김일성종합대학학보
정보과학

주체108(2019)년
제65권 제4호

망자료흐름에서 중간결과를 위한 자료형식선택의 한가지 방법

리경심, 리일남

Hadoop클라스터상에서 대규모의 자료분석은 자료집중형흐름 DIW(Data Intensive Workflow)에 의하여 수행되고있다.

Hive와 Pig는 Hadoop상에서 분석과제실행을 쉽게 하기 위한 프레임워크로서 여러개의 관흐름화된 MapReduce일감[2]들에 과제들을 분배한다.

모든 과제수속결과들은 중간결과를 참고하며 중간결과들은 련속적으로 일어나는 이후의 과제에 재리용된다.

MapReduce프레임워크의 동작과정에 발생되는 작업량을 분석한데 의하면 중간결과의 80%가 DIW(Data Intensive Workflow)의 여러 부분에서 재호출된다는것을 알수 있다. 재호출하는 자료흐름의 속도를 높이기 위해 타당한 중간결과들을 실체화하는것이 중요하다.

분산Hadoop화일체계 HDFS[2]상에서는 실체화를 위한 입출력연산비용이 크다.

그러므로 불필요한 읽기/쓰기수행으로 DIW의 실행비용이 증가한다.

선행연구[2]에서는 읽기/쓰기연산량을 줄이기 위해 빠른 적재, 빠른 질문처리, 효과적인 기억리용을 위한 자료형식을 제안하였다. 그러한 자료형식에는 RCFile, Avro, Parquet 그리고 SequenceFile들이 있다.

선행한 자료형식들은 서로 론의하는 구조가 다르다.

Avro에서는 실체화를 위해 수평구조를 리용한다면 Parquet에서는 혼성구조를 리용하고있는데 이것들은 다 넓은 의미에서 최상의 선택이라고 볼수 없다.

서로 다른 작업(workload)은 서로 다른 구조를 요구하므로 하나의 고정된 자료형식으로 중간결과들을 저장하는것은 불합리하다.[1]

리상적으로 실체화된 결과들은 가장 적합한 형식으로 그 이후의 재리용을 위해 저장해야 한다.

론문에서는 적당한 자료형식을 선택하여 이후의 련속적인 리용에서 중간결과의 적재시간을 줄이고 전체적으로 DIW의 실행비용을 줄일수 있는 방법을 제안하였다.

1. 자료형식선택을 위한 규칙

실체화된 중간결과를 위한 자료형식을 선택하는 규칙에 앞서 모형에 대한 형식화를 진행하였다.

$DIW \leftarrow DAG(V, E)$

$V = \{v_1, v_2, \cdots, v_n\}$, $E = \{e_1, e_2, \cdots, e_n\}$

$M \subseteq V$, $\forall x \in M, O(x) \subseteq E$

$getOP : E \rightarrow \{op_1, op_2, \cdots, op_n\}$

$getType : E \rightarrow \{Type_1, Type_2, \cdots, Type_n\}$

$getCol_{op} : op \rightarrow P\{col_1, col_2, \cdots, col_n\}$

$getCol_v : V \rightarrow P\{col_1, col_2, \cdots, col_n\}$

$getBest : M \rightarrow \{format_1, format_2, \cdots, format_n\}$

DIW는 정점 V와 변두리 E로 이루어진 그룹이다.

하나의 정점은 하나의 자료묶음을 나타내며 하나의 변두리는 하나의 연산을 의미한다. 즉 변두리는 시작정점자료에 적용한 연산이다. 끝정점은 입력자료를 처리한 후의 연산자에 의하여 전달되는 자료를 의미한다. 하나의 변두리는 스키마정보들로 이루어진다.

함수 $getOP$와 $getType$는 하나의 주어진 변두리에 해당한 연산자의 실체와 형태를 얻는다. 그외에 함수 $getCol_{op}$는 연산이 수행된 렬묶음을 얻는 함수이며 $getCol_v$는 하나의 정점의 렬묶음을 얻는 함수이다. 그리고 M은 정점의 부분묶음으로서 실체화된 마디, $O(x)$는 변두리의 부분묶음으로서 실체화된 마디로부터 나가는 변두리이다. 결국 하나이상의 형식을 결정하는 규칙들중의 부분조합은 실체화된 마디를 위해서 적합하다.

