

클러스터 시스템 기반 사출 성형 해석 시뮬레이터의 가용도 분석

최창열*, 박기진**, 최권일***, 구분홍***

*아주대학교 정보통신전문대학원

**안양대학교 문리과학대학 컴퓨터학과

***브이엠테크 주식회사

e-mail : clchoi@ajou.ac.kr, kiejin@aycc.anyang.ac.kr, {[pot527](mailto:pot527@kbh90@vmtech.co.kr),
[kbh90](mailto:kbh90@vmtech.co.kr)}@vmtech.co.kr

Availability Analysis of Inject Modeling Analysis Simulator based on Cluster System

Changyeol Choi*, Kiejin Park**, Kwonil Choi***, Bonhung Ku***

*Graduate School of Information and Communication, Ajou University

**Dept. of Computer Engineering, Anyang University

*** VMTech Inc.

요 약

사출 성형 공정은 서로 복잡하게 연관된 수많은 공정 변수들의 조합으로 인해 대량 컴퓨팅 자원을 필요로 하는데, 이를 위한 비용 효율적인 대안이 클러스터 시스템이다. 본 논문에서는 사출 성형 시뮬레이터 개발을 위한 클러스터 시스템의 소프트웨어 구조를 제안하고 각 노드의 상태 정보를 파악하기 위한 모니터링을 수행하여 전체 시스템의 가용도를 향상시킬 수 있는 메커니즘을 개발하고 이에 대한 클러스터 시스템의 가용도를 분석한다.

1. 서론

사출 성형은 기계 공학에서 가장 널리 쓰이는 플라스틱 성형 공정 중의 하나이다. 사출 성형 공정의 컴퓨터 시뮬레이션은 짧은 시간에 저 비용으로 여러 설계 및 재료와 공정 조건의 조합을 시험할 수 있는 수단이며, 이를 통해 설계상의 오류를 실제 시제품을 생산하기 전에 미리 파악할 수 있다. 하지만 서로 복잡하게 연관된 수많은 공정 변수들의 적절한 조합이 성형품의 품질을 결정하는데, 처리 속도 및 메모리 공간 등과 같은 컴퓨팅 자원의 한계로 인해 시뮬레이션을 통해 빠른 시간내에 정확히 예측하기란 매우 어려운 실정이다. 또한, 해석 시간을 줄이기 위해서 고가의

서버를 구입하는 것은 기업 입장에서는 부담으로 작용할 수 있다. 이와 같은 문제를 해결하기 위한 비용 효율적인 대안이 클러스터 시스템을 통해 시뮬레이션을 수행하는 것이다. 하지만 클러스터 시스템은 응용 프로그램에 매우 의존적인 성질을 가지고 있어 사출 성형 시뮬레이터의 요구 사항을 정확히 분석하여 적절한 구성 메커니즘을 개발할 수 있는 환경에 대한 연구가 필수적이다.

따라서 본 논문에서는 사출 성형 시뮬레이터에서 수행되는 작업을 분석하여 독립적으로 실행될 수 있는 컴퍼넌트를 클러스터 시스템의 각 노드에 설치하여 단일 노드에서 순차적인 실행으로 인한 부하를 분산할 수 있는 소프트웨어 구조를 제안한다. 또한 클러스터 시스템을 구성하는 각 노드의 상태를 파악하는 모니터링 기법을 통해 가용도를 향상시킬 수 있는 메커니즘을 제안하고 이에 대한 분석을 수행한다.

본 연구는 2003년도 중소기업청 산학연 공동 기술 개발 컨소시엄 사업 지원으로 수행되었음.

2. 사출 성형 해석 시뮬레이터

복잡한 기하학적 형상과 하중 그리고 재료의 성질을 가지고 있는 문제들에 대하여, 일반적으로 해석적인 수학해를 구하는 것은 불가능한 일이다. 그러므로 적절한 근사해를 구하기 위해서 수치해석에 의존해야 하며 유한요소법(FEM, Finite Element Method)으로 문제를 수식화하면 미분방정식이 아닌 이에 대응하는 연립 대수방정식이 얻어진다[1]. 여기서 물체를 더 작은 요소로 나누고 그 요소들을 두 개 이상의 요소들이 공유하는 점이나, 경계선, 또는 경계면으로 연결된 대동한 시스템으로 만드는 과정을 이산화 혹은 격자분할이라 한다. 그림 1은 격자 분할의 예를 보여주고 있다.

