

## RFID 미들웨어에 적합한 부하 분산 기법 연구

박재걸<sup>○</sup>, 임종호, 이광민, 황경수, 채홍석  
부산대학교 컴퓨터공학과

{jgpark<sup>○</sup>, jhlim, leekm, hwangks, hschae}@pusan.ac.kr

### A Study of Load Distribution Approach Optimized for RFID Middleware

Jaegool Park<sup>○</sup>, Jongho Lim, Kwangmin Lee, Kyungsoo Hwang, Heung Seok Chae  
Department of Computer Science and Engineering, Pusan National University

#### 요 약

RFID 시스템은 다양한 분야에 혁신을 제공하는 기술이다. RFID 시스템은 많은 양의 태그 정보 및 데이터에 대한 실시간 처리를 필요로 하는 경우가 많으며, 자동화된 프로세스를 통한 비즈니스 플로우를 지원한다. RFID 미들웨어 시스템의 중추 역할을 하는 RFID 미들웨어에 부하가 집중되는 현상이 지속 된다면 처리의 효율이 낮아짐은 물론 처리 지연으로 인한 비즈니스 플로우의 지연 및 장애를 초래하게 된다. 본 논문에서는 RFID 미들웨어 시스템의 특성을 고려하여 RFID 미들웨어에 적합한 부하 분산 알고리즘을 제안한다. 그리고 일반적인 분산 시스템 환경에서의 부하 관리 메커니즘과 비교할 수 있는 RFID 미들웨어의 부하 관리를 위한 기법을 제시한다.

#### 1. 서 론

부하 분배는 분산되어 있는 자원을 효율적으로 사용하고 높은 성능을 얻기 위해서 자원의 상태와 각 노드의 부하 상태에 따라 자원을 적절히 분배하여 부하를 균등하게 분배 하는 것이다.[1,2,3]

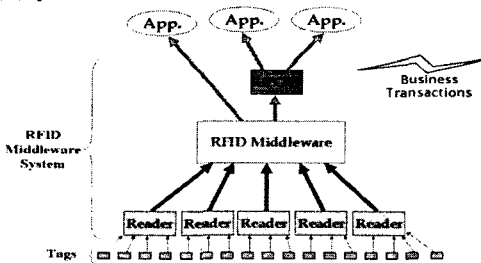


그림 1. RFID 미들웨어 시스템

[그림 1]은 RFID 미들웨어 시스템의 구성을 보여준다. Reader는 태그정보를 읽어 들인다[4]. RFID 미들웨어는 Reader로부터 수집한 정보를 여과하여 의미 있는 정보로 변환한 후 상위 레이어로 전송한다[5]. EPCIS(EPC Information Services)는 사용자가 EPC 네트워크를 통해서 EPC와 관련된 정보를 주고받을 수 있도록 한다[6].

RFID 미들웨어 시스템에서 RFID 미들웨어는 리더 그룹과 상위 레이어 사이의 데이터 교환을 가능하게 하는 중추적인 역할을 한다.[7] RFID 미들웨어에 부하가 집중되는 현상이 지속 된다면 처리의 효율성이 낮아짐은 물론 처리 지연으로 인한 비즈니스 플로우의 지연 및 장애가 발생하게 된다. 이와 같이 부하가 특정 노드에 집중되는 현상을 방지하기 위해서는 부하 분산 기법의 도입이 필요하다.

과부하 문제를 해결하기 위한 방법은 단일 노드의 성능을 향상시키는 것과 클러스터링 기법을 이용하는 것으로 나누어질 수 있다. 단일 노드의 성능을 개선하여 하는 방법을 선택할 경우 성능 개선동안 서비스가 중단되어야함은 물론 요청이 증가되었던 머지않아 다시 과부하가 되어 더 높은 성능의 노드로 다시 개

선해 주어야한다. 이와는 반대로 클러스터링 기법을 이용하면, 확장성과 비용의 효율성 및 신뢰성의 측면에서 큰 이점을 얻을 수 있다.[8] RFID 미들웨어 시스템에서도 [그림 2]와 같이 클러스터링 기법을 이용하여 시스템의 전체적인 성능을 높이고 확장성을 제공하는 시스템을 구축할 수 있다.

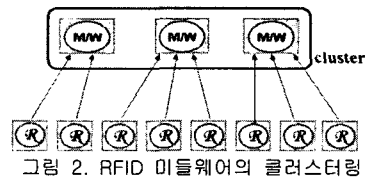


그림 2. RFID 미들웨어의 클러스터링

본 논문은 다음과 같이 구성된다. 2절에서는 분산 시스템에서 부하관리를 위해 고려되어야 할 사항들을 설명한다. 3절에서는 RFID 미들웨어 시스템의 특성을 고려한 부하분산 기법을 제시한다. 마지막으로, 4절에서는 결론 및 향후 연구방향에 대해서 기술한다.

#### 2. 분산 시스템에서의 부하 분산

본 절에서는 분산 시스템에서의 부하 분산 메커니즘에 대하여 소개한다. 부하 분산 메커니즘은 정보 수집 정책, 초기화 정책, 작업 전송 정책, 선택 정책, 위치 선정 정책과 같은 정책들을 포함하고 있다.[9,10]

- 정보 수집 정책(Information Gathering Policy) : 정보 수집 정책은 클러스터 내의 노드들로부터 작업 부하에 대한 정보를 수집하고 관리하는 것을 뜻한다. 정보 수집 정책을 위하여 고려해야 할 것은 부하 관리를 위한 정보 수집을 수행하는 빈도와 정보 수집 방법으로 나누어 볼 수 있다. 정확한 최신 부하 정보를 수집하는 것과 비용을 최소화 시키는 것은 서로 상충된다. 정보 수집 정책은 작업 부하의 추정치와 명세에 대한 정보도 역시 포함하게 된다. 이러한 정보들은 시스템 정보인 CPU 사용률(CPU Utilization), 프로세스 큐 길이(CPU run queue length)과 어플리케이션 정보인 초당 요청 수(Request per Second), 처리량(Throughput), 요청에 대한 응답 시간(Response Time) 등으로 나눌 수 있다.[9,11,12]

이 논문은 교육인적자원부 지방연구중심대학육성사업 (차세대플류IT기술연구사업단)의 지원에 의하여 연구되었음.

