

자율 그리드 컴퓨팅을 위한 자기구성 기법

김영균¹⁾ 조금원¹⁾ 나정수¹⁾ 오길호²⁾

1) 한국과학기술정보연구원 슈퍼컴퓨팅센터 슈퍼컴퓨팅 응용실
 2) 금오공과대학교 컴퓨터공학부
 ygkim@kisti.re.kr

Self-Configuring Method for Autonomic Grid Computing

Young-Gyun Kim¹⁾ Kum Won Cho¹⁾ Jeong-su Na¹⁾ Gil-Ho Oh²⁾

1) Supercomputing Application Technology Department, Supercomputing Center, Korea Institute of Science and Technology Information
 2) School of Computer Engineering, Kum-Oh National Institute of Technology

요 약

본 논문에서는 기존의 그리드 컴퓨팅(Grid Computing)에 자율 컴퓨팅(Autonomic Computing)기술을 적용한 자율 그리드 컴퓨팅(Autonomic Grid Computing)에 관하여 연구하였다. 자율 그리드 컴퓨팅 기술 중 자기 구성(Self-Configuration)을 기존의 그리드 컴퓨팅에 적용하는 기법을 새롭게 제안하였다. 장시간 수행 되어야 하는 과학기술 분야의 연산 작업은 연산시간을 단축하기 위해 새로운 노드의 추가와 같은 시스템의 환경 변화를 자동 감지하여 수행중인 연산 작업에 새로운 노드를 추가할 필요가 있다. 제안한 방법은 새롭게 추가된 노드를 연산 작업에 포함하도록 시스템의 변화를 스스로 구성한다. 기존의 그리드 컴퓨팅 보다 시스템의 변화에 보다 잘 적용한다.

1. 서론

최근 그리드 컴퓨팅(Grid Computing)과 자율 컴퓨팅(Autonomic Computing) 및 이들 기술의 결합에 관한 많은 연구들이 진행되고 있다.

Ganek와 Corbi는 그리드의 배치가 많은 시스템들 사이에서 컴퓨팅의 영역을 확장 할 수 있기 때문에 성공적인 그리드 시스템은 자율적인 기능(Autonomic functionality)를 요구한다고 지적하였다[1]. 차세대 그리드 컴퓨팅 기술들은 다음과 같은 기술들로 대표된다[2].

- 자율 컴퓨팅(Autonomic Computing)
- 요구기반 비즈니스와 인프라스트럭처의 가상화
- 서비스 지향 구조와 그리드
- 시맨틱 그리드들(Semantic Grids)

이들 기술 중 본 논문에서는 기존의 그리드 컴퓨팅에 자율 컴퓨팅 기술 중 자기구성(Self-Configuration)기법을 전산유체역학 문제에 적용하는 방법을 새롭게 제안하였다. 자율 컴퓨팅은 컴퓨터 기반 시스템들의 설계에 중요한 전략적 접근방식으로 빠르게 채택되고 있다[3].

2. 관련연구

2.1 자율 컴퓨팅

2001년에 IBM에 의해 연구가 시작된 자율 컴퓨팅은 효과적인 컴퓨팅 시스템들의 설계에 대한 귀중한 새로운 접근 방식으로 대두되고 있다[4].

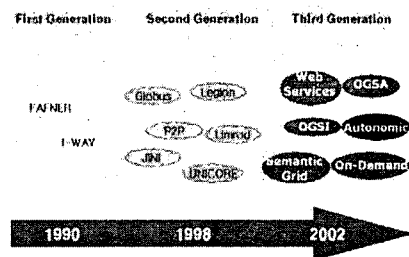


그림 1. 그리드 컴퓨팅 기술의 로드맵

자율 컴퓨팅은 최소한으로 인간의 간섭을 갖는 자기 스스로 관리하는 컴퓨팅 시스템으로서 새로운 컴퓨팅 모델이다[5]. 사용자로부터 복잡성을 숨기고 자기 조절의 전에 없던 수준을 제공한다. 자율 컴퓨팅 시스템들은 다음의 4가지의 문제에 초점을 맞춘 자기 관리(Self-Management) 능력을 필수적으로 갖는다[6].

- 자기 구성(Self-Configuration)
 자율 시스템들은 높은 수준의 정책들에 따라서 시스템 스스로 자동적으로 구성한다.