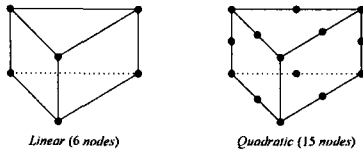


그림 1 격자 분할의 예

유한요소법에서는 전체물체에 대한 문제를 한번 풀어서 보지 않고 각각의 유한요소에 대한 방정식을 세우고, 그들의 조합하여 전체물체에 대한 방정식을 구한다. 간단한 예로는 구조문제에 대한 해는 대개 주어진 하중에 대한 각 절점의 변위와 그 구조물을 구성하고 있는 각 요소 안에서의 응력을 계산하는 것을 뜻한다. 공학에서는 일반적으로 유한요소법을 8 단계로 나누는데 대형 매트릭스 계산을 피하기 위해 본 연구에서는 요소 방정식의 조합을 통한 전체 행렬구성 및 경계조건 적용까지만 유한요소법에서 수행한다. 그림 2는 유한요소법을 적용하여 사출 성형 해석을 수행하는 시뮬레이터의 흐름도에 대한 전체적인 윤곽을 보여준다.

먼저, 속도와 압력을 가정을 하고나서 속도를 풀기 위한 매트릭스를 구성한 후, 경계 조건을 적용하여 속도를 계산한다. 이와 같이 계산된 속도를 사용하여 압력을 구하는데, 이를 위해 매트릭스 생성한다. 또한, 생성된 압력에 대한 매트릭스에 경계 조건을 적용하여 압력을 계산한다. 이후, 얻어진 속도와 압력이 일정값에 수렴되었는지 판단한다. 만약 수렴되었으면, 캐비티(Cavity)가 모두 충전되었는지를 확인하고 충전이 완료될 때 까지 유동은 진전하게 되며 진전할 때 마다 속도와 압력을 새로 계산한다. 하지만 수렴되지 않았으면, 속도와 압력을 위에서 언급한 과정을 다시 반복하여 재계산하게 된다.

현재 개발된 사출 성형 해석 시뮬레이터는 속도와 압력을 각각 순차적으로 계산을 한다. 압력은 속도에 대한 값이 계산되어야만 얻을 수 있으므로 이를 병행적으로 처리하는 것은 불가능하지만 속도에 대한 계산은 각 방향 요소에 따라 계산할 수 있으므로 이를

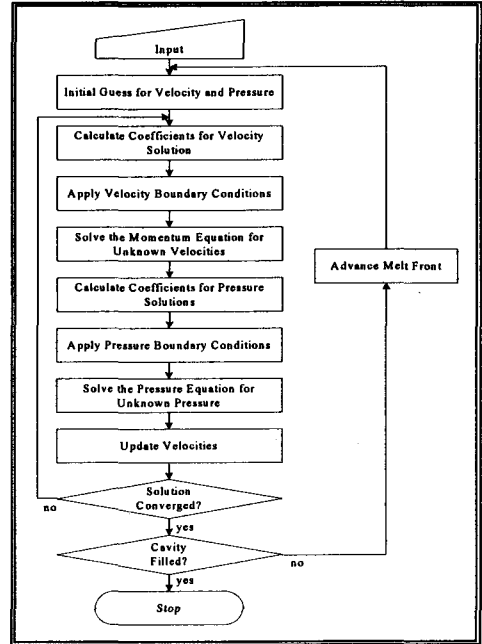


그림 2 사출 성형 해석 시뮬레이터의 흐름도

병행적으로 처리하면, 시뮬레이터의 계산속도를 향상시킬 수 있다. 다시 말해서, 한 개의 주서버에서 초기 값에 대한 분석과 압력에 대한 계산 및 수렴 여부, 캐비티 충전 여부를 파악하고, 3 개의 보조서버에서 속도를 계산하기 위해 필요한 좌표 정보 x, y, z 를 각각 계산할 수 있다. 따라서 이와 같은 최소한의 필요 컴퓨팅 자원을 갖출 수 있는 클러스터 시스템의 구조가 필요하다.

