그리드 컴퓨팅 환경에서의 작업 마이그레이션의 설계 및 구현

김영균° 1) 조금원2) 송영덕2) 고순흠3) 나정수2) 오길호1)
1) 금오공과대학교 컴퓨터공학부
2) 한국과학기술정보연구원 슈퍼컴퓨팅센터 슈퍼컴퓨팅 응용실
3) 서울대학교 기계항공공학부
ygkim@cespc1.kumoh.ac.kr

Design and Implementation of Job Migration on a Grid Computing Environment

Young-Gyun Kim° 1) Kum Won Cho2) Young-Duk Song2) Soon-Heum Go3) Jeong-su Na2)
Gil-Ho Oh1)
1) School of Computer Engineering, Kumoh National Institute of Technology
2) Supercomputing Application Technology Department, Supercomputing Center,
Korea Institute of Science and Technology Information
3) School of Mechanical and Aerospace Engineering, Seoul National University

요 약

본 논문에서는 Cactus와 Globus를 사용하는 그리드 컴퓨팅 환경에서 작업 마이그레이션(Job Migration)에 대해 연구 하였다. 그리드 컴퓨팅은 고속의 네트워크로 연결된 다중의 사이트에 분산되어 있는 연산 자원들을 활용하는 것으로서, 연산 자원들의 효율적인 이용이 중요하다. 연산자원의 효율적인 이용의 한 방법으로서 작업 마이그레이션은 이동 에이전트, 부하 균등화, 결함 허용 등을 위해 사용될 수 있다. 본 논문에는 한 사이트에서 실행중인 연산 작업이 중단된 경우, 유휴한 다른 사이트의 연산자원으로 이동한 후 체크포인팅 파일을 이용하여 중단된 지점부터 복구하여 연산을 계속 수행하도록 하는 연구를 수행하였다. K*Grid 환경에서 연산시간을 효과적으로 단축함을 실험으로 확인하였다. 보다 동적인 그리드 컴퓨팅에서 결함허용, 연산자원의 효율적인 이용 방법으로 사용될 수 있다.

1.서론

최근 Grid Computing에 관한 많은 연구들이 진행되고 있다. 그리드 컴퓨팅은 문제들을 해결하기 위해 고속의 네트워크로 연결된 연산자원(Computational resources)들을 활용하는 것이다. 다양한 컴퓨팅 자원들의 집합을 연결하는 네트워크 기술은 고속의 ATM으로부터 무선 또는 모뎀 접속 등의 어떤 것도 가능하다. 이들 연결된 자원들을 활용하는 것은 단일의 컴퓨터로는 불가능했던 대규모의 시뮬레이션을 가능하게 하고 지리적으로 분산된 협동작업(Collaborations)의 연산을 지원하며 자원들의 원격 사용을 단순하게 한다[1]. 다중의 사이트에 분포되어 있는 연산 자원들을 사용하는 그리드 컴퓨팅에서는 연산 자원들을 효율적으로 사용하는 것은 중요하다. 그러한 방법들의 하나로서 하나의 컴퓨터로부터 다른 컴퓨터로 작업을 마이그레이션 하는 것은 매우 유용하다. 예를 들어서, 작업의 마이그레이션은 이동 에이전트(Mobile agent), 부하 균등화(Load balancing), 주기적으로 마이그레이션 작업들을 정적인 기억장치로 기록 함으로서 결함 허용(Fault-tolerance)을 구현하기 위해 사용되어 질 수 있다[2].

