecuária, Campinas, SP, alan.nakai@embrapa.br

são gerados. Cada modelo possui diversas variações, de acordo com a quantidade de emissão de carbono considerada na projeção. Além disso, os cenários agrícolas futuros podem considerar diferentes períodos dos modelos nas simulações (ex. 2025, 2050, 2075, etc). Combinando-se tudo isso, com as diferentes culturas que podem ser simuladas (atualmente 20), diferentes tipos de solo, além parâmetros arbitrários definidos pelos pesquisadores, o escopo de possibilidades de simulações pode crescer muito, chegando a centenas de milhares.

Um problema de se utilizar o MapReduce na simulação de cenários agrícolas é que, apesar de envolver o cálculo do Isna para milhares de coordenadas geográficas, o cálculo em si é relativamente rápido, se comparado ao *overhead* da distribuição das tarefas entre as diversas máquinas do *cluster*. Isto faz com que o ganho de desempenho ao se distribuir as tarefas via MapReduce seja menos significativo.

Neste contexto, o objetivo deste trabalho é apresentar um estudo sobre a otimização da utilização do MapReduce para tratar tarefas relativamente pequenas, como é o caso do cálculo do Isna. Os resultados apresentados mostram que, a partir da agregação de tarefas, pode-se melhorar consideravelmente o desempenho do MapReduce nestes casos, melhorando significativamente o tempo de simulação de um cenário.

O restante do texto é organizado da seguinte forma: A seção 2 introduz o problema tratado. A seção 3 apresenta a metodologia empregada. A seção 4 apresenta os resultados obtidos e, finalmente, a seção 5 apresenta as conclusões do trabalho.

Descrição do problema

O MapReduce aplica a técnica de dividir para conquistar, no qual o problema é dividido em problemas menores e processados separadamente de forma distribuída, nas máquinas do *cluster*. Neste modelo de programação a computação é realizada em termos de dois tipos de tarefas maps (mapeadores) e *reducers* (redutores). As tarefas de mapeamento são responsáveis por executar a computação sobre frações dos dados de entradas (*splits*) e geram resultados intermediários. Cada *split* é processado por uma instância diferente do mapeador. Os resultados dos mapeadores são consolidados pelos redutores, que agregam os resultados intermediários e geram o resultado final.

O formato dos splits variam conforme a natureza dos dados de entrada. Um *split* pode ser, por exemplo, uma quantidade específica de linhas de um arquivo muito grande, uma quantidade fixa de *bytes* ou um arquivo inteiro, quando os dados de entrada são uma coleção grande.

O software que implementa o MapReduce (ex. Hadoop) cuida do escalonamento das tarefas, ou seja, paraleliza automaticamente a computação dos mapeadores e redutores nas máquinas do *cluster*. Além disso, fornece tolerância a falhas de forma transparente. As tarefas são monitoradas e, no caso de falhas, são reexecutadas. Estas características facilitam o desenvolvimento das aplicações, pois os programadores não precisam se preocupar com aspectos de escalonamento e tolerância a falhas.

A Figura 1 ilustra como o MapReduce é utilizado na simulação de cenário agrícolas. Os dados de entrada são