3. 클러스터 시스템 구조

클러스터 시스템의 응용 분야는 크게 고성능 서버, 고가용성 서버, 웹 서버로 나눌수 있는데, 사출 성형 해석 시뮬레이터와 같은 과학 계산을 응용 분야에는 고성능 클러스터(High Performance Cluster)가 적합하다[2,3]. 이를 위해서는 클러스터를 구성하는 모든 컴퓨터들은 네트워크에 연결되어 있어서 상호간에 통신이 가능하도록 하여 다수의 프로세서가 협동적으로 문제를 풀 수 있는 환경을 제공해야한다.

그림 3은 사출 성형 해석 시뮬레이터 구축을 위한 클러스터 시스템의 구조를 나타내며, 크게 관리 노드(Mainframe)와 컴퓨팅 노드(PC₁, PC₂, ... PC_n)로 이루어져 있다. 관리 노드는 성형품에 대해서 유한요소법을 통해 분석한 결과를 매트릭스로 생성하고 대형 매트릭스 분할하여 컴퓨팅 노드에게 분배하는 작업 스케줄링을 수행한다. 또한 컴퓨팅 노드의 분할된 결과를 통합하는 기능도 수행한다. 그리고, 클러스터 자원을 관리하기 위해 컴퓨팅 노드가 정상적으로 작동하는가를

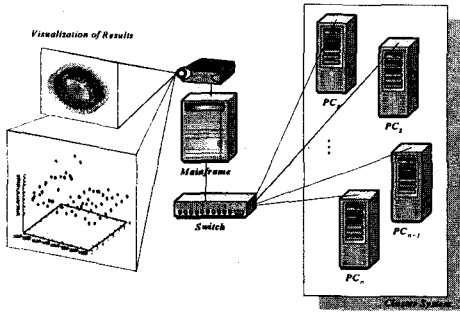


그림 3 클러스터 시스템 구조도

판별하기 위해 모니터링 수행한다. 한편, 컴퓨팅 노드는 해석을 위한 기초 작업을 처리하고 해석 결과를 다시 관리 노드에 전송하여 통합적인 해석 작업을 수행할 수 있도록 한다. 그리고 확장성을 위해 컴퓨팅 노드는 자발적으로 서비스에 참여할 수 있으며, 관리 노드는 서비스 참여 의사를 밝힌 컴퓨팅 노드를 서비스를 제공하기 위한 클러스터 시스템에 포함한다. 마지막으로, 관리 노드는 해석 결과를 가시화하기 위해 Net Juggler 와 같은 병행 시각화 인터페이스(Parallel Visualization Interface)를 사용할 수 있으며, 이는 성형품의 애니메이션이나 그래프 형식으로 나타난다 [4]. 관리 노드에서 컴퓨팅 노드의 가용성(Availability)을 판별하기 위해 모니터링을 수행하는데, 이를 위해 컴퓨팅 노드는 계산 과정에 참여 여부를 관리 노드에게 알리고 나서 자신이 정상적으로 작동하고 있다는 메시지인 "I'm alive with my IP address" 메시지를 전송하게 된다. 관리 노드는 컴퓨팅 노드의 참여 메시지를 받으면 작업 스케줄링 테이블에 등록된 후 해석 작업이 완료될 때까지 컴퓨팅 노드의 상태를 점검하게 된다.

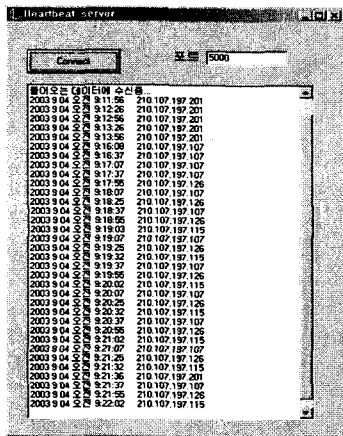


그림 4 모니터링 작업

그림 4 는 관리 노드에서 컴퓨팅 노드의 참여 여부 의사 및 상태 정보에 대한 데이터를 수신하게 되는

과정을 보여준다. 이를 관리 노드의 작업 스케줄링을 위한 테이블의 변화에 따라 표현하면 그림 5 와 같다.