2. 관련연구

2.1 그리드 컴퓨팅

그리드(Grid)는 몇몇 단체나 가상의 조직(Virtual Organiation)의 요구를 서비스하기 위해 유지되는 컴퓨터와 기억장치 자원들의 분산된 집합이다[3]. 그리드 컴퓨팅환경에서 작업 마이그레이션은 분산된 자원들을 효율적으로 사용하고, 결함 허용을 위해 중요하다. 예를 들어서, 작업이 시작될 때 가장 적합한 자원이 사용 중이고 몇 시간 동안 사용할 수 없다면 대안으로 최적은 아니지만 좀 더 나은 자원이 사용될 수 있을 때 까지 마이그레이션 할 수 있다. 컴파일 시간에 알려지지 않은 필요한 자원들의 양은 동적으로 변화할 수 있다. 더 많은 메모리나 아니면 원격지에 떨어진 컴퓨터의 대규모 데이터 집합의 접근 등은 작업의 실행 시간에 발견 되어질 수 있다[4].
실행중인 작업을 마이그레이션하기 위해 작업의 현재 상태를 정적인 기억장치(하드 디스크)로 기록하는 체크포인팅(Checkpointing) 기법을 흔히 사용한다. 체크 포인팅 기법은 크게 시스템 수준의 체크 포인팅(System-level checkpointing)과 사용자 정의 체크 포인팅(User-defined checkpointing)으로 나누어진다. 시스템 수준의 체크 포인팅은 일정한 간격으로 체크 포인팅을 수행하지만, 사용자 정의 체크 포인팅은 프로그래머가 정의한 논리적인 상태에서 응용프로그램의 상태를 저장한다. 따라서 사용자에게 아주 높은 유연성을 제공한다. 이러한 이유로 사용자 수준의 체크 포인팅은 후처리(Post-processing) 및 시각화(Visualization)와 같은 다른 용도들을 위해 사용될 수 있다[6]. 동적인 그리드 컴퓨팅(Dynamic Grid Computing)환경에서 Cactus의 연산 작업을 이주시키는 많은 연구들이 선행되었다. 대표적인 연구로서 Garbrielle Allen은 Cactus의 연산

작업을 응용 프로그램 수준의 체크 포인팅 기법을 적용하여 글로버스 툴킷 기반의 그리드 환경에서 Cactus응용 프로그램의 마이그레이션을 시도하였다[5]. Cartsten Ernemann은 계산 그리드(Computational Grid)상에서 단일 작업의 자원 소모를 다중 사이트 스케쥴링에 사용하는 적응형 다중 사이트 스케쥴링(Adaptive Multi-site Scheduling)기법을 제안 하였다[7].

표 1 K*Grid 중 실험에 사용한 Venus, newcluster 시스템

Organization		KISTI	KonKuk Univ.
Model		Venus	Newcluster
Architecture		Linux Cluster	Linux Cluster
OS		Redhat Linux 7.3	Redhat Linux 7.3
CPU	CPU	Intel Pentium® IV	Intel Pentium® IV
	Clock	2.0GHz	2.0GHz
	#CPU/Node	1	2
	#Node	63	7
	Total	63	14
RAM	#RAM/Node	512MB	1GB
	Total	31.5GB	7GB
Hard Disk	#Hard Disk/Node	40GB	16GB
	Total	40GB+ 500GB(nfs)	16GB
Network	Login node	venus	newcluster
	Host name	ve001~ve063	node2~node8
	Domain name	gridcenter.or.kr	konkuk.ac.kr
	Interface	Fast Ethernet (100Mbps)	Gigabit Ethernet (1Gbps)

2.2 Globus와 MPICH-G2

글로버스는 분산된 자원들간의 통신, 자원의 위치, 자원의 스케쥴링, 인증과 데이터 접근 등에 있어서 기본적인 능력과 인터페이스를 제공하는 메타 컴퓨팅 기반 툴킷(metacomputing infrastructure toolkits)이다[8].

MPICH-G2는 MPI(Message Passing Interface)의 그리드 컴퓨팅이 가능하도록 한 구현으로서 사용자가 병렬 컴퓨터 상에서 사용되는 동일한 명령어를 사용하여 동일하거나 다른 사이트에 위치한 다중 컴퓨터(Multiple Computers)들 간에 MPI 프로그램들을 실행할 수 있도록 한다. MPICH-G2는 글로버스 툴킷 서비스를 사용해서 인증, 허가, 프로세스의 생성, 모니터링, 제어, 통신, 표준 입출력 재지정, 원격 파일 접근 등의 목적을 위해 글로버스 툴킷 서비스를 사용함으로써 이질성을 숨긴다. 결과적으로 사용자는 다른 사이트에 있는 다중의 컴퓨터들을 통해 병렬 컴퓨터상에서 사용되는 동일한 명령어들을 사용해서 MPI프로그램들을 실행할 수 있다[9].