론문에서는 그것들중 하나를 선택하기 위해 함수 $getBest$를 제안하였다.

주어진 실체화된 마디를 위해 어떤 형식을 선택하겠는가를 결정하는 규칙은 잘 알려진 수직, 수평, 혼합구조들로부터 도출된다.

규칙 1 $x \rightarrow$ SequenceFile, IF size($getCol_v(x)$) = 2

규칙 2 $x \rightarrow$ Parquet, IF $\exists e \in O(x)$, WHERE $getType(e) \in \{AggregationOPs\}$

규칙 3 $x \rightarrow$ Parquet, IF $\exists e \in O(x)$, WHERE $getCol_{op}(getOP(e)) \subseteq getCol_v(x)$

규칙 4 $x \rightarrow$ Avro, IF $\forall e \in O(x)$, WHERE $getCol_{op}(getOP(e)) = getCol_v(x)$

규칙 5 $x \rightarrow$ Avro, IF $\exists e \in O(x)$,

WHERE $getType(e) \in \{Join, CartesianProduct, GroupALL, Distinct\}$

규칙 1은 2개의 렬로 이루어진 실체화된 마디에 대해서는 SequenceFile을 선택한다.

규칙 2는 자료들을 통합할 때 Parquet를 선택한다. 그것은 Parquet가 매 렬들에 대하여 정적인 정보들을 저장하기때문에 통합할 때 가장 효과적이기때문이다.

규칙 3은 Parquet가 혼성구조를 실현하였기때문에 자료의 부분묶음을 읽을 때 그리고 렬들의 부분조합에 연산을 적용할 때 가장 좋은 선택이라고 본다.

반대로 Avro는 모든 자료를 읽거나 연산을 렬들의 부분조합에 적용하지 않을 때 선택한다. 이것은 수평구조를 실현하는 Avro의 련속적인 과정이다. 규칙 4와 5는 Avro를 리용한다.

정의된 탐색규칙은 서로 배제한다. 즉 정해진 순서가 없이 서로 독립적으로 적용된다. 또한 규칙들은 혼성구조에 의하여 부분합계되므로 수직구조를 론의하지 않는다.

탐색규칙들을 적용할 때 다중선택이 적합한 경우들이 있으므로 이때 함수 $getBest$를 리용한다.

이 함수는 Parquet에 제일 높은 우선권을 할당한다. 그것은 Parquet가 많은 특성들을 가지고있고 하나이상의 유연한 동작들을 수행할수 있기때문이다. 두번째 우선권은 Avro에 할당한다. 그것은 자료를 읽는 속도가 높고 다음 스키마정보를 저장하기때문이다.

끝으로 제일 마지막순서는 SequencegFile이다. SequenceFile은 열쇠－값형태의 자료를 선택할 때에만 리용한다.

2. 자료형식선택에 기초한 중간결과의 저장

PROJECT와 FILTER연산자가 출력되는 실체들의 자료량을 줄이는 연산이라면 JOIN, GROUP, CoGROUP(GROUP BY와 JOIN연산결합)연산자들은 계산집중적인 연산이다. 이러한 연산들은 DIW에서 련속적으로 재리용되므로 실체화된 중간결과들을 저장하는것이 유용하다. 즉 입력된 작업흐름을 가지고 실체화된 마디들을 가장 적합한 자료형식으로 선택하여 DIW에 되돌려준다.(그림 1)

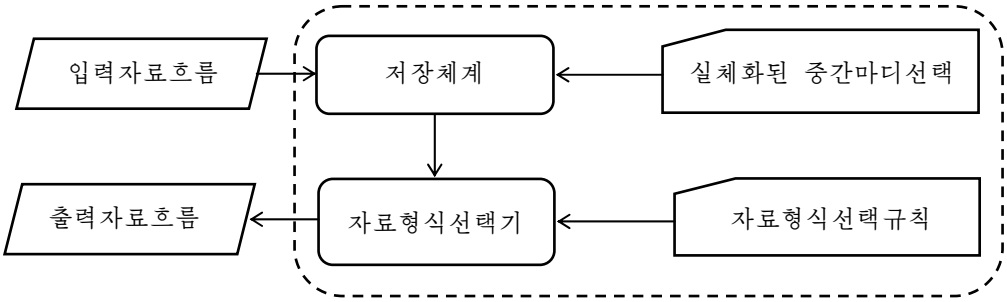

그림 1. 실체화된 중간결과저장기

Pig는 Hadoop클라스터상에서 실행되는 DIW들을 위한 가장 대중적인 언어이므로 실체화된 중간결과저장기와의 호환성문제를 해결할수 있다.

