評論

優點:

Spark目前已成熟，數據處理工具包可以對大量數據集進行處理，且不用擔心底層架構。工具包可以進行數據採集、查詢、處理，也可以進行機器學習，進而建構出分散式系統的數據抽象模型。

處理速度也是Spark的優勢之一，MapReduce在處理過程中將數據放到內存中，而不是放在磁碟上進行持久化，使Spark的處理速度提升。

應用方面:

1.數據分析:即對進入的數據流作實時分析。Spark能夠高效處理來自各數據源的大量數據，支持HDFS、Kafka、Flume、Twitter和ZeroMQ，也能對自定義的數據源進行處理。

2.趨勢數據:Spark能夠用來對進入的事件流進行處理，用於計算趨勢數據。

3.物聯網:物聯網系統產生了大量的數據，而這些數據會被推往後台處理。Spark能夠建構出數據管線，在特定的時間間隔內進行轉換，還可以基於一組事件觸發一系列動作。

4.機器學習:由於Spark能夠對數據進行批量處理，並且提供機器學習類庫（MLib），因此我們能夠對數據集使用機器學習算法，結合MLib與Streaming一同使用，就可以構建起機器學習系統。

缺點:

1.複雜的部署過程

Spark支持Mesos和Yarn，但如果對這兩者任一個不熟悉的話，部署過程就會變得艱難。如果不能正確處理，Spark雖然會單獨運行，但在cluster模式下，會遇到發出Classpath異常的情況。

2.內存問題

由於Spark被用來處理海量數據，因此必須對內存的使用情況進行監控，並對Spark內存相關的設置進行調整。

3.頻繁的版本更新導致API發生變化

Spark每三個月就進行一次副版本發布；而每隔三到四個月，進行一次主版本發布。儘管頻繁的版本發布代表推出了更多的功能，但這也意味著這些版本更迭的背後，某些情況下會要求API也要發生變化。如果沒有意識到新版本所帶來的變化，就會陷入困境，而若想要確保Spark應用不受這些版本更迭變化影響，也會帶來額外的開銷。

4.對Python的功能不完善

Spark支持Scala、Java和Python，但是Spark的最新版本中，對Python語言API的支持不像對Java和Scala語言的支持完善。如果使用Spark最新版本，可能需要用Scala或Java語言來實現，至少需要檢查是否已經有Python版本功能或API的實現。

結語

Spark在運算時，將中間產生的資料暫存在記憶體中，因此可以加快執行速度。尤其需要反覆操作的次數越多，所需讀取的資料量越大，越能看出Spark的效能，因此Spark適合用來處理大量數據，但如果只是要處理小規模數據，spark就並非首選，因其處理程序較為複雜，反而會消耗更多的時間。

在這門課學了系統優化、大型系統、分層...等等，而除了這些課內知識，更體會到了實作的樂趣，在做每個作業時就像進入了新世界，了解到課內內容的實際應用。