

HDFS에서 소형 파일의 효율적인 접근을 위한 메타데이터 동기화 기반의 분산 캐시 관리 기법

A Distributed Cache Management Scheme Based on Metadata Synchronization for Efficient Accesses of Small Files in HDFS

오현교, 김기연, 황재민, 박준호*, 임종태, 복경수, 유재수
(교신저자)

충북대학교,
국방과학연구소*

Hyunkyo Oh, Kiyeon Kim, Jae-min Hwang, Junho Park*,
Jongtae Lim, Kyoungsoo Bok, Jaesoo Yoo

Chungbuk National Univ,
Agency for Defense Development*

요약

최근 개인 사용자에게 최적화된 정보 및 서비스를 제공하기 위해 다수의 소형파일에 대한 분석이 요구되고 있다. 이런 요구사항을 고려해 본 논문에서는 통신주기에 따라 캐시 메타데이터 동기화를 적용한 분산 캐시 관리 기법을 제안한다. 제안하는 기법은 다수의 소형파일을 병합하여 청크에 저장함으로써, 네임노드가 관리하는 메타데이터의 양을 줄였으며, 클라이언트 캐시에 캐시 메타데이터를 추가적으로 저장하는 구조를 갖는다. 또한, 클라이언트와 네임노드 간 통신주기 조절을 통해 클라이언트의 캐시 메타데이터를 갱신함으로써 불필요한 처리시간을 감소시킨다.

I. 서론

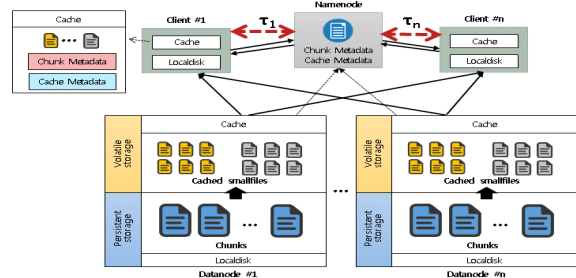
인터넷 서비스 등의 발달로 매우 다양한 형태의 빅 데이터가 일상에서 급속히 생성되어 유통되고 있다. 이러한 상황에서 방대한 데이터를 축적하고 분석하려는 노력이 활발하게 이어지고 있다[1]. 이러한 대용량 데이터를 처리하기 위해 설계된 HDFS는 수십~수백GB이상의 파일들을 64MB의 청크(Chunk)에 나누어 관리한다. 하지만 HDFS에서 소형 파일을 저장하기 위해선 하나의 청크에 수십~수백KB밖에 되지 않는 단일 파일 정보가 관리되기 때문에 저장해야 할 소형 파일의 수가 증가하면 파일을 관리해야 할 청크의 수도 증가함으로써 네임노드 병목현상이 발생한다. 이를 해결하기 위해 소형파일을 HDFS환경에서 효율적으로 저장하고 관리하기 위한 연구[3]가 진행 되었지만, 단일 캐시만을 사용하기 때문에 많은 양의 소형파일 정보들을 유지할 수 없다.

본 논문에서는 HDFS에서 소형 파일의 읽기 성능향상을 위해 통신 주기에 따른 캐시 메타데이터 동기화 정책을 적용한 분산 캐시 관리 기법을 제안한다. 제안하는 기법에서는 청크 메타데이터와 요청한 소형 파일만을 클라이언트의 캐시에서 유지하던 기법[2]과 달리 캐시 메타데

이터를 추가적으로 저장하는 구조를 갖으며, 데이터 노드의 캐시도 함께 사용한다. 또한, 각 클라이언트가 유기적으로 통신주기(τ)를 조절해 네임노드의 최신 업데이트된 캐시 메타데이터를 클라이언트의 캐시 메타데이터와 동기화 시켜줌으로써 클라이언트의 캐시 메타데이터가 최신 정보를 유지하도록 한다. 이를 통해 데이터 노드의 디스크에 접근하는 횟수를 크게 감소시켜 소형 파일의 처리 속도를 향상시킨다.