론문에서 제안한 탐색규칙은 오직 실체화된 마디들을 읽어내는 련속적인 스크립트들의 첫번째 연산자만 고찰한다. 그것은 첫번째 연산자가 디스크로부터 자료를 읽어내는데 효과적이고 련속적인 연산자들은 기억으로부터 자료를 읽어내는데 효과적이기때문이다. 자료형식이 결정된 다음 선택된 자료형식으로 직렬전송된다.

직렬과정은 Avro와 Parquet에서 복잡하지 않다. 그것은 Avro와 Parquet가 직렬 혹은 비직렬과정에 투플들의 스키마를 자동적으로 추측하기때문이다.

SequenceFile은 직렬 혹은 비직렬화를 위해 열쇠－값형태의 자료를 요구한다.

그러므로 론문에서 제안한 체계는 개개의 투플을 열쇠－값형태의 자료(하나의 속성은 열쇠, 다른것은 값)로 자동적으로 변환한다.

3. 실험 및 결과분석

론문에서는 자료묶음과 질문을 만드는 표준TPC-H도구를 리용하며 1～128GB의 자료를 생성하여 분석하였다.

그림 2에 이전의 고정된 자료형식들을 리용할 때와 론문에서 제안한 방법을 리용할 때의 성능을 보여주었다.

론문에서 제안한 방법은 평균 SequenceFile에 비해 32%, Avro에 비해서 19%, Parquet에 비해 4%의 성능개선을 보여주어 전체적으로 18%의 성능을 개선하였다.

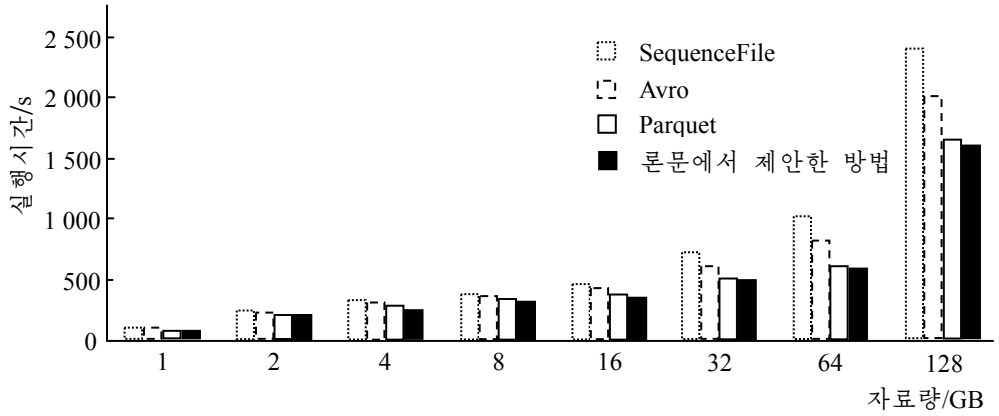

그림 2. 고정된 자료형식을 리용할 때와의 비교

맺 는 말

DIW에서 적당한 자료형식으로 변환할 실체화된 중간결과들을 선택하여 이후의 련속적인 연산에 리용하는 자료형식선택저장방법을 제안하여 전체적인 실행시간을 줄이였다.

참 고 문 헌

[1] Z. Mahmood; Data Science and Big Data Computing, Springer, 191～220, 2016.

[2] C. Lam; Hadoop in Action, Manning, 55～120, 2010.

주체108(2019)년 8월 5일 원고접수

A Method of Data Format Choosing for the Intermediate Results in the Data Intensive Workflows

Ri Kyong Sim, *Ri Il Nam*

In this paper was proposed the method to choose and store the appropriate data format for materializing intermediate results in DIW(Data Intensive Workflows) because subsequent workflows would be used.

Our approach provides on average 32% in speedup over fixed SequenceFile, 19% in speedup over fixed Avro, 4% in speedup over fixed Parquet and overall, it provides 18% in speedup.

Key words: intermediate result, DIW-Data Intensive Workflow