Server_Name	Server_Address	Server_State
Node 1	210.107.197.107	Available
Node 4	210.107.197.201	Available

(a)

Server_Name	Server_Address	Server_State
Node 1	210.107.197.107	Available
Node 3	210.107.197.126	Available
Node 4	210.107.197.201	Not_Available

(b)

Server_Name	Server_Address	Server_State
Node 1	210.107.197.107	Available
Node 2	210.107.197.115	Available
Node 3	210.107.197.126	Available
Node 4	210.107.197.201	Not_Available

(c)

Server_Name	Server_Address	Server_State
Node 1	210.107.197.107	Available
Node 2	210.107.197.115	Available
Node 3	210.107.197.126	Available
Node 4	210.107.197.201	Available

(d)

그림 5 작업 스케줄링 테이블 상태

노드 4 가 참여 의사를 밝히고 서비스를 수행하다가 결함이 발생하여 현재 자신이 정상적으로 작업을 수행하고 있다는 사실을 알리지 못하게 된다. 관리 노드는 일정 시간(Timeout) 동안 컴퓨팅 노드로부터 상태 정보를 받지 못하면 해당 노드의 상태를 서비스 불능(Not_Available) 상태로 전환하고 작업을 할당하지 않는다. 또한 서비스 가능한 컴퓨팅 노드수가 2 이하이면 작업을 할당하지 않는다. 왜냐하면 3 장에서 분석하였듯이 사출 성형 시뮬레이터를 클러스터 시스템에 적용하여 목표로 하는 성능을 제공하기 위해서는 최소 3 대의 컴퓨팅 노드수가 필요하므로 2 대 이하 노드만이 서비스 가능하다면 성능 향상 효과를 얻을 수 없기 때문이다. 하지만 기존 클러스터 시스템에서 정의한 가용도는 한 노드의 결함 발생과 수리 시간에 따라 분석하였기 때문에 이와 같은 성능적인 면을 고려하지 않았으며, 더욱이 단지 컴퓨팅 노드 1 대만이 동작해도 시스템이 가용하다고 분석하였다. 따라서 서비스 가능한 최소 컴퓨팅 노드수가 3 이상일 때만 가용하다는 가용도에 대한 정의가 필요하다.

4. 가용도 분석

클러스터 시스템의 응용 분야 중 고가용성 서버의 가용도는 보조 서버의 이중화에 의한 가용도 향상까지 가용도 분석에 포함을 하고 있는데, 제안된 구조에 필요한 가용도 분석에서는 제공하고자 하는 성능을 만족할 수 없는 노드수로 구성되었을 경우에는 시스템이 가용하지 않다고 정의한다. 그림 6 은 가용도 분석을 위한 마르코프(Markov) 모델이다.

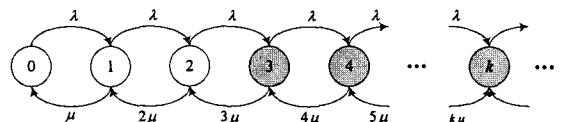


그림 6 사출 성형 해석 시뮬레이터를 위한 클러스터 시스템의 가용도 분석 모델

관리 노드는 결함이 발생하지 않는 서버라 가정하고 컴퓨팅 노드의 참여율(λ), 한 컴퓨팅 노드의 결함 발생율(μ)은 지수 분포(Exponential Distribution)를 따른다고 가정한다. 노드 참여율이란 신규 클러스터 노드의 증가 속도 혹은 참여 이후 결함이 발생한 노드의 수리 속도를 의미한다. 마지막으로 서비스 제공에 참여했던 노드는 결함이 발생한 이후에 수리를 통해 제거가 되면 다시 서비스를 제공할 수 있다고 가정한다. 따라서 그림 6의 마르코프 모델은 사출 성형 시뮬레이터를 위해 필요한 컴퓨팅 자원의 가용도를 나타내는 것으로 한 컴퓨팅 노드가 서비스에 참여하면 사용 가능한 자원은 증가하게 되고 결함이 발생하면 감소하게 되는 상태 전이를 보여준다. 가용도 분석 모델에서 음영으로 칠해진 상태는 클러스터 시스템이 목표로 했던 성능을 제공할 수 있는 상태이며, 음영이 없는 상태는 서비스를 제공할 수 있는 노드는 존재하지만 목표로 한 성능을 제공하지 못하는 상태이거나 서비스를 제공할 노드가 없는 상태이다.

그림 6의 마르코프 모델을 풀기 위해서 균형 방정식(Balance Equation)[5]을 통해 임의의 상태 k 에 머무를 확률을 구하면 아래와 같다.