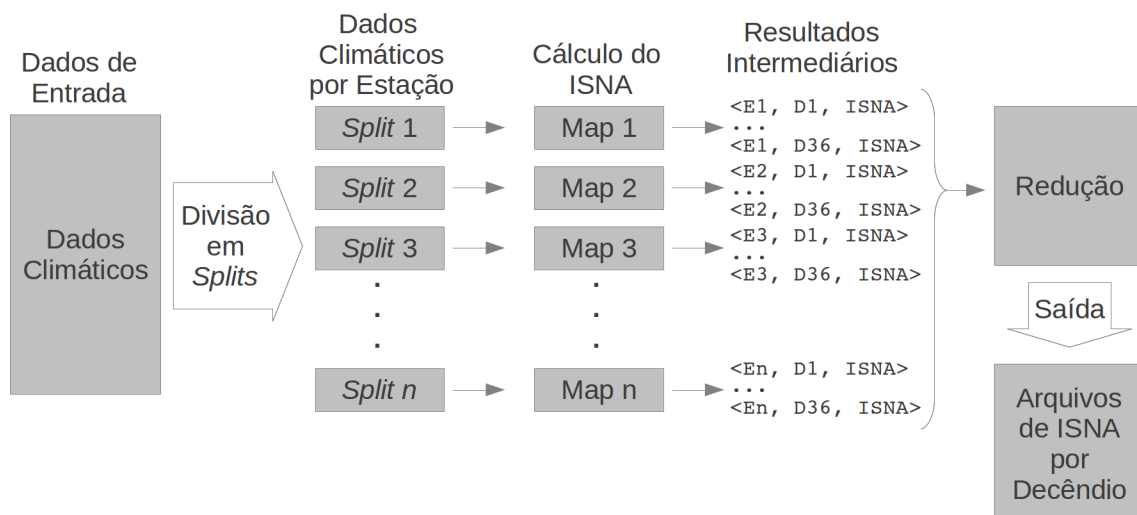


Figura 1. Esquema da simulação de cenários agrícolas utilizando MapReduce; n é o número de estações.

dados climáticos utilizados no modelo Bipzon. Em um primeiro momento, estes dados são divididos em *splits* que correspondem ao dados de uma estação climatológica. O Hadoop cria uma instância de mapeador para cada *split*. O mapeador implementa o modelo Bipzon para calcular o valor do Isna correspondente a uma estação, para todos os decêndios do ano. Todos esses Isnas são agregados pelo redutor, que gera um arquivo por decêndio contendo os Isnas de todas as estações. Posteriormente, esses arquivos serão espacializados para criação dos mapas dos cenários agrícolas.

Um problema de se utilizar MapReduce no cálculo do Isna é que este cálculo é relativamente rápido, em torno de centenas de milissegundos. Isto faz com que o tempo gasto pelo Hadoop com escalonamento e monitoramento das tarefas seja significativo para o tempo total da simulação. Por esta razão, apesar do tempo total de execução utilizando o MapReduce ser melhor do que o tempo que seria gasto em uma execução não distribuída (sequencial), o ganho é afetado negativamente.

Metodologia

A metodologia empregada neste estudo parte da seguinte hipótese: se o tempo gasto com o *overhead* imposto pelo controle das tarefas afeta negativamente o tempo total da execução, a agregação de várias tarefas e a consequente diminuição do número total de tarefas aumentaria o ganho de tempo da execução distribuída.

Para avaliar esta hipótese, desenvolveu-se uma variação do mapeador descrito na Seção 2. Esta nova versão recebe como entrada um arquivo contendo dados climatológicos de k estações climatológicas, onde k é igual ao número de estações dividido pelo número de *splits* que se deseja. O mapeador calcula o valor do Isna para as k estações, para os 36 decêndios. A saída gerada para o redutor é a mesma da versão anterior. A Figura 2 apresenta o esquema da nova versão. Este novo esquema reduz o número de tarefas (instâncias do mapeador) em k vezes.

Para testar a nova implementação também foi necessário desenvolver uma ferramenta para agregar os dados climatológicos das estações. Esta ferramenta recebe com entrada os n arquivos de dados climatológicos e o número de splits desejados. A saída é n/k arquivos com dados de k estações cada um.

Resultados

Para avaliar o impacto da agregação de tarefas no tempo de execução das simulações utilizou-se um *cluster* Hadoop com 16 computadores, configurados com 80 slots de mapeamentos, ou seja, foram permitidos 80 mapeamentos simultâneos (5 por computador).

Os dados utilizados nos experimentos são dados de 3313 estações climatológicas e parâmetros específicos para uma cultura arbitrária. Realizou-se 3 conjuntos de simulações:

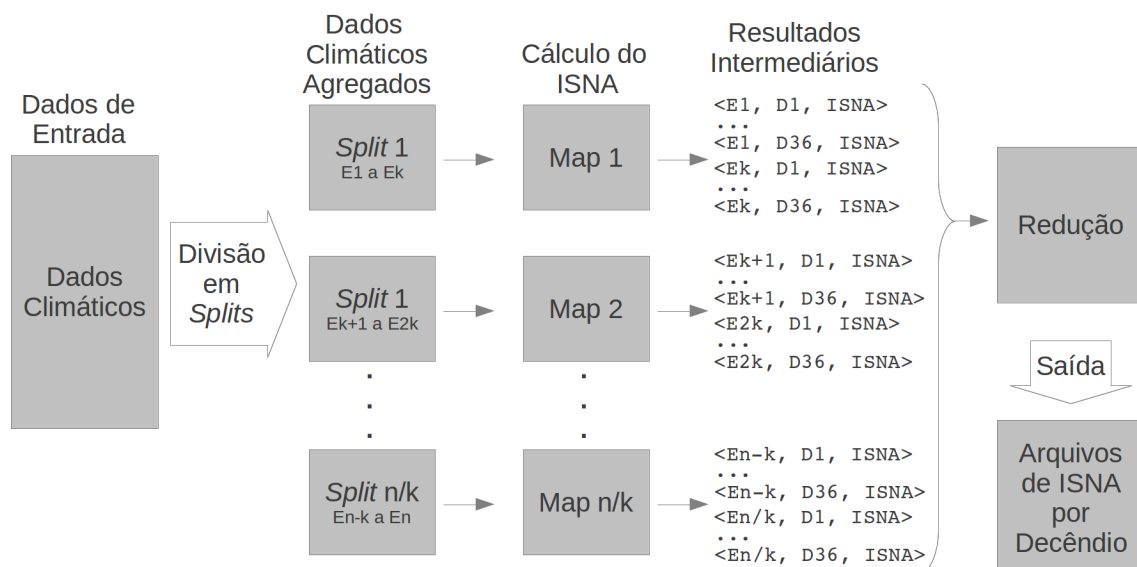


Figura 2. Esquema modificado da simulação de cenários agrícolas utilizando MapReduce; E_i é a estação i . k é número de estações por *split*; n é o número de estações.

- Normal: utilizou a implementação original com 3313 splits e mapeamentos;
- 75 splits: utilizou a implementação otimizada, com os dados agregados em 75 splits.
- 150 splits: utilizou a implementação otimizada, com os dados agregados em 150 splits.

Os números de *splits* foram escolhidos com base no número de *slots* de mapeamento. Se o número de *splits* é menor que o número de *slots*, o *cluster* é subutilizado. Se o número de *splits* é maior que o número de *slots*, o Hadoop deve escalonar os mapeamentos de forma que os *slots* de mapeamentos já terminados sejam realocados para outros mapeamentos.

Com 75 *splits*, o objetivo foi avaliar o tempo de execução da simulação com o poucas tarefas, mas sem subutilizar o sistema (apenas 5 *slots* não utilizados). Com 150 *splits*, o objetivo foi avaliar o impacto de duplicar o número de tarefas (em relação à simulação com 75 tarefas).

Cada tipo de simulação foi executada 10 vezes. A Figura 3 apresenta os tempos médios de execução para cada conjunto de simulações.

Pela figura, pode-se constatar que o tempo médio da execução da simulação normal (3313 *splits*) foi de 245 s. Diminuindo o número de *splits* para 75, o tempo médio de execução baixou para 66 s, uma redução de aproximadamente 73%. Este resultado mostra que a diminuição do número de *splits* teve um impacto significativo no tempo da execução da simulação.

Entretanto, um fato interessante é que, aumentando o número de *splits* de 75 para 150, o tempo médio de execução baixou para 62 s. Embora esta diferença seja relativamente pequena, este fato contradiz a hipótese inicial.

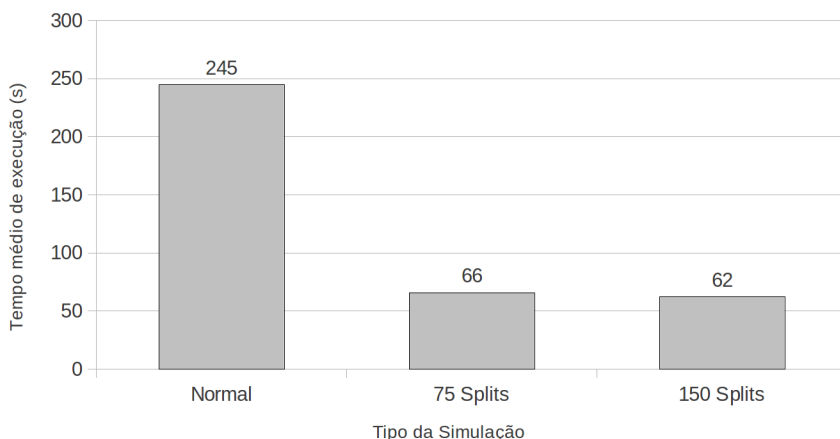


Figura 3. Tempos médios de execução.

