1 - Sobre mapReduce com o exemplo de temperatura, como no EP. Calcular a temperatura por mês

A) Porque o mapreduce é tão popular e tão importante

Código aberto

- Comunidade ativa
- Apoio de grandes corporações
- Correções de erros frequentes
- Constante evolução do arcabouço

Econômico

- Software livre
- Uso de máquinas e redes convencionais
- Aluguel de serviços disponíveis na nuvem:
  - Amazon Elastic MapReduce
  - Google App Engine MapReduce
  - etc.

Robusto

- Se em 1 máquina há probabilidade de haver falhas...
- Tempo médio entre falhas para 1 nó: 3 anos
- Tempo médio entre falhas para 1.000 nós: 1 dia

Estratégias

- Replicação dos dados
- Armazenamento de metadados

Escalável

- Permite facilmente adicionar máquinas ao aglomerado
- Adição não implica na alteração do código-fonte
- Limitação apenas relacionada a quantidade de recursos disponíveis

Foco na regra de negócio

- Hadoop realiza todo o “trabalho duro”
- Desenvolvedores podem focar apenas na abstração do problema

slides 26:31 - aula10.pdf

B) Duas desvantagens de MapReduce

Map reduce não é indicado para:

a) processamento em tempo real
b) aplicações que precisam realizar comunicação entre tarefas
c) processamento de fluxo contínuo de dados
d) aplicações que necessitam de garantias transacionais (OLTP)
e) problemas difíceis de serem expressados com a abstração proporcionada pelo modelo MapReduce

slide 60 - aula10.pdf

Único nó mestre • Ponto único de falha • Pode impedir o escalonamento Dificuldade das aplicações paralelas • Problemas não paralelizáveis • Processamento de arquivos pequenos • Muito processamento em um pequeno conjunto de dados

slide 32 - aula10.pdf

C) **Pseudo-código como funcionaria para calcular a média de temperatura com conjunto de dados dado**

https://www.youtube.com/watch?v=8wjvMyc01QY esse vídeo pode ajudar em como fazer o código, mas seria legal se alguém colocasse aqui um exemplo

Dado um arquivo com dados meteorológicos, calcular a temperatura média por mês.
Data Precipitacao;Temperatura; Umidade Relativa; Velocidade do Vento;
01/01/1990 15.5;22.24;88;1.766667;
02/01/1990 35.9;21.2;89.75;2.333333;
...
30/09/2013 0;18.34;91.25;1.54332;
01/10/2013 6.6;19.94;80.25;2.74368

```
public static class MonthTempMapper
     extends Mapper<Text, Text, IntWritable, FloatWritable> {

  private IntWritable mes         = new IntWritable();
  private FloatWritable temperatura = new FloatWritable();

  public void map(Text key, Text value, Context context)
      throws IOException, InterruptedException
  {
    // key contém a data (dd/mm/aaaa)
    String   chave = key.toString();
    String[] valor = value.toString().split(";");

    if(chave.charAt(0) == '#' || valor.length != 4 || valor[1].isEmpty())
      return; // linha comentada ou com valor faltando

    // mês; as datas seguem o formato dd/mm/aaaa
    int mes = Integer.parseInt(chave.substring(3,5));

    // value contém os dados meteorológicos separados por ";"
    float temperatura = Float.parseFloat(valor[1]);

    context.write(new IntWritable(mes), new FloatWritable(temperatura));
  }
}
```

```
public static class AverageReducer
  extends Reducer<IntWritable,FloatWritable,Text,FloatWritable> {
  private FloatWritable media = new FloatWritable();

  public void reduce(IntWritable key, Iterable<FloatWritable> values,
                     Context context
                     ) throws IOException, InterruptedException {
    float sum = 0.0f;
    int length = 0;
    for (FloatWritable val : values) {
      sum += val.get();
length += 1;
    }

    media.set(sum/length);

    String[] nomeDoMes = {"Jan", "Fev", "Mar", "Abr", "Mai", "Jun",
                          "Jul", "Ago", "Set", "Out", "Nov", "Dez"};
    Text mes = new Text(nomeDoMes[key.get()-1]);

    context.write(mes, media);
  }
}
```

