Spark / Cluster Hadoop ARM executado com Cubieboards

Um chip ARM não é recomendável para o processamento de arquivos grandes. No entanto, ele foi gradualmente se tornando forte o suficiente para fazê-lo em conjunto com outros chips. Muitas pessoas têm tentado executar o Apache Hadoop em cima de um cluster ARM. Em uma notícia publicada em agosto de 2014, cubies maníacos divulgaram que foram capazes de executar o Hadoop em uma máquina de 8-nodes.

Além disso, Jamie Whitehorn parece ser a primeira pessoa a executar com sucesso o Hadoop em um cluster Raspberry Pi em outubro de 2013. Ambos mostram que um cluster ARM é bom para levantar e executar o Hadoop.

Mas será que é viável fazer um Big Data confiável em um cluster ARM de baixo custo?

Com I / O e rede de SoCs ARM, o MapReduce do Hadoop atualmente não é capaz de processar um Big Data de 15GB, em média, por arquivo.

Sim, a verdadeira manipulação do Big Data não é tão grande. E, muitas vezes, fazer o processamento em lote por meio de uma lista de arquivos, não é maior que 15GB. Portanto, se um cluster é capaz de processar um arquivo de 15GB, é bom o suficiente para um Big Data.

Agora, respondendo a pergunta acima, temos um protótipo de um cluster baseado em ARM, projetado para processar um Big Data. É uma Cubieboard A10 22-nodes, ou seja, 22 placas Cubieboard A10, com 100 Mbps Ethernet. Veja a foto abaixo:

O cluster rodando Spark and Hadoop

Como o Map Reduce do Hadoop não é uma boa escolha para processar este tipo de cluster, decidimos usar apenas HDFS.

Embora o cluster tenha 20GB total de RAM, há apenas 10 GB disponíveis para o processamento de dados. Se tentar alocar uma quantidade maior de memória, alguns nodes vão parar de funcionar durante o processamento.

Então, qual é o resultado? Nosso cluster é bom o suficiente para armazenar todos os artigos do Wikipedia desde 2012 com um tamanho de 34GB.

Seu tamanho é 2 vezes maior do que o tamanho médio de um arquivo de Big Data, mencionado acima. Além disso, é 3 vezes maior do que a memória atribuída para processar dados.

Simplesmente executamos um programa de contagem de palavras no Spark's shell e esperamos por 1h50min até que, finalmente, o cluster respondeu que o arquivo contém 126.938.368 palavras com "a" dentro do Wikipedia.

O cluster gastou cerca de 30,5 horas no total em todos os nodes.

O resultado foi impresso a partir do Spark's shell

Observação

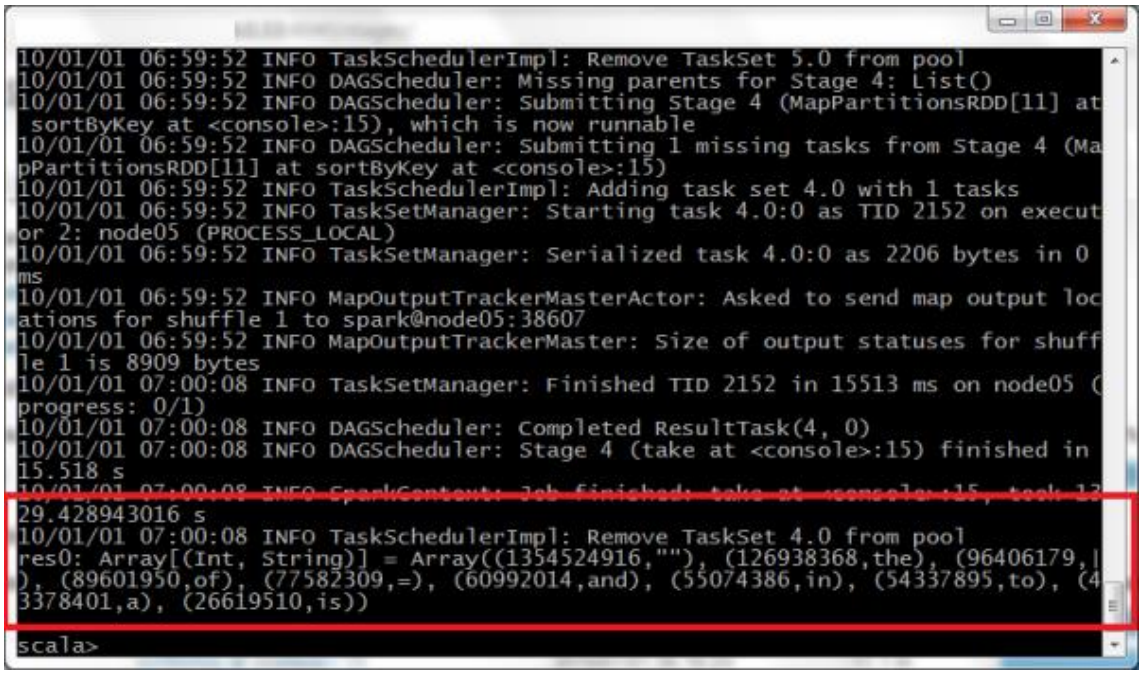
Temos 20 Spark worker nodes, e dois deles também executam Hadoop Data Nodes. Isso nos permite compreender a localidade de dados de nossa Spark cluster / Hadoop.

Executamos Hadoop's Name Node e Spark's Master Node na mesma máquina. A outra máquina é o driver. O arquivo é armazenado em 2 SSDs com SATA conectados a Nodes de dados.

Temos observado que o bottleneck atual pode ser de 100 Mbps Ethernet, mas ainda não temos uma chance de confirmar isso até criarmos um novo cluster com 1 Gbps Ethernet.

Nós temos 2 fontes de alimentação conectados e o cluster parece não consumir muita energia. Vamos medir isso em detalhes mais tarde.

É ótimo ter um cluster em execução sem um custoso centro de dados, certo?



Referência: cubieboard.org