

데이터 집약적인 과학 응용을 위한 적응형 데이터배치 기법

김희원⁰ 오유리¹ 최지은² 김서영⁰ 김윤희^{0*}

⁰숙명여자대학교 컴퓨터학과

¹한국특허정보원, ²한국과학기술정보연구원 슈퍼컴퓨팅본부

⁰{gmldnjs0610, yulan}@sookmyung.ac.kr

¹yroh0203@kipi.or.kr ²jieun1205@kisti.re.kr ⁰ssyyy77@gmail.com

An Adaptive Data Placement Strategy for Data-Intensive Scientific Applications

Heewon Kim⁰, Yoori Oh¹, Jieun Choi², Seoyoung Kim⁰, Yoonhee Kim^{0*}

⁰Dept. of Computer Science, Sookmyung Women's University

¹Korea Institute of Patent Information

²Supercomputing, KISTI

요 약

데이터 집약적인 과학 워크플로우 응용 프로그램의 데이터는 대량의 데이터가 여러 데이터센터에 분산되어 있고, 응용을 실행하면서 이를 이용하기 때문에 데이터 위치에 따라 실행결과가 달라진다. 또한 응용 실행을 하면서 생산된 중간 데이터의 저장 위치도 전송을 최소화하기 위해 위치 선정이 중요하다. 이를 위해 데이터 집약적인 응용의 효율적인 수행을 위해 동적 자원의 가용상태를 고려한 적응형 데이터 배치 기법을 제안한다. 실험을 통해 워크플로우가 실행되는 동안 데이터 이동을 효과적으로 줄일 수 있음을 보여준다.

1. 서 론

데이터 집약적인 과학 클라우드 워크플로우 응용 프로그램은 고성능 컴퓨팅 리소스와 함께 대용량의 스토리지가 필요하다[1]. 천문학[2], 항공 유체역학[3]과 같은 분야에서 사용되는 데이터 집약적인 HPC 응용은 대용량의 입력 데이터를 분석하고, 작업 수행 중 생성하는 중간데이터를 처리 및 대용량의 최종 결과 데이터를 산출한다. 과학적 워크플로우의 응용 프로그램 데이터는 대량의 데이터가 여러 데이터 센터에 분산되어 있고, 응용을 실행하면서 이를 이용하기 때문에 데이터 위치에 따라 실행결과가 달라진다. 따라서 HPC 자원뿐만 아니라 데이터들을 저장하는 위치에 대한 고려가 필요하다. 또한 응용 실행을 하면서 생산된 중간 데이터의 전송을 위해 서로 다른 데이터센터 간의 접근이 필요하다. 그러므로 데이터 집약적인 응용의 효율적인 수행을 위해 데이터의 배치 기법이 중요하다.

최근 데이터 배치 기법 연구는 실험 계획 단계에서 데이터 간의 의존도[4], 데이터 사이즈[5], Bandwidth[6]에 집중되어 있으나 실험 중 자원 변화에 대한 고려는 하지 않고 있다. 그래서 본 논문에서는 데이터 집약적인 특성을 갖는 응용인 워크플로우 응용을 대상으로 실험 중 동적 데이터 상태를 고려한

적응형 데이터 배치 기법을 제안한다.

본 논문의 구성은 다음과 같다. 1장 서론에 이어 2장에서는 관련 연구들을 살펴보고, 3장 본문에서는 적응형 데이터 배치 기법에 대해 설명을 한 후, 4장에서는 본문에 설명한 기법에 대한 실험을 진행한다. 마지막으로 5장에서는 결론 및 향후 연구에 대해서 설명한다.

