

Batch Processing

Fabiola Santore

Universidade Federal do Paraná

Sumário

1. Introdução
2. Ferramentas Unix
 - 2.1 Análise de log
 - 2.2 Filosofia Unix
3. MapReduce
 - 3.1 Procedimento
 - 3.2 Reduce: Joins e Agrupamento
 - 3.3 Análise de eventos de Atividade do Usuário
 - 3.4 Fluxo de Trabalho em Lote
 - 3.5 Comparando Hadoop com Banco de Dados Distribuídos
4. Outras Ferramentas
 - 4.1 Materialização

Sumário

1. Introdução

2. Ferramentas Unix

- 2.1 Análise de log
- 2.2 Filosofia Unix

3. MapReduce

- 3.1 Procedimento
- 3.2 Reduce: Joins e Agrupamento
- 3.3 Análise de eventos de Atividade do Usuário
- 3.4 Fluxo de Trabalho em Lote
- 3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

- 4.1 Materialização

Introdução

Sistemas de processamento em lote recebem uma grande quantidade de dados de entrada, executam um trabalho para processá-los e produzem alguns dados de saída, sendo geralmente programados para serem executados periodicamente.

Sumário

1. Introdução

2. Ferramentas Unix

- 2.1 Análise de log
- 2.2 Filosofia Unix

3. MapReduce

- 3.1 Procedimento
- 3.2 Reduce: Joins e Agrupamento
- 3.3 Análise de eventos de Atividade do Usuário
- 3.4 Fluxo de Trabalho em Lote
- 3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

- 4.1 Materialização

Ferramentas Unix

- Unix é um formato de sistema operacional;
- O Linux compartilha diversos conceitos originais do sistema Unix;
- Fácil e versátil para desenvolvimento de scripts e automatização de tarefas;
- Permite que o usuário execute uma sequência de comandos direto do terminal;
- Muitas análises são feitas em questão de poucos minutos usando combinações das ferramentas Unix: awk, sed, grep, sort, uniq e xargs.

Sumário

1. Introdução

2. Ferramentas Unix

2.1 Análise de log

2.2 Filosofia Unix

3. MapReduce

3.1 Procedimento

3.2 Reduce: Joins e Agrupamento

3.3 Análise de eventos de Atividade do Usuário

3.4 Fluxo de Trabalho em Lote

3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

4.1 Materialização

Análise de log

Digamos que você tenha um servidor da web que adapte uma linha a um arquivo de log toda vez que ele atender a uma solicitação:

Endereço IP do cliente

Em 27 de fevereiro de 2015, às 17:55:11

O servidor recebeu um pedido para o arquivo /css/typography.css

Status e tamanho da resposta ←

```
216.58.210.78 - - [27/Feb/2015:17:55:11 +0000] "GET /css/typography.css HTTP/1.1"
200 3377 "http://martin.kleppmann.com/" "Mozilla/5.0 (Macintosh; Intel Mac OS X
10_9_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.115
Safari/537.36"
```

↑

↑

↑

↓

Navegador Web

Como faz para encontrar as cinco páginas mais populares em seu Website?

Análise de log

Você pode fazer isso usando as ferramentas do Unix:

```
cat /var/log/nginx/access.log |
  awk '{print $7}' |
  sort |
  uniq -c |
  sort -r -n |
  head -n 5
```

```
awk '{print $7}'
↓
/css/typography.css
```

Análise de log

A saída dessa série de comandos fica assim:

```
4189 /favicon.ico
3631 /2013/05/24/improving-security-of-ssh-private-keys.html
2124 /2012/12/05/schema-evolution-in-avro-protocol-buffers-thrift.html
1369 /
915 /css/typography.css
```

Análise de log

- Um programa poderia ser utilizado para fazer a mesma coisa;
- A diferença entre uma cadeia de comando e um programa está no fluxo de execução;
- A melhor abordagem depende por exemplo, de quantos URLs diferentes se tem;
- Para um conjunto grande de URLs, a cadeia de comandos do Unix é facilmente dimensionada, sem ficar sem memória, sendo este o problema de determinados programas.

Sumário

1. Introdução

2. Ferramentas Unix

2.1 Análise de log

2.2 Filosofia Unix

3. MapReduce

3.1 Procedimento

3.2 Reduce: Joins e Agrupamento

3.3 Análise de eventos de Atividade do Usuário

3.4 Fluxo de Trabalho em Lote

3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

4.1 Materialização

Filosofia Unix

É um conjunto de princípios de design, baseados na ideia de conectar programas com pipes. Duas delas são consideradas as principais:

- Faça cada programa fazer apenas uma coisa bem. Para fazer um novo trabalho, crie de novo, em vez de complicar programas antigos adicionando novos "recursos";
- Sempre espere que a saída de cada programa se torne a entrada para outro programa.