- 자기 최적화(Self-Optimization)
시스템과 구성 요소들(Components)은 자신의 성능과 효율을 개선하기 위한 기회들을 계속 찾는다.
- 자기 치료(Self-Healing)
자율 시스템들은 지역 소프트웨어와 하드웨어 문제들을 자동적으로 감지, 진단, 수리한다.
- 자기 보호(Self-Protection)
자율 시스템들은 자동적으로 악의적인 공격에 저항하고 실패들(Failures)에 저항한다. 시스템 측면의 실패들을 예측하고 예방하기 위해 초기의 경고를 한다.

2.2 그리드 컴퓨팅

그리드 컴퓨팅은 연산자원과 저장 장치 자원들에 대한 단일의 접근을 제공하며 값싸고 유비쿼터스 분산 컴퓨팅의 기회를 제공한다. 그리드 기술은 e-Government, e-Health와 같은 새로운 영역, 연산자원과 데이터 저장 서비스들과 같은 새로운 사업 기회 그리고 좀 더 큰 조직과 서비스의 변화와 같은 사업 모델들의 변화로 경제의 근본적인 충격을 가져올 것이다 [4,7,8]. 그리드 컴퓨팅은 단일 사이트의 HPC, 클러스터 시스템이 제공하지 못하는 대용량, 고속의 연산 자원들을 제공하여 지구규모의 기상예측, 블랙홀의 충돌, 초신성의 폭발과 같은 천문학적 문제, 우주선, 항공기 등의 전산유체역학(CFD: Computational Fluid Dynamics) 문제, 유전자 분석 등의 계산에 널리 사용이 되고 있다[9,10].

2.3 자율 그리드 컴퓨팅

자율 그리드 컴퓨팅(Autonomic Grid Computing)은 이중의 컴퓨팅 자원들의 집합적인 능력(Collective power)을 단일의 대규모 컴퓨터 시스템으로 이용하고 관리할 수 있도록 하기 위해 그리드 기술들과 자율 컴퓨팅 기술을 결합하는 것이다[6]. 본질적으로 대규모 자원들의 공유와 협력 작업들(Collaborative tasks)을 자기 구성, 자기 최적화, 자기 치료, 자기 보호 방식으로 완수하는 기술이다.

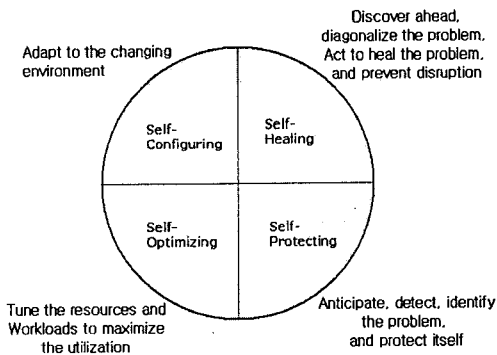


그림 2 자율 컴퓨팅

자율 그리드 컴퓨팅의 목적 중의 하나는 그리드 시스템을 가능한 쉽게 사용하도록 만들고 복잡한 관리 업무로부터 그리드 사용자들을 보호하여 그리드 시스템과 관련된 복잡도(Complexity)를 감소시키는 것이다.

3. 자기 구성 기법

제안한 자율 그리드 컴퓨팅을 위한 자기 구성 기법의 알고리즘은 그림 3과 같다. Step 2에서 노드에 관한 정보를 획득하고

Step 3에서 새로운 노드가 추가되었는지 판단한다. 새로운 노드가 추가되고 새로운 노드가 사용 가능하다면 Step 6을 수행한다. Step 6에서 현재 수행중인 연산 작업의 중간 결과를 체크포인팅하여 저장하고 연산 작업을 중단한다. 새롭게 추가된 노드를 RSL 파일에 포함하도록 RSL 파일을 수정한다. 그 다음으로 수정된 RSL 파일과 체크포인팅 파일로부터 연산이 중단된 지점부터 복구하여 연산 작업을 계속 실행할 수 있도록 연산을 재실행한다.