2. 관련 연구

Dong Yuan[4]는 의존도(dependency)에 따라 데이터들을 배치한 행렬 기반 k-means 클러스터링 전략을 사용해 데이터 배치 전략을 제안하였다. Qing Zhao[5]는 데이터 크기를 고려한 연구이고, 고정된 위치의 데이터셋을 고려하여 데이터 레이아웃을 위한 오프라인 전략과 온라인 전략을 제안하였다. Qiang Xu[7]는 유전 알고리즘(Genetic algorithm)을 사용하여 데이터들을 배치하고 데이터센터간의 데이터 스케줄링을 최소화한다.

위 논문들의 연구에서는 클라우드 워크플로우 응용 프로그램에서 데이터의 이동을 효과적으로 줄일 수 있지만, 실행 중간에 변화된 데이터 상태에 대해서는 고려하지 않았다. 이에 본 논문에서는 실행 도중 가용한 자원의 상태가 변하였을 때, 자원의 상태에 따라 데이터들을 다시 적절하게 재배포하는 기법을 제안한다.

* 교신저자

3. 본 론

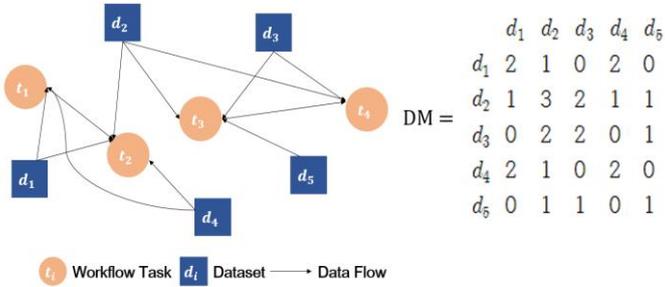
본 장에서는 데이터를 데이터센터에 배치할 때, 데이터들 간에 의존도를 고려한 논문[4]을 기반으로 실험 중 동적으로 변하는 데이터 상태를 고려한 적응형 데이터 배치 기법을 제안한다.

3.1 데이터 배치의 기본 전략

데이터센터 간의 데이터 이동에는 많은 시간이 소요될 수 있다. 데이터 센터에 데이터들을 랜덤하게 배치하는 것보다 정적으로 적용되는 알고리즘을 사용하여 배치한다[4]. 이 알고리즘은 실험 계획 단계에서 데이터센터 간의 데이터들의 이동을 줄이기 위해 데이터들간의 의존도를 고려하여 배치한다.

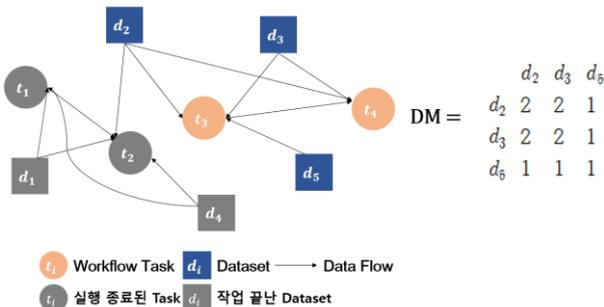
본 논문에서는 실험 실행 중 동적으로 변하는 자원의 상태를 고려하여 데이터 재배치를 수행한다.

3.2 의존도에 따른 데이터 그룹핑 및 작업 재배치



[그림1] 워크플로우 [그림2] 그림1의 DM

그림1은 데이터셋들과 워크플로우 태스크들간의 단순한 워크플로우의 예를 보여준다. 그림1에 나온 데이터셋들과 워크플로우 태스크들 간의 의존도를 계산하여, 그에 따라 의존도 행렬(Dependency Matrix)인 DM을 만들었다. 그림 2는 논문[4]의 알고리즘을 기반으로 그림 1의 의존도를 계산하여 생성된 DM(Dependency Matrix)을 보여준다. DM은 $n \times n$ 의 대칭형 행렬이며, n 은 데이터셋을 의미한다. 행렬의 값은 데이터셋들과 워크플로우 태스크들 간의 의존도를 계산한 값이다. 다음으로 우리는 BEA(Bond Energy Algorithm)[8]을 사용하여 의존도 행렬인 DM을 변형시켜 모든 데이터셋들을 의존도에 따라 그룹핑 시킨 후 데이터 센터에 배치하였다.