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} = p_0 \frac{(\lambda/\mu)^k}{k!} \quad (k=1, 2, 3, \dots)$$

또한, 각 상태에 머무를 확률의 총합은 1이라는 법칙(Conservation Equation)[5]을 적용하면 서비스를 제공할 노드가 없는 상태에 머무를 확률을 구할 수 있다 ($p_0 = e^{-\lambda/\mu}$). 따라서 임의의 상태에 머무를 확률은 식 (1)과 같다.

$$p_k = \frac{(\lambda/\mu)^k}{k!} e^{-\lambda/\mu} \quad (1)$$

따라서 목표로 했던 성능을 제공하기 위해서 클러스터 시스템의 가용도의 정의는 아래와 같다.

$$Avail = \sum_{i=1}^{\infty} p_i = 1 - \sum_{i=0}^{\infty} p_i \quad (2)$$

그림 7은 컴퓨팅 노드의 참여율(Participate_Rate)과 한 노드의 결함 발생율(Failure_Rate)에 따른 클러스터 시스템의 가용도 분석 결과이다. 참여를 원하는 컴퓨팅 노드의 수는 무한하다고 가정하였다. 가용도는 결함 발생율이 커질수록 감소하였으며, 참여율이 높을수록 증가하였다. 또한 참여율이 낮을 경우에는 결함 발생율에 따라 가용도 변화가 심하게 발생하였는데, 이는 결함이 발생한 노드를 수리하기 위한 시간 동안 새로 추가된 자원이 없기 때문에 목표로 하는 성능을 제공하기 위해 필요한 최소한의 컴퓨팅 자원을 확보하지 못하기 때문이다. 하지만, 서비스를 제공하려는 노드가 많을수록 결함 발생율이 증가해도 일정 수준의 가용도를 유지하는 것을 볼 수 있다. 그 이유는 결함이 발생한 노드가 수리 중이어도 새로운 자원이 확

보되므로 최소한의 컴퓨팅 자원을 지속적으로 확보할 수 있기 때문이다.

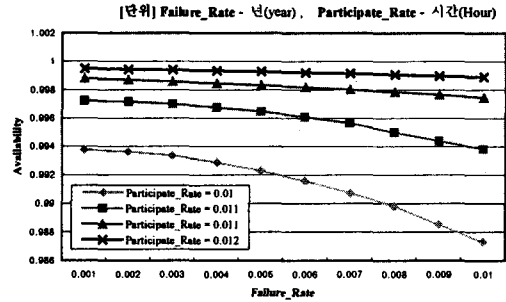


그림 7 사출 성형 해석 시뮬레이터용 클러스터 시스템의 가용도 분석

5. 결론

본 논문에서는 클러스터 시스템의 응용 분야 확장을 위해 사출 성형 해석 시뮬레이터를 위한 시스템의 구조 및 필요한 메커니즘을 제안하였다. 순차적으로 실행되기 때문에 고성능 컴퓨팅 자원을 필요로 하는 사출 성형 시뮬레이터의 문제점을 해결하기 위해 병행적으로 처리될 수 있는 부분에 대해 파악하고 이에 적합한 클러스터 시스템의 구조를 도출하였다. 또한 목표로 했던 성능을 유지하기 위해 컴퓨팅 자원 상태를 지속적으로 감시하는 모니터링 기법을 개발하였으며, 제안한 시스템의 가용도 분석 모델을 제시하고 가용도 분석을 수행하였다. 하지만 사출 성형 해석 시뮬레이터의 종속적인 요소들을 독립적으로 분류할 수 있는 알고리즘을 개발하여 좀더 향상된 성능을 제공할 수 있는 연구가 필요하다.

참고문헌

- [1] R.D. Cook, D.S. Malkus, M.E. Plesha, and, R.J. Witt, *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons, 2002.
- [2] F. Zara, F. Faure, and J-M. Vincent, "Physical Cloth Simulation on a PC Cluster," 4th Eurographics Workshop on Parallel Graphics and Visualization, pp. 105-112, Sep. 2002.
- [3] R. Buyya, *High Performance Cluster Computing: Architectures and Systems*, Prentice-Hall, 1999.
- [4] J. Allard, V. Gouranton, L. Lecointre, E. Melin, and B.Raffin, "Net Juggler: Running VR Juggler with Multiple Displays on a Commodity Component Cluster," IEEE Virtual Reality Conference, pp. 273-274, Mar. 2002.
- [5] L. Kleinrock, *Queueing Systems Volume I: Theory*, Wiley, 1975.