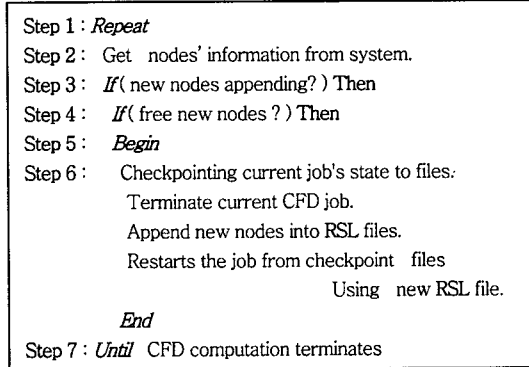


그림 3. 자율 그리드 컴퓨팅을 위한 자기 구성 기법 알고리즘

그림 3에 기술한 자기 구성 기법 알고리즘을 간략히 순서도로 나타내면 그림 4와 같다.

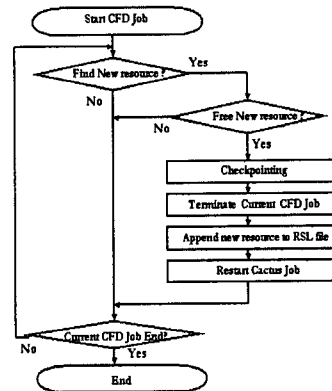


그림 4. 자기 구성 기법 알고리즘의 순서도

제안한 자기 구성 기법은 그림 5에 나타난바와 같이 연산의 초기에는 적은 개수의 노드를 가지고 연산을 수행하다가 새로운 노드의 추가가 발견되는 t라는 시점에서 새롭게 추가된 사용가능한 노드를 현재 작업에 동적으로 추가함으로써 시스템의 노드 변화를 감지하여 연산 작업에 포함하도록 스스로 구성하는 방법을 사용한다. 제안한 방법을 사용하면 연산 작업 중에 추가된 노드를 포함하여 연산을 수행함으로써 장기간 수행되는 과학기술 연산의 연산 시간을 단축할 수 있다.

그림 6에서와 같이 초기의 4노드를 갖는 RSL로 연산작업을 수행하고 새로운 4노드가 추가 된 그림 7의 RSL을 갖고 중단 지점의 체크포인팅으로부터 재시작 한다(Stop/Restart기법).

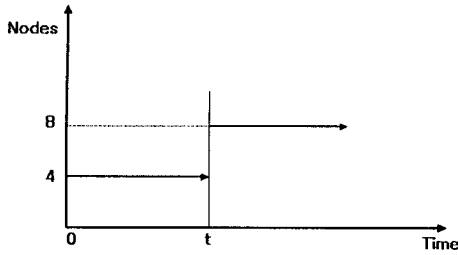


그림 5. 제안한 자기 구성 기법에 의한 노드의 동적 증가

```
+ (&{resourceManagerContact="venus.gridcenter.or.kr")
(count=4) // this selects 4 CPUs
(label="subjob 0")
(environment=(GLOBUS_DUROC_SUBJOB_INDEX 0)
(LD_LIBRARY_PATH /usr/local/grid/GLOBUS/lib/))
(directory="/home/kgrid/kgrid021/")
(executable="/home/kgrid/kgrid021/cactus_m9")
(arguments="Euler3D2.par" )
```

그림 6. 초기의 RSL 파일

```
+ (&{resourceManagerContact="venus.gridcenter.or.kr")
(count=8) // this selects 8 CPUs
(label="subjob 0")
(environment=(GLOBUS_DUROC_SUBJOB_INDEX 0)
(LD_LIBRARY_PATH /usr/local/grid/GLOBUS/lib/))
(directory="/home/kgrid/kgrid021/")
(executable="/home/kgrid/kgrid021/cactus_m9")
(arguments="Euler3D2.par" )
```

그림 7. 새로운 노드를 추가한 후의 자기구성기법에 의해 수정된 RSL 파일

4. 실험 결과 및 평가

전산유체역학의 143×33×65개 격자 크기를 갖는 3차원 Euler 방정식을 10,000회 연산을 수행 하였다. Linux 7.3 운영체제와 인텔 펜티엄 IV 2.0GHz와 512MB의 주기역을 갖는 100Mbps의 Fast Ethernet으로 연결되어 있는 노드가 4개를 사용할때 연산시간이 총1123분이고, 1회 연산에 평균 0.1123분 소요 된다. 노드가 8개를 사용할때 연산시간은 총715분이고, 1회 연산에 평균0.0715분 소요되었다. 따라서 처음 5,000회를 노드 4개로 연산을 수행하고 새로운 노드 4개가 추가되어 나머지 5,000회를 노드 8개로 연산을 수행할 때 연산 시간은 561.5분 + 357.5분=919분 소요되는 것으로 예측 된다. 노드 4개만 수행했을 때보다 204분이 단축되어 18.17%의 연산시간 단축을 한다. 그림 8는 새로운 노드의 추가가 발견된 시점에 따라서 단축될 수 있는 연산 시간과의 관계를 그래프로 보여주고 있다. 새로운 노드가 전체 연산 회수 10,000회의 초기에 발견될수록 단축되는 연산시간은 큰 값을 갖는다. 체크포인팅에 소요되는 시간은 41초이다. 체크 포인팅 시간은