[그림3] 실행 도중의 워크플로우 [그림4] 그림3의 DM

본 논문에서는 작업이 어느 정도 실행 되었을 때, 데이터들 간의 상태가 변화한다고 가정하였다. 그림 4는 논문[4]를 기반으로 작업 실행 도중의 데이터셋들과 워크플로우 태스크들 간의 상태 변화의 모습을 보여준다. 그림 4는 그림 3에 따라 다시 데이터셋들과 워크플로우 태스크들 간의 의존도를 계산한 후의 의존도 행렬인 DM을 나타낸다. 다음으로 DM을 BEA[8]을 사용하여 변형시켜, 다시 데이터셋들을 의존도에 따라 그룹핑을 시켜 데이터센터에 배치한다. 그림 2와 그림 4에 나타난 행렬을 살펴보면 데이터셋들과 워크플로우 태스크들 간의 의존도가 변하였음을 알 수 있다.

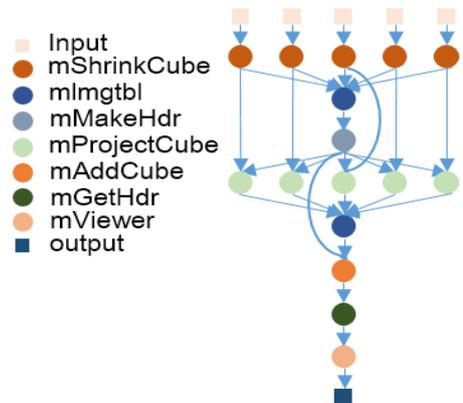
데이터셋들과 워크플로우 태스크들 간의 의존도는 동적으로 변하기 때문에 데이터들을 기존에 배치한대로 실행하는 것보다 작업의 실행 도중 데이터들 간의 의존도를 고려하여 데이터를 재배치 시키는 것이 더 효율적이다.

4. 실험 및 결과

본 장에서는 앞 장에서 언급한 실험 중 동적으로 변하는 데이터 상태를 고려하여 데이터를 재배치 하였을 때, 효율적이라는 것을 증명하기 위해 데이터 재배치의 전과 후의 성능 차이를 비교하였다. 이를 위해 실험을 진행하고 그 결과를 분석했다. 대상 응용의 특성에 대해 설명하고 실험을 진행한 환경에 대해 설명한 후, 마지막으로 실험에 대해 설명하고 결과를 분석한다.

4.1 대상 응용 특성

제안하는 기법의 우수성을 확인하기 위하여 워크플로우 특성을 가진 천문학 분야의 Mosaic 이미지 생성 엔진(Montage GALFA)[9]을 대상으로 실험을 수행하였다. Montage GALFA 응용 프로그램은 대용량의 input 데이터를 사용하여 많은 수의 중간데이터와 하나의 output 데이터를 생성한다. 그림5는 Montage GALFA 워크플로우를 나타낸다.



[그림5] Montage GALFA 워크플로우
Montage의 응용 특성은 메모리 집약적이며 자원의

메모리에 따라 실행에 영향을 받는다. 본 실험에서는 mShrinkCube 단계에서 15 planes를 사용하여 실험을 수행하였다.

4.2 실험 환경

본 실험에서는 사실 클라우드인 OpenStack[10]을 사용하여 표1과 같은 자원을 사용하였다. 표1은 실험에 사용된 자원의 정보를 보여준다.

<표1> 실험 자원 환경

Name	vCPU	Mem(GB)	Disk(GB)
m1.medium	2	4	40

4.3 실험 전략

가용한 자원의 상태가 변할 경우, 자원의 상태를 고려하여 데이터를 재배포하기 전(실험 1)과 후(실험 2)의 실행시간을 비교하기 위해 오픈스택의 2개의 자원(m1.medium)에서 Montage 응용의 실행시간을 측정하였다.