Filosofia Unix

- O Unix possui uma interface uniforme;
- Muitos arquivos diferentes podem ser representados por essa interface;
- Faz o uso de entrada e saída padrão, respectivamente stdin e stdout;
- Essa abordagem permite, maior flexibilidade do usuário conectar a entrada e a saída da maneira que desejar.

Filosofia Unix

Parte do que torna as ferramentas Unix tão bem-sucedidas é que elas tornam muito fácil ver o que está acontecendo:

- Você pode executar os comandos sempre que quiser, tentando várias opções de linha de comando, sem danificar os arquivos de entrada;
- Você pode terminar o pipeline em qualquer ponto, olhar para os resultados parciais e ver se está como esperado;
- Você pode gravar a saída de um estágio de pipeline em um arquivo e usar esse arquivo como entrada para o próximo estágio.

Filosofia Unix

A maior limitação das ferramentas do Unix é que elas são executadas apenas em uma única máquina. É para esta finalidade que ferramentas como MapReduce são utilizadas.

Sumário

1. Introdução

2. Ferramentas Unix

2.1 Análise de log

2.2 Filosofia Unix

3. MapReduce

3.1 Procedimento

3.2 Reduce: Joins e Agrupamento

3.3 Análise de eventos de Atividade do Usuário

3.4 Fluxo de Trabalho em Lote

3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

4.1 Materialização

MapReduce

- O MapReduce é um modelo de programação paralela;
- Possui características parecidas com as ferramentas Unix;
- Principal diferença é por seu processo ser distribuído em diversas máquinas;
- Não modifica a entrada e não possui efeitos colaterais além de produzir a saída.

MapReduce

- As tarefas MapReduce leem e gravam arquivos de um Sistema de Arquivos Distribuídos(SAD);
- SAD é um sistema onde os arquivos são armazenados em hardware diferentes, interconectados através de uma rede;
- A implementação do MapReduce é feita utilizando o sistema de arquivos chamado HDFS (Hadoop Distributed File System);
- A paralelização é baseada no particionamento.

Sumário

1. Introdução

2. Ferramentas Unix

- 2.1 Análise de log
- 2.2 Filosofia Unix

3. MapReduce

- 3.1 Procedimento
- 3.2 Reduce: Joins e Agrupamento
- 3.3 Análise de eventos de Atividade do Usuário
- 3.4 Fluxo de Trabalho em Lote
- 3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

- 4.1 Materialização

Procedimento

Para criar um trabalho MapReduce, você precisa implementar duas funções:

Map É chamada uma vez para cada registro de entrada. Para cada entrada o seu trabalho é extrair a chave e o valor.

Reduce Processa os pares de chave-valor ordenados, integrando os valores pertencentes à mesma chave.

Procedimento

Usando o exemplo do log, a aplicação MapReduce ficaria assim:

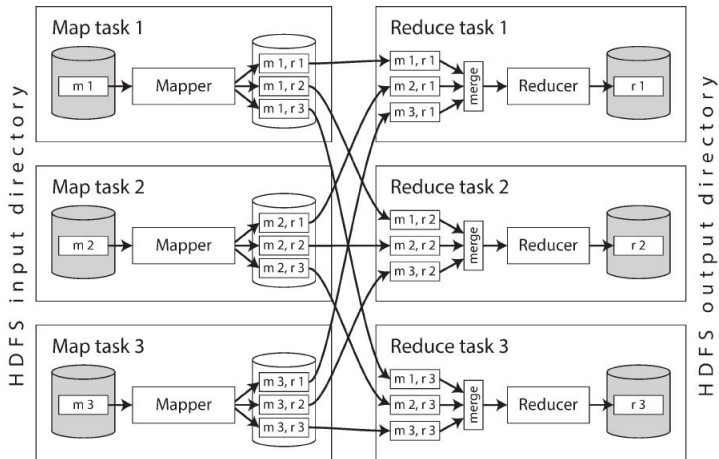
```
cat /var/log/nginx/access.log | ❶  
awk '{print $7}' | ❷  
sort | ❸  
uniq -c | ❹  
sort -r -n |  
head -n 5
```

1. Ler um conjunto de arquivos de entrada e dividi-los em registros.
2. Chame a função Map para extrair uma chave e um valor de cada registro de entrada.
3. Ordene todos os pares de chave-valor por chave.
4. Chame a função Reduce para integrar os pares de chave-valor ordenados.

Fluxo de Trabalho MapReduce

- O intervalo de problemas que você pode resolver com um único trabalho MapReduce é limitado;
- Trabalhos do MapReduce podem ser encadeados em fluxo de trabalho;
- A saída de um trabalho MapReduce se torna a entrada para o próximo trabalho.