전체 연산시간에 비해 큰 비중을 차지하지 않지만 체크 포인팅 횟수가 늘어나면 전체 연산 소요 시간도 증가한다. 연산에 사용되는 격자의 크기가 클수록 전체 연산시간과 체크 포인팅시간은 증가한다.

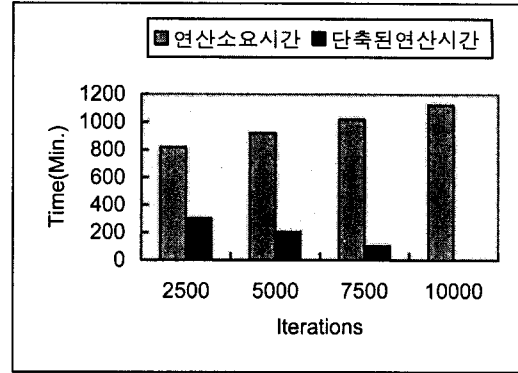


그림 8. 자기 구성 기법에 의한 단축되는 연산 시간

5. 결론 및 향후 연구방향

본 연구에서 제안한 자율 그리드 컴퓨팅을 위한 자기 구성 기법은 인간의 개입 없이 시스템에 추가된 노드들을 연산 작업 중 동적으로 발견하여 현재 연산 작업에 추가 할당하여 연산 작업을 수행 함으로써 장시간 수행 되는 연산 작업의 전체 연산시간을 단축하도록 하고 그리드 컴퓨팅 환경에서의 연산 자원들을 보다 효율적으로 사용하도록 한다. 차후, 보다 대규모의 연산자원과 연산 작업을 사용하도록 하는 추가의 연구가 필요하다.

참고문헌

- [1] Ganek, A.G., Corbi, T.A., "The dawning of the autonomic computing era", IBM Systems Journal 42,5-18(2003).
- [2] Joshy Joseph, Craig Fellenstein, "Grid Computing", PRENTICE HALL PTR, pp. 60-68.
- [3] Roy Sterritt, Dave Bustard, Andrew McCrea, "Autonomic Computing Correlation for Fault Management System Evolution", Proceedings of the IEEE International Conference Industrial Informatics(INDIN 2003), Banff, Alberta, Canada, 21-24, August 2003
- [4] Roy Sterritt, "Pulse Monitoring: Extending the Health-check for the Autonomic GRID", Proceeding of the IEEE Workshop on Autonomic Computing Principles and Architecture(AUCOPA'2003) at IEEE Int. Conf. Industrial Informatics(INDIN 2003), Banff, Alberta, Canada, 22-23 August 2003
- [5] Autonomic Computing , <http://www.research.ibm.com/autonomic>
- [6] Feilong Tang, Minglu Li, Joshua Zhexue Huang, "Real-time transaction processing for autonomic Grid applications", Engineering Applications of Artificial Intelligence 17(2004) 799-807.
- [7] D. De Roure, N. Jennings, N. Shadbolt, "A Future e-Science Infrastructure" aka Research Agenda for the Semantic Grid, EPSRC/DTI Core e-Science Programme, December 2001.
- [8] I. Foster, C. Kesselman, J.M. Nick, S. Tuecke, "The Physiology of the Grid-An Open Grid Service Architecture for Distributed Systems Integration", <http://www.globus.org/research/papers/ogsa.pdf> June 2002.
- [9] Nicholas T. Karonis, Brian Toonen, Ian Foster, "MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface." In Proceedings of ASCM/IEEE SC'98 Conference, ACM Press, 1998.
- [10] Garbielle Allen, Werner Bengler, Thomas Dramlitsch, Tom Goodale, Hans-Christian Hege, Gerd Lanfermann, Andre Merzky, Thomas Radke, Edward Seidel, John Shalf, "Cactus Tools for Grid Application," Cluster Computing, Vol.4(3), pp. 179-188,2001.