실험 1은 데이터를 각각의 2개의 데이터센터에 배치하였다. 실험 2는 가용한 자원의 상태가 변했다고 가정하고, 데이터 이동이 많은 mProjectCube 모듈에서 데이터들을 한 개의 데이터센터에 재배포하여 mMakeHdr 모듈에서 mlmgmt 모듈로 데이터센터 간 이동하는 횟수를 줄였다.

4.4 실험 결과

mMakeHdr 모듈에서 실험 1과 실험 2의 데이터가 같은 mProjectCube 모듈로 갈 때, 실험2는 실험 1에 비해 시간이 평균적으로 42.5% 감소했다. mProjectCube 모듈에서 두 실험의 데이터가 같은 mlmgmt 모듈로 갈 때, 실험2는 실험 1에 비해 시간이 평균적으로 11.8% 감소했다. mMakeHdr 모듈에서 mlmgmt 모듈로 데이터들이 이동하는데 총 걸리는 시간은 실험 1과 실험 2에서 각각 20분 32초, 16분 20초가 소요되었다.

본 논문에서 제안한 동적으로 변하는 데이터의 상태를 고려한 데이터 재배포를 반영하면 응용 수행 시간이 달라진다. 따라서 본 논문에서는 데이터 배치가 매우 중요한 요소임을 알 수 있다.

5. 결론 및 향후 연구

본 논문에서는 데이터 집약적인 과학응용을 위한 적응형 데이터배치 기법을 제안하였다. 작업을 실행하면서 데이터들 간의 의존도가 변한다고 가정하였다. 데이터센터 간 데이터 이동에 많은 시간이 소요되는 것을 방지하기 위해 데이터들의 상태를 고려해 의존도를 다시 계산하여 데이터센터에 재배포하였다. 실험 결과, 데이터 재배포를 통해 태스크의 수행시간이 줄어들어 본 논문에서 제안한

기법이 효율적임을 알 수 있다.

향후 클라우드 자원 환경을 추가해 다양한 응용에 이 기법을 적용하여 본 논문을 확장하고자 한다.

Acknowledgement

이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(2015M3C4 A7065646)

참고문헌

[1] E. Deelman, A. Chervenak, Data management challenges of data-intensive scientific workflows, in: IEEE International Symposium on Cluster Computing and the Grid, pp. 687-692.(2008)

[2] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M.-H. Su, K. Vahi, M. Livny, Pegasus: Mapping scientific workflows onto the grid, in: European Across Grids Conference, pp. 11-20.(2004)

[3] CFD, www.cfd-online.com, viewed 2009

[4] D.Yaun, Y.Yang, X.Liu and J.Chen, A data placement strategy in scientific cloud workflows, J.Future Generation Computer Systems, 26, 8(2010)

[5] Q.Zhao, Xiong, C., Zhang, K., Yue, Y., & Yang, J. A Data Placement Algorithm for Data Intensive Applications in Cloud. International Journal of Grid and Distributed Computing, 9(2), 145-156 (2016)

[6] S. Agarwal et al., Volley: Automated Data Placement for Geo-Distributed Cloud Services, Proc. 7th USENIX Conf. Networked Sys. Design and Implementation (2010)

[7] Q. Xu, Z. Xu, and T. Wang, A Data-Placement Strategy Based on Genetic Algorithm in Cloud Computing, International Journal of Intelligence Science 5 (2015)

[8] W.T. McCormick, P.J. Schweitzer, T.W. White, Problem decomposition and data reorganization by a clustering technique, Operations Research, 20, pp. 993-1009 (1972)

[9] 천문학 분야의 Mosaic 이미지 생성 엔진(Montage GALFA), <http://montage.ipac.caltech.edu/docs/cubemosaicstutorial.html>

[10] OpenStack, <https://www.openstack.org/>