2.3 Cactus의 특성 및 구조

특정 문제 해석을 위해 필요한 모든 계산적 편의를 제공하는 컴퓨터 시스템을 문제 풀이 환경(PSE : Problem Solving Environment)이라 지칭하며, Nimrod/G[11], Triana[12] 및 Cactus[9,10] 등이 현재 개발되어 있다. 이 중 Cactus는 기본적으로 천체 물리학 연구자들의 공동 연구를 위한 기반으로서 개발 되었으나, 천체 물리학 연구뿐만 아니라 전산유체역학 분야(CFD)에서도 활용 가능하다. Cactus는 응용 과학자들이 일반적으로 활용하는 프로그래밍 언어 Fortran 77/Fortran 90, C, C++등의 프로그래밍 언어를 지원하므로 수치해석을 위해 새로운 프로그래밍 언어를 배울 필요가 없다는 장점과, 컴퓨터구조와 운영시스템에 영향을 받지 않고 실행 가능 하다는 점, Cactus 프레임웍이 자동 병렬화를 담당하므로 사용자가 해석자의 병렬화를 위한 노력을 기울일 필요가 없다는 점, 객체 지향적인 문제 풀이 환경을 제공하므로 다분야 연구자들이 공동으로 수치해석 코드를 개발 및 적용할 수 있다는 특징을 가진다. 또한 글로버스 툴킷, HDF5 I/O, PETSc 과학 라이브러리, 가시화 도구 등을 연결하여 활용할 수 있다.

3. 작업 마이그레이션

3.1 시스템의 구성

실험에 사용한 시스템은 표1과 같은 사양을 갖는 클러스터 시스템으로서 K*Grid 중 대전에 위치한 KISTI의 Venus Cluster와 서울에 위치한 건국대의 newcluster 시스템으로서 대전 KISTI에 위치한 K*Grid Gateway와 Venus Cluster는 0.1Km이내의 거리에 위치하고 건국대의 newcluster시스템과는 약140Km 거리에 위치한다.

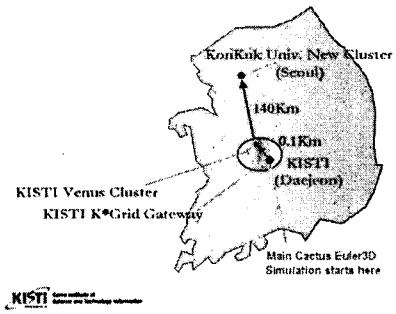

그림 1 실험에 사용한 K*Grid의 지리적 위치

3.2 Cactus기반의 CFD해석과 3차원 Euler방정식

실험에 사용한 연산 작업은 3차원 압축성 Euler방정식이 사용되었다. 공간 차분법으로는 AUSMPW+(modified Advection Upstream Splitting Method Pressure-based Weight function)를 사용하였다.

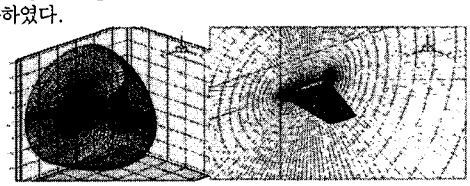

그림 2 실험에 사용한 3차원 Onera-M6 날개 해석 격자계(O-type)

시간적분은 LU-SGS(Lower-Upper Symmetric Gauss Seidel)기법이 사용되었다. 시간 간격은 각 셀에서 선형화 행렬의 고유 값이 CFL안정 조건을 만족시키는 국소 시간 간격(local time stepping)으로 크기를 결정하였다. 해석에 사용된 격자계는 body-fitted 좌표계를 사용하고 단일 블록으로 구성된 3차원 Onera-M6 날개 형상을 활용하여 일반 좌표계를 사용하는 격자 시스템에서의 Cactus를 활용하여 연산을 수행 하였다.