II. 제안하는 분산 캐시 관리 기법

본 논문에서는 HDFS상에서 소형 파일의 처리 속도를 향상시키기 위해 클라이언트의 캐시 메타데이터 동기화를 적용한 분산 캐시 관리 기법을 제안한다. 제안하는 기법은 각 클라이언트들이 네임노드와 통신주기(τ)를 유기적으로 조절하며 캐시 메타데이터를 업데이트함으로써 소형파일 읽기 처리속도를 향상 시켰다. 그림 1은 제안하는 캐시 관리 기법의 구조를 나타내며, 데이터 노드의 캐시와 이에 접근하는 클라이언트로 구성된다.

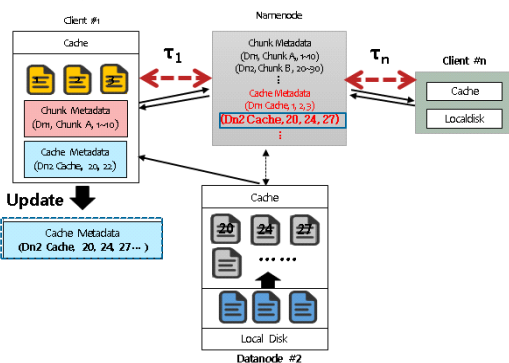


▶▶ 그림 1. 제안하는 캐시 관리 기법의 구조

* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업(14-824-09-001, 실시간 대규모 영상 데이터 이해·예측을 위한 고성능 비주얼 디스커버리 플랫폼 개발), 미래창조과학부 및 정보통신산업진흥원의 대학IT연구센터육성 지원사업/IT융합고급인력과정지원사업(NIPA-2014-H0301-14-1022)과 교육부와 한국연구재단의 지역혁신인력양성사업(No.2013H1B8A2032298)으로 수행된 연구결과임.

각 클라이언트들이 소형파일을 요청함에 따라 데이터 노드의 캐시에 유지되는 소형파일들이 변화하게 된다. 이에 따라 네임노드에서 관리되는 데이터 노드 캐시의 캐시 메타데이터 정보도 변하게 된다. 이때, 클라이언트의 캐시에 유지된 캐시 메타데이터 정보와 네임노드의 캐시 메타데이터 정보가 다르게 되면, 소형파일 요청 시 잘못된 접근을 하게 될 확률이 높아진다. 따라서 각 클라이언트마다 주기적으로 통신주기(τ)를 조절하여 클라이언트의 캐시 메타데이터 정보를 업데이트함으로써 잘못된 접근 확률을 낮춰 빠른 읽기 처리를 수행한다.

그림 2는 클라이언트 캐시의 메타데이터 갱신을 통한 캐시 변화를 나타낸 것이다. 클라이언트 #1은 주기적인 통신주기(τ)마다 네임노드와 통신을 하여 자신의 캐시 메타데이터 정보를 갱신하며, 네임노드로부터 자신의 캐시에 유지하고 있는 캐시 메타데이터의 정보만을 받아온다. 이때, 클라이언트의 캐시 메타데이터 업데이트를 위한 네임노드와의 통신주기(τ)는 단위 시간당 클라이언트 캐시에 업데이트 되는 메타데이터의 업데이트율에 따라 각 클라이언트마다 다르게 적용된다. 만약, 해당 클라이언트의 업데이트율이 임계치를 넘으면 해당 클라이언트는 네임노드와의 통신 주기를 낮춘다. 이유는 기존에 클라이언트에서 유지하던 오래된 메타데이터 정보가 최신의 네임노드 캐시 메타데이터 정보로 대거 교체되기 때문이다. 반대로 업데이트율이 임계치보다 낮으면 통신 주기를 높여 네임노드의 메타데이터 정보를 받아 업데이트 한다. 이렇게 각 클라이언트마다 주기적인 통신주기를 통해 캐시 메타데이터를 갱신 하여 네임노드와 동기화를 시켜줌으로써 불필요한 작업 처리 시간이 감소된다.