Fluxo de Dados em Um Trabalho MapReduce



Sumário

1. Introdução

2. Ferramentas Unix

- 2.1 Análise de log
- 2.2 Filosofia Unix

3. MapReduce

- 3.1 Procedimento
- 3.2 Reduce: Joins e Agrupamento**
- 3.3 Análise de eventos de Atividade do Usuário
- 3.4 Fluxo de Trabalho em Lote
- 3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

- 4.1 Materialização

Joins e Agrupamento

- Em muitos conjuntos de dados, é comum que um registro tenha uma associação com outro registro;
- Uma junção é necessária sempre que algum código precise acessar arquivos em ambos os lados de associação;
- Junções no contexto de processamento em lote, é razoável resolver todas as ocorrências de uma associação em um conjunto de dados;
- Em um banco de dados, uma consulta que envolva apenas um pequeno número de registros, utilizado um índice para localizar rapidamente os registros de interesse;
- Além de técnicas Join, também podem ser utilizados GROUP BY, COUNT e SUM.

Sumário

1. Introdução

2. Ferramentas Unix

- 2.1 Análise de log
- 2.2 Filosofia Unix

3. MapReduce

- 3.1 Procedimento
- 3.2 Reduce: Joins e Agrupamento
- 3.3 Análise de eventos de Atividade do Usuário**
- 3.4 Fluxo de Trabalho em Lote
- 3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

- 4.1 Materialização

Análise de Eventos de Atividade do Usuário

User activity events

user 105 clicked button ... to load URL ...
user 296 viewed the profile of user 134
user 251 logged out from browser session ...
user 184 loaded page with URL ...
user 101 posted a response to message ...
user 156 clicked the help button on page ...
user 301 clicked a link in email ...
user 123 searched for the keyword ...

User database

user_id	email	date_of_birth
100	atilla@example.com	1991-02-15
101	beth@foo.com	1952-06-29
102	chunzhi@test.net	1967-09-08
103	devaraj@example.net	1947-12-18
104	evelyn@example.com	1989-01-30
105	flavio@foo.com	1971-05-05
...

Análise de Eventos de Atividade do Usuário

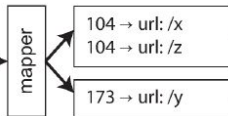
- Uma tarefa de análise pode precisar correlacionar a atividade com as informações do perfil do usuário;
- Os eventos de atividade precisam ser associados ao banco de dados do perfil do usuário;
- Colocando cada banco de dados dentro de um conjunto de arquivos no HDFS, é possível usar o MapReduce para reunir todos os registros relevantes no mesmo local e processá-los com eficiência.

Análise de Eventos de Atividade do Usuário

User activity mapper

```

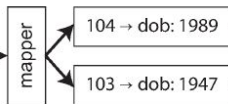
user 104 loaded URL /x
user 173 loaded URL /y
user 104 loaded URL /z
    
```



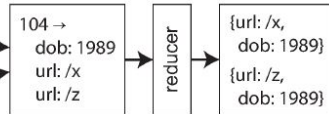
User database mapper

```

{user_id: 103,
 email: devaraj@example.net,
 date_of_birth: 1947-12-18}
{user_id: 104,
 email: evelyn@example.com,
 date_of_birth: 1989-01-30}
    
```



Reducer partition 1 (even-numbered user IDs)



Reducer partition 2 (odd-numbered user IDs)



Sumário

1. Introdução

2. Ferramentas Unix

- 2.1 Análise de log
- 2.2 Filosofia Unix

3. MapReduce

- 3.1 Procedimento
- 3.2 Reduce: Joins e Agrupamento
- 3.3 Análise de eventos de Atividade do Usuário
- 3.4 Fluxo de Trabalho em Lote**
- 3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

- 4.1 Materialização

Fluxo do Trabalho em Lote

Aplicação de consultas em bancos de dados:

- Processamento de transações;
- Processamento analítico;

Por que executamos todos esses trabalhos MapReduce? Qual é o resultado de todo esse processamento? Onde o processamento em lote se encaixa?

Fluxo do Trabalho em Lote

O processo em lote geralmente é executado em grandes partes de um conjunto de dados de entrada. Porém a saída de um processo em lote geralmente não é um relatório, mas algum outro tipo de estrutura.

Criando índices de pesquisa

- O uso original do MapReduce pelo Google era criar índices para o mecanismo de pesquisa;
- O processo em lote é eficaz quando se é preciso realizar uma pesquisa de textos completa sobre um conjunto fixo de documentos:
 - Cada tarefa MAp particiona o conjunto de documentos conforme necessário;
 - Cada tarefa Reduce constrói o índice para sua partição;
 - Os arquivos de índices são gravados no SAD.