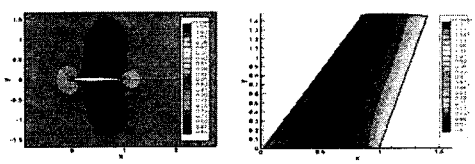

그림 3 압력분포 연산결과(날개 뿌리 단면 및 표면 압력)

4. 구현 및 실험 결과

그림4에서 Migration Server와 Migration Manager는 Perl v5.6.1을 사용하여 Redhat Linux 7.3 운영체제에서 구현 하였다. 실험을 하기 위해 사용한 3차원 Euler방정식은 Fortran77로서 Cactus 4.0 beta 12프레임웍에서 구현 되었다. Migration Server와 K*Grid Gateway는 작업전송의 보안을 위해 SSH의 RSA키를 생성하여 인증과정을 거친다. 또한 K*Grid에 참여하는 모든 연산자원들도 GSISSH의 RSA키를 사용하여 인증한다. 그림 4에서와 같이 사용자는 Migration Server를 통해 K*Grid환경에서 실행될 사용자의 Cactus응용 프로그램을 웹을 통해 제출, 관리하고 K*Grid Gateway에 위치한 Migration Manager를 통해서 연산자원이 있는 적절한 사이트로 작업 마이그레이션을 수행하게 된다. Migration Manager는 Migration Server 또는 다른 사이트의 Migration Manager로부터 전송된 작업을 실행, 중단, 새로운 마이그레이션 사이트 선정(Resource selector), 마이그레이션 여부를 판단(Migration Logic)하여 다른 사이트로 작업을 마이그레이션하거나 작업에 대한 실시간 모니터링 정보를 Migration Server에게 제공하는 역할을 수행 한다.