▶▶ 그림 2. 통신주기에 따른 클라이언트 캐시의 메타데이터 갱신

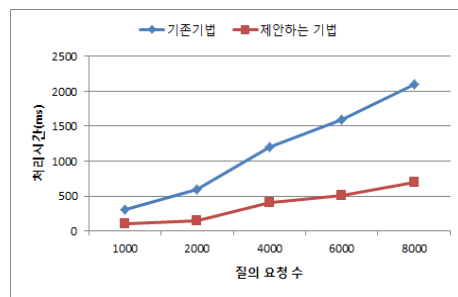
주기적인 캐시 메타데이터 갱신정책을 적용하지 않을 경우, 소형 파일을 요청했을 때, 클라이언트 캐시의 캐시 메타데이터 정보와 네임노드에서 유지하고 있는 캐시 메타데이터 정보가 달라 재요청을 하는 경우가 자주 발생하게 된다. 이에 따라 소형파일 읽기 처리시간을 지연시킨다. 그러므로 각 클라이언트의 업데이트율에 따라 유연한 통신주기 조절을 통한 캐시 메타데이터 업데이트 정책이 필요하다.

III. 성능 평가

본 논문에서 제안하는 기법의 우수성을 보이기 위해, 클라이언트의 캐시를 사용하여 소형 파일 접근을 수행한

기존 기법[2]과의 성능 평가를 수행하였다. 성능 평가는 Inter(R) Core(TM) i5-3470 CPU 프로세서 4G 메모리를 가진 Window 7 운영체제를 사용하는 PC에서 수행하였다. 질의 요청수를 변화해가며 처리 시간과 데이터 접근 빈도수를 측정했다.

그림 3은 소형파일 질의 요청을 1,000에서 8,000까지 변화시켜 처리시간을 측정하였다. 기존 기법[2]의 경우, 캐시 메타데이터를 갖고 있지 않으며, 갱신 정책을 수행하지 않는다. 또한, 클라이언트 캐시 메타데이터만을 사용함으로 캐시에 데이터가 존재하지 않으면 빈번하게 데이터 노드의 청크에 접근하여 파일을 가져온다. 이에 따라, 질의 요청이 많아지면 처리 시간도 증가하는 것을 볼 수 있다. 하지만, 제안하는 기법은 클라이언트 캐시에 캐시 메타데이터를 유지하며, 각 데이터 노드의 캐시를 클라이언트 캐시와 함께 사용한다. 따라서 클라이언트의 캐시에 파일 정보가 존재하지 않아도 캐시 메타데이터를 통해 각 데이터 노드의 캐시에서 유지하고 있는 파일을 요청받아 사용한다. 또한, 주기적 통신주기에 따른 클라이언트 캐시 메타데이터 동기화를 통해 불필요한 접근을 감소시킨다. 이에 따라, 청크의 접근 빈도가 줄어들어 질의 요청이 증가하여도 처리 시간은 많이 증가하지 않는다. 성능평가 결과, 제안하는 분산 캐시 기법이 기존 기법에 비해 처리시간이 최대 약69% 감소했다.



▶▶ 그림 3. 소형 파일 요청에 따른 처리 시간

IV. 결론

본 논문에서는 HDFS에서 소형 파일의 접근 속도를 향상시키기 위해서 유연한 통신주기 조절을 통해 캐시 메타데이터 동기화를 적용한 분산 캐시 관리 기법을 제안하였다. 제안하는 기법은 각 클라이언트가 업데이트율에 따라 네임노드와 주기적으로 통신주기(τ)를 조절하는 캐시 메타데이터 갱신정책을 통해 불필요한 작업처리 시간이 감소된다. 성능평가 결과 제안하는 기법이 기존 기법보다 우수한 성능을 나타내는 것을 확인했다.

■ 참고 문헌 ■

- [1] J. Dittrich and J. Quiané-Ruiz, "Efficient BigData Processing in Hadoop MapReduce," Proc. of VLDB Endowment, Vol.5, No.12, pp.2014-2015, 2012.
- [2] D. Chandrasekar, R. Dakshinamurthy, P. G. Sechakumar, and B. Prabavathy, "A Novel Indexing Scheme for Efficient Handling of Small Files in Hadoop Distributed File System," Proc. of International Conference on Computer Communication and Informatics, pp.1-8, 2013.