A explicação para a diminuição do tempo de execução quando o número de *splits* aumentou de 75 para 150 é explicada pela ação de outros fatores. Apesar da diferença do número de *splits*, a quantidade de dados a serem processados pelo redutor, a segunda fase do MapReduce, continua a mesma. Acontece que, no Hadoop, por padrão, a fase de redução só é iniciada após 5% do mapeamento ser concluída. Se o número de *splits* é menor que o número de *slots*, e se as tarefas de mapeamento são executadas em tempos semelhantes, todos os mapeamentos tendem a terminar aproximadamente juntos. Com isso, a fase de redução é iniciada quando praticamente 100% do mapeamento é concluído. Isto acontece com as simulações com 75 *splits*. No caso das simulações com 150 *splits*, o redutor é iniciando antes, e, por isso, a simulação é um pouco mais rápida.

O início da fase de redução pode ser alterada na configuração do Hadoop. O escopo deste trabalho não abrangeu o estudo de variações desta configuração, mas certamente ela deve ser considerada em implantações de produção.

Conclusão

Este trabalho apresentou um estudo sobre a otimização do Hadoop MapReduce para tarefas pequenas, mais especificamente, para simulação de cenários agrícolas. Em resumo, as conclusões obtidas foram:

- O impacto do tempo gasto com o controle da distribuição de tarefas é significativo quando as tarefas de mapeamento são relativamente pequenas. A diminuição do número de tarefas, pela agregação destas, tem impacto positivo no tempo de execução;
- Um número muito pequeno de tarefas nem sempre é a melhor escolha, pois outros fatores, como o disparo do redutor, também influenciam no tempo de execução.
- Apesar de o ganho absoluto no tempo de execução ser pequeno (183 s), o ganho agregado é significativo em estudos que envolvem muitas simulações (milhares).

Vale ressaltar que, em muitos casos, mesmo com a agregação de tarefas, o MapReduce pode não ser a melhor opção de tecnologia. Para muitos problemas, nos

quais o tempo de execução é extremamente crítico, o desenvolvedor pode considerar modelos de programação de níveis mais baixos, como por exemplo *Message Passing Library* (MPL). Porém, utilizando este tipo de solução, problemas como escalonamento e recuperação de falhas, que são transparentes no Hadoop/MapReduce, devem ser tratados pelo desenvolvedor.

Referências

ASSAD, E. **Simulation de l'irrigation et du drainage pour les cultures pluviales de riz et de mais en sols de bas-fonds a Brasilia.**

Brasília, DF: CNPq / IRAT, 1983. 31p. Trabalho apresentado no Seminário sobre Relacoes Agua/Solo/Planta, Brasília, DF, 1983.

BORTHAKUR, D. **The hadoop distributed file system: architecture and design.** 2007. Disponível em: <http://hadoop.apache.org/common/docs/r0.18.0/hdfs_design.pdf>. Acesso em: 18 out. 2013.

DEAN, J.; GHEMAWAT, S. Mapreduce: simplified data processing on large clusters, **Communications of The ACM**, New York, v. 51, p. 107–113, Jan. 2008.

FOSTER, F. **Simulation du bilan hydrique des cultures pluviales: présentation et Utilisation du Logiciel Bip.** Montpellier: IRAT-CIRAD, 1984. 63 p.

LAM, C. **Hadoop in action.** Stamford: Manning, 2011. 325 p.

VAKSMANN, M. Le modèle BIPODE: logiciel. Bamako: IRAT, 1990. Não paginado.

Comunicado Técnico, 115

Embrapa Informática Agropecuária
Endereço: Caixa Postal 6041 - Barão Geraldo
13083-886 - Campinas, SP
Fone: (19) 3211-5700
Fax: (19) 3211-5754
<http://www.cnptia.embrapa.br>
e-mail: cnptia.sac@embrapa.com.br



Ministério da
Agricultura, Pecuária
e Abastecimento



1ª edição on-line - 2013

Todos os direitos reservados.

Comitê de Publicações

Presidente: *Sílvia Maria Fonseca Silveira Massruhá*

Membros: *Adhemar Zerlotini Neto, Stanley Robson de Medeiros Oliveira, Thiago Teixeira Santos, Maria Goretti Gurgel Praxedes, Adriana Farah Gonzalez, Neide Makiko Furukawa, Carla Cristiane Osawa (Secretária)*

Suplentes: *Felipe Rodrigues da Silva, José Ruy Porto de Carvalho, Eduardo Delgado Assad, Fábio César da Silva*

Expediente

Supervisão editorial: *Stanley Robson de Medeiros Oliveira, Neide Makiko Furukawa*

Normalização bibliográfica: *Maria Goretti Gurgel Praxedes*

Revisão de texto: *Adriana Farah Gonzalez*

Editoração eletrônica: *Neide Makiko Furukawa*