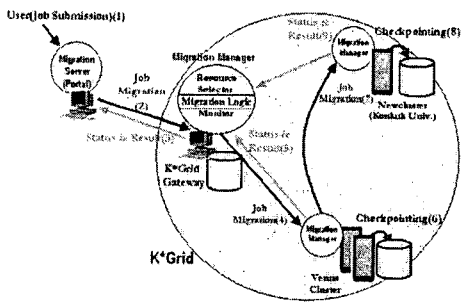

그림 4 K*Grid 환경에서의 체크 포인팅을 이용한 작업 마이그레이션

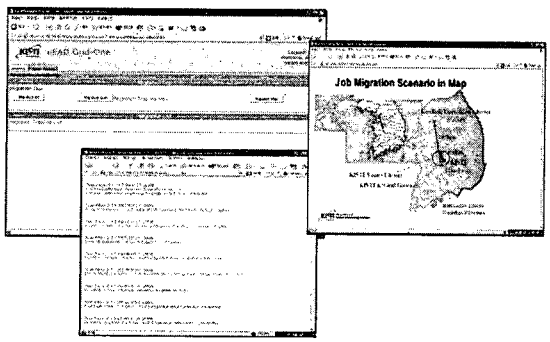

그림 5. 웹을 통한 작업 마이그레이션의 실시간 모니터링

그림 5는 웹으로부터 제출된 Cactus 연산 작업이 K*Grid Gateway를 거쳐, Venus, newcluster로 작업 마이그레이션 과정을 실시간으로 웹을 통해 모니터링한 결과를 보여 주고 있다. 실험에 사용한 3차원 Euler 방정식을 이용한 연산작업은 143×33×65개의 격자를 갖는 것으로서 3차원 Onera-M6 날개 형상의 압력분포를 계산한다. 연산결과는 그림3과 같으며, 작업 파일의 크기는 15.965Mbytes의 크기를 갖는다. 작업의 연산회수는 10,000회로서 Venus와 newcluster 각각 8개의 CPU를 사용하여 연산 작업을 수행하였다. Venus로 작업 마이그레이션을 수행한 결과 715분 48초의 연산 시간을 갖으며 newcluster로 작업 마이그레이션을 수행한 결과 603분 38초의 연산 시간을 갖는다. 따라서, Venus 클러스터 시스템의 연산자원이 결함 허용, 연산자원의 시간제한, 부하 균등화 등의 이유로 다른 사이트로(newcluster)작업을 마이그레이션 함으로써 보다 효율적인 그리드 컴퓨팅을 수행할 수 있다.

5. 결론 및 향후 연구방향

본 연구에서는 Cactus와 Globus를 사용하는 그리드 컴퓨팅 환경에서 리눅스 기반의 클러스터 컴퓨터들 간의 작업 마이그레이션에 대해 연구하였다. 고속의 네트워크로 연결되어 분산된 연산 자원들의 활용이 동적으로 변화하는 그리드 컴퓨팅 환경에서 연산자원을 보다 효율적으로 사용하고, 결함 허용을 고려 함으로서 작업 시간을 단축할 수 있는 방법으로서 작업 마이그레이션이 효과적인 한 방법임을 보였다. 차후, 보다 대규모의 연산 자원들을 활용한 동적인 그리드 컴퓨팅에 대해 연구해 보겠다.

참고문헌

[1] Luís Fabrício Wanderley Góes, Carlos Augusto Paiva da Silva Martins, "Reconfigurable Gang Scheduling Algorithm", 10th workshop on Job Scheduling Strategies for parallel processing in conjunction with SIGMETRICS 2004, Columbia university, NewYork, NY June 13, 2004.

[2] P. Roe, C. Szyperski, "Transplanting in Gardens: Efficient Heterogeneous Task Migration for Fully Inverted Software Architectures", Proceedings of the Fourth Australasian Computer Architecture Conference, Aukland, New Zealand, January 18-21,1999.

[3] I. Foster and C. Kesselman. Editors. The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, 1999.

[4] Gabrielle Allen, Tom Goodale, Michael Russell, Edward Seidel and John Shalf, "Classifying and enabling grid applications", CONCURRENCY-PRACTICE AND EXPERIENCE, Concurrency: Pract. Exper. 2000; 00:1-7

[5] Garbrielle Allen, David Angulo, Ian Foster, Gerd Lanfermann, Chuang Liu, Thomas Radke, Ed Seidel, John Shalf, "The Cactus Worm: Experiments with Dynamic Resource Discovery and Allocation in a Grid Environment", The International Journal of High-Performance Computing Applications and Supercomputing 15(4), Winter, 2001.

[6] Sriram Krishnan, Dennis Gannon, "Checkpoint and Restart for Distributed Components in XCAT3", In Proceedings of the fifth IEEE/ACM International Workshop on Grid Computing, pages 281~288, Pittsburgh, Pennsylvania, 8 November, 2004.

[7] Carsten Ernemann, Volker Hamscher, Archim Streit, Ramin Yahyapour, "Enhanced Algorithms for Multi-Site Scheduling", In 3rd Int'l Workshop on Grid Computing, pages 219-231, 2002.

[8] I. Foster and Carl Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", International Joural of Supercomputing Applications, 11(2),1997.

[9] Nicholas T. Karonis, Brian Toonen, Ian Foster, "MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface", In Proceeding of ASCM/IEEE SC'98 Conference, ACM press, 1998.

[10] Cactus, An open source problem solving environment, http://www.cactuscode.org

[11] D. Abramson, K. power, L.Kolter, "High performance parametric modeling with Nimrod/G: A killer application for the global Grid", In Proceedings of the International Parallel and Distributed Processing Symposium, pp. 520-528, Cancun, Mexico, 2000.

[12] Triana, An open source problem solving environment, http://www.triana.co.kr