Slide 62:63 - aula10.pdf

Podemos usar o pseudo código do word count, substituindo a palavra pelo mês, e o valor pela temperatura, no reducer somar as temperaturas pra cada valor com chave mês, acrescentar um count no for.
Eu acho bem provável que ele mude o tipo de problema, e use aqueles do final da aula 10, citações, anagramas, …. mas a mecânica é a mesma.
Pode ser que caia o calculo do desvio padrão tbm

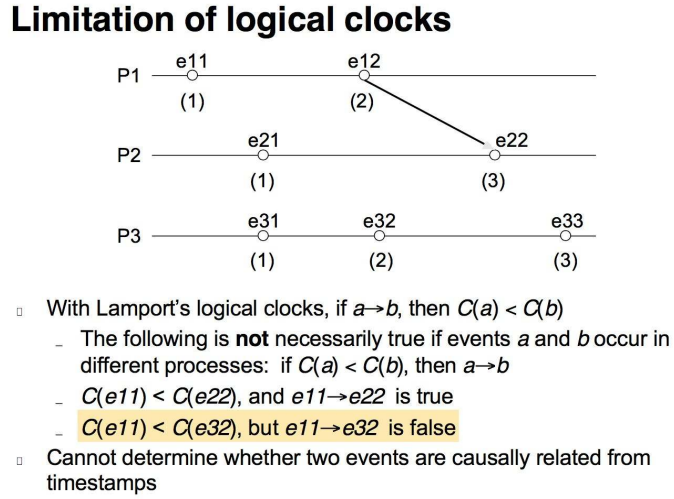
2 - Sobre Lamport (relógios lógicos)

A) Qual definição ele introduz (em relação aos vetoriais?)

Relógios físicos são muito difíceis de sincronização em sistemas distribuídos, e como só precisamos saber a ordem em que os eventos ocorreram, o conceito "aconteceu-antes" surge. O relógio lógico de Lamport traz a visão global do comportamento do sistema em relação ao "aconteceu-antes". Ele propõe que seja mantido um conjunto de relógios lógicos consistentes, um para cada processo, de forma que ele seja incrementado a cada evento e passado junto à mensagem para o outro processo, outro processo quando recebe esta mensagem ajusta o seu relógio local de acordo com max{Cj, ts(m)} -- se o timestamp recebido na mensagem for maior que seu relógio, o substitui, se não, o mantém inalterado --.

B) Mostrar um exemplo mostrando porque o Lamport não define uma relação causal

Utilizando os relógios lógicos de Lamport, tem-se que se A -> B então C(A) < C(B), porém o contrário não pode ser afirmado pois este relógio não mantém a relação de causalidade, somente a ordenação parcial. Exemplo:

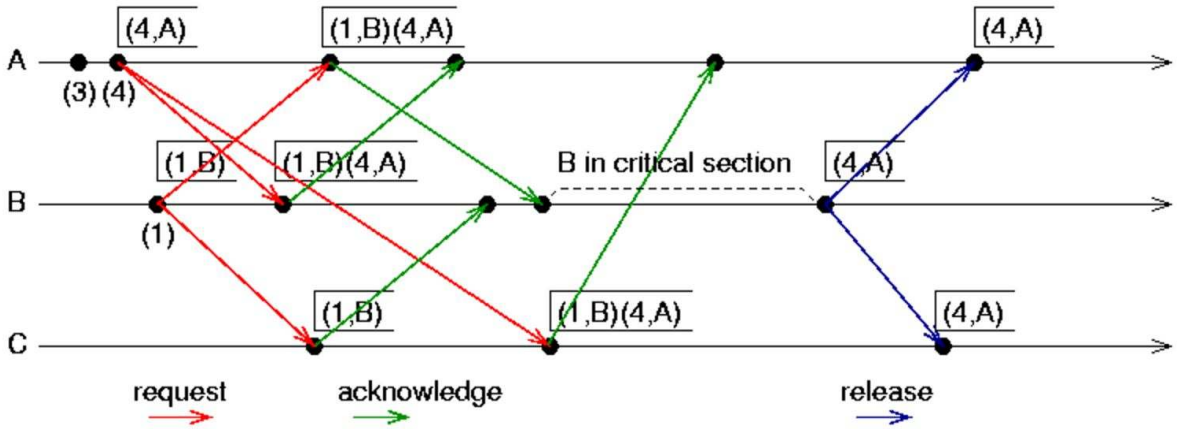


C) Explicar como, usando o Lamport, fazer a exclusão mútua.

- Assumes messages are delivered in FIFO order between each pair of sites
- Each request gets a timestamp
- Requests with lower timestamps take priority over requests with higher timestamps
- Each site maintains a queue of pairs (timestamp, site), ordered by timestamp

Algoritmo:

- Request
    - Si sends REQUEST(tsi, i) to all sites in its request set Ri and puts the request on request_queuei
    - when Sj receives REQUEST(tsi, i) from Si it returns a timestamped REPLY to Si and places Si's request on request_queuej
- Si waits to start the CS until both
    - [L1:] Si has received a message with timestamp > (tsi, i) from all other sites
    - [L2:] Si's request is at the top of request_queuei
- Release
    - Si removes request from top of request_queuei and sends time-stamped RELEASE message to all the sites in its request set
    - when Sj receives a RELEASE messages from Si it removes Si's request from request_queuej



3 - Relógios Vetoriais

A) O que são e quais as diferenças entre os de Lamport (o que um garante e outro não)

- Mantém um vetor de relógios de cada processo junto com a mensagem.
- Mantém a propriedade de causalidade.

B) Um exemplo real de troca de mensagens entre várias pessoas com timestamps para descobrir qual foi a última mensagem entregue para decidir.
https://youtu.be/lT5AqUEGLwM?t=2m36s

4 - Sobre consistência

A) O que é consistência sequencial: Todas operações deverão aparecer na mesma ordem em todos os processos.

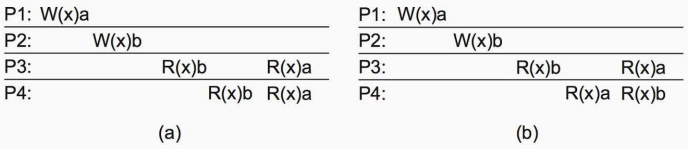
B) O que é consistência causal: Apenas as operações causais de escrita deverão aparecer na mesma ordem em todos os processos.

C) Exemplos das consistências (desenhos dos slides):

CONSISTÊNCIA SEQUENCIAL

Definição

O resultado de qualquer execução é o mesmo, como se as operações de todos os processos fossem executadas na mesma ordem sequencial e as operações de cada processo aparecer nessa sequência na ordem especificada pelo seu programa.

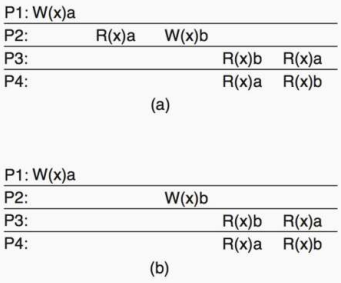


(a) é um *data store* com consistência sequencial; (b) não apresenta consistência sequencial

CONSISTÊNCIA CAUSAL

Definição

Operações de escrita que potencialmente possuem uma relação de causalidade devem ser vistas por todos os processos na mesma ordem. Escritas concorrentes podem ser vistas em uma ordem diferente por processos diferentes.



(a) uma violação da consistência causal; (b) uma sequência correta de eventos em um *data store* com consistência causal

5 - Fale sobre o curso :)

Uma bosta digasse de passage

Pelo menos teremos um ponto de graça na P2 :-)rs