Armazenando chaves-valores

- Construção de sistemas de aprendizado de máquinas, como classificadores e sistemas de recomendações;
- A saída dessas tarefas em lote é um tipo de banco de dados que pode ser consultado pelo ID, o que facilita diversas aplicações;
- Por exemplo, um banco de dados que pode ser consultado pelo ID do produto, para obter uma lista de produtos relacionados.

Sumário

1. Introdução

2. Ferramentas Unix

2.1 Análise de log

2.2 Filosofia Unix

3. MapReduce

3.1 Procedimento

3.2 Reduce: Joins e Agrupamento

3.3 Análise de eventos de Atividade do Usuário

3.4 Fluxo de Trabalho em Lote

3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

4.1 Materialização

Comparando Hadoop com Banco de Dados Distribuídos

- MapReduce é uma "implementação" de um processo Unix;
- HDFS é o sistema de arquivos distribuídos;
- Assim o Hadoop é um pouco como uma versão distribuída do Unix;
- Os banco de dados distribuídos se concentram na execução paralela de consultas SQL analíticas, em um cluster de máquinas;
- O Hadoop fornece algo muito mais parecido com um sistema operacional de propósito geral que pode executar programas arbitrários.

Filosofia MapReduce

- O Google possui datacenters de uso misto;
- Toda tarefa tem alocação de recursos e prioridades;
- Como tarefas MapReduce são prioridade baixa, possuem maior chance de serem interrompidas. É por isso que seu sistema é designado para tolerar esse tipo de falha;
- Não é indicado para ambientes que as tarefas não são frequentemente terminadas;

Sumário

1. Introdução

2. Ferramentas Unix

- 2.1 Análise de log
- 2.2 Filosofia Unix

3. MapReduce

- 3.1 Procedimento
- 3.2 Reduce: Joins e Agrupamento
- 3.3 Análise de eventos de Atividade do Usuário
- 3.4 Fluxo de Trabalho em Lote
- 3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

- 4.1 Materialização

Outras Ferramentas

- O MapReduce é apenas um dentre muitos modelos de programação possíveis para sistemas distribuídos.;
- Dependendo do volume de dados, da estrutura dos dados e do tipo de processamento que os utiliza, outras ferramentas podem ser mais apropriadas;
- Porém o MapReduce é uma abstração razoavelmente clara e simples sobre um sistema de arquivos distribuído.;
- Alguns modelos de programação de tarefas de processamento de lote, são similares ao MapReduce, porém mais fáceis de implementar (Pig, Hive, Cascading, Crunch) ;

Sumário

1. Introdução

2. Ferramentas Unix

- 2.1 Análise de log
- 2.2 Filosofia Unix

3. MapReduce

- 3.1 Procedimento
- 3.2 Reduce: Joins e Agrupamento
- 3.3 Análise de eventos de Atividade do Usuário
- 3.4 Fluxo de Trabalho em Lote
- 3.5 Comparando Hadoop com Banco de Dados Distribuídos

4. Outras Ferramentas

- 4.1 Materialização

Materialização

- Todo trabalho MapReduce é independente de qualquer outro trabalho;
- Possui a disponibilidade da saída de um trabalho se tornar a entrada de um segundo trabalho;
- Alguns arquivos no SAD são simplesmente estado intermediário.

Materialização

Problemas causados pela materialização:

- Ter que esperar até que todas as tarefas do trabalho anterior sejam concluídas atrasa a execução do fluxo de trabalho como um todo;
- Em muitos casos, o código da função Map poderia ser parte da função Reduce anterior;
- Um estado intermediário de armazenamento em um SAD, significa que esses arquivos são replicados em vários nós, o que geralmente é um exagero para esses dados temporários.

Materialização

- Alguns mecanismos de execução foram criados para tratar do problema de materialização (Spark, Tez, Flink);
- A similaridade entre eles é determinada com um fluxo de trabalho inteiro como um único trabalho, em vez de dividi-lo em submissões independentes;
- O Flink incrementalmente passa a saída de um operador para outros operadores, não esperando que a entrada esteja completa antes de começar a processá-la.

Resumo

- Processamento em lote;
- Filosofia e ferramentas Unix;
- MapReduce e Sistemas de Arquivos Distribuídos;
- Seguem o princípio que as entradas são imutáveis, as saídas destinam-se a ser entrada para outro programa;
- Técnicas de junção e agrupamento;
- Problemas de materialização do estado intermediário;

Perguntas

1. Explique como funciona o processamento em lote com as ferramentas Unix e apresente uma aplicação prática para a qual o modelo é apropriado.
2. Explique como funciona o processamento em lote com o MapReduce e qual a diferença entre ele e as ferramentas Unix. Por fim apresente uma aplicação prática para a qual o modelo é apropriado.