

Já na primeira frase, do blog da Cloudera, temos a afirmação que trabalhar com arquivos pequenos no Cloudera Hadoop é um grande problema "Small files are a big problem in Hadoop"

Excelente referência : <http://blog.cloudera.com/blog/2009/02/the-small-files-problem/>

Mas por quê?

Primeiro minha opinião ok.

O HDFS (Hadoop Distributed File System), foi projetado para trabalhar com arquivos gigantescos e não com arquivos com mega ou Kb.

Haja visto que os blocos do hadoop, no início eram de 64Mb, hoje os blocos recomendados pela Cloudera são de 256Mb.

Segundo o blog da Cloudera pequenos arquivos (significativamente menores que 64Mb, o recomendado pela Cloudera na época que este blog foi escrito) podem fazer com que o Cloudera perca o controle de muitos arquivos. Se você trabalha com arquivos pequenos, você deveria reavaliar se um banco de dados tradicional não poderia resolver seu problema.

Cada arquivo, diretório e block no HDFS, representa um objeto que no namenode consome de memória 150 bytes, isso é o que eles chamam de regra de outro. Sendo assim, 10 milhões de arquivos, cada um utilizando um bloco consome 3 gigabytes de memória. Neste mundo de Big Data, 10 milhões de arquivos não é absolutamente nada, logo-logo, vamos chegar a bilhões de objetos e isso vai tornar a arquitetura de hardware atual inviável.

Hoje o mapreduce e o HDFS conseguem gerenciar arquivos pequenos, a Cloudera recomenda hoje blocos de arquivos com 256Mb e a tendência é a de aumentar o tamanho do bloco melhorando assim o desempenho das aplicações. No início o bloco inicial recomendado era de 64Mb.

O HDFS foi projetado para acessar enormes blocos de dados, leituras de arquivos pequenos vai forçar o DataNode procurar dados pequenos através de vários DataNodes e isso é ineficiente para o HDFS.

Infelizmente muitas pessoas pensam que pelo fato de um arquivo estar no HDFS ele já está acessível através de uma tabela Hive e/ou Impala ou Hbase. A única coisa que isso quer dizer é que o arquivo está armazenado em um sistema de arquivo que o replica por padrão 3x.

Por exemplo, uma tarefa de map para 1Gb de arquivo, se dividirá em 16 blocos utilizando-se poucos mapas para resolver a leitura de 64Mb enquanto que por

exemplo 10Mb vai gerar 10.000 arquivos de 10Kb gerando um grande número de mapas para ler estes 100Mb.

Então senhores não sou eu que estou dizendo que ter um bloco de 256Mb e arquivos com 0Kb ou 1Kb ou 1Mb são extremamente prejudiciais ao Hadoop.

Leiam, estudem, façam a coisa como deve ser feita para não termos problemas no futuro.

Abaixo alguns links complementares sobre os blocos.

<http://inquidia.com/news-and-info/working-small-files-hadoop-part-1>

<http://inquidia.com/news-and-info/working-small-files-hadoop-part-2>

<http://inquidia.com/news-and-info/working-small-files-hadoop-part-3>

Outra coisa que me incomoda muito. Toda linguagem de programação tem suas palavras reservadas. Bom se uma tecnologia criou uma palavra reservada é porque por exemplo esta palavra não pode ser utilizada como uma variável e/ou como uma coluna em um banco de dados.

Abaixo estão as palavras reservadas do Apache Hive.

<https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DDL>

Só como exemplo, não podemos criar uma coluna no Hive com o nome "timestamp" simplesmente porque "timestamp" é um data type.

E não venha me dizer que pode-se utilizar caracter de escape para resolver isso, isso é coisa de júnior.

Espero de coração ter ajudado alguém com este post.

Abraços

Pequeno

Conhecimento gera Conhecimento!

Projeto Spartacus