- 시작 정책(Initiation Policy) : 시작 정책은 누가 부하 분배 프로세스의 수행을 트리거 할 것인지에 관련된 것이다. Initiator는 근원 노드(source node) 또는 목적 노드(destination node)가 될 수 있으며, 두 가지 모두가 될 수도 있다(symmetric initiations).
  - 중앙 집중식(Centralized Initiation)[13]
 

중앙 집중식 시작 정책은 전체 시스템 내에 부하 분산을 책임지는 하나의 Initiator가 존재한다. 각 노드는 부하 정보를 수집하여 노드에게 제공하고, 노드는 부하 분산 스케줄링을 통하여 부하를 목적 노드에 분산시킨다.
  - 분산 방식(Distributed Initiation)[14]
 

분산 방식 시작 정책은 어떤 노드도 Initiator가 될 수 있다. 각 노드는 자신의 부하에 변화가 있으면 다른 노드들에게 이를 알리게 된다. Initiator는 이렇게 수집된 부하 정보와 부하 정보의 시간 경과를 고려하여 부하를 분산할 목적 노드를 선택하게 된다. 분산 방식에는 송신자-시작(Sender-Initiation) 정책과 수신자-시작(Receiver-Initiation) 정책이 있다[15].
- 작업 전송 정책(Job Transfer Policy) : 작업 전송 정책은 Initiator가 다른 노드들에 작업 부하를 재할당하는 것을 고려해야 하는 시점을 결정한다. 즉, 부하 분산이 필요한 시점이 언제인가를 고려하는 것이다. 이러한 시점에 대한 결정은 노드의 상태만을 기초로 만들어 지기도 하며 서로 교환된 부하 정보를 취합한 노드의 정보를 기초로 하기도 한다.
- 선택 정책(Selection Policy) : 선택 정책은 어떤 작업을 재할당할 것인지 결정하는 것이다. 비 선정 정책에서는 한 번 할당된 작업은 작업이 완료 될 때까지 이동되지 않는다. 하지만 선정 정책에서는 노드에서 수행중인 작업이라도 부하 분배의 대상이 될 수 있다.
- 위치 선정 정책(Location Policy) : 위치 선정 정책은 재할당하기 위해 선정된 작업을 어느 노드에 할당할 것인지를 결정한다. 가장 간단한 위치 선정 정책은 노드를 랜덤 또는 순차적으로 선택하도록 하는 것이다. 보다 복잡한 정책을 사용하게 될 때에는 클러스터내의 각 노드에 대한 다양한 정보들을 고려하여 부하를 분배 받게 될 대상을 결정한다.[9]

분산 시스템에 부하 분산 기능을 제공하기 위해서는 이와 같은 정책들을 필요로 하게 된다. 그리고 이러한 메커니즘은 시스템 설계에 반영되어야 하며 구현되어야 한다.

### 3. RFID 미들웨어 시스템을 위한 부하 분산 기법

본 절에서는 RFID 미들웨어 시스템에 적합한 부하 분산 기법을 제시하기 위하여 기본적인 사항들을 정의한다. 그리고 정의에 따른 부하 분산 알고리즘을 제시한다.

#### 3.1. 기본 전략

RFID 미들웨어 시스템에서는 RFID 미들웨어에 대한 부하의 균등한 분배도 중요하지만 리더의 이동 횟수를 최소화 시키는 것 역시 중요한 목표가 된다. 즉, 리더의 이동 횟수를 최소화 하고 리더 재배치를 위한 시도 횟수를 최소화 시킬 수 있는 알고리즘이 필요하다.

RFID 미들웨어 시스템에서 각 리더의 시간에 따른 작업 부하량은 매우 불규칙적이며 예측하기 힘들다. 뿐만 아니라, 과부하 상태인 RFID 미들웨어에 연결된 리더들을 다른 미들웨어로 재배치하는 방법을 통해 부하를 분산 시킬 수 있다. 특정 RFID 미들웨어에 연결되어있는 리더를 다른 미들웨어로 이동시키기 위해서는 접속을 해제하고, 목적지 주소를 부여 받고 다시 접

속하는 과정을 거치는 부하 분산을 위한 비용이 수반된다. 그러므로 최소의 이동으로 가장 균등한 산포도를 유지할 수 있도록 하는 것이 바람직하다.

#### 3.2. 기본 정의

RFID 미들웨어에 적합한 부하 분산 알고리즘을 기술하기 위하여 표기법을 정의한다.

정의 1. 시스템에 연결된  $n$ 개의 각 RFID 미들웨어(Edge Manager, 이하  $em$ )를  $em_1, em_2, \dots, em_n$  이라고 하면, 시스템에 연결된 전체  $em$ 의 집합  $EM$  은

$$EM = \{em_1, em_2, em_3, \dots, em_n\} \text{ 이다.}$$

정의 2. 시스템에 연결된  $m$ 개의 각 RFID 리더(RFID Reader, 이하  $r$ )를  $r_1, r_2, \dots, r_m$  이라 하면, 시스템에 연결된 전체  $r$ 의 집합  $R$  은

$$R = \{r_1, r_2, \dots, r_m\} \text{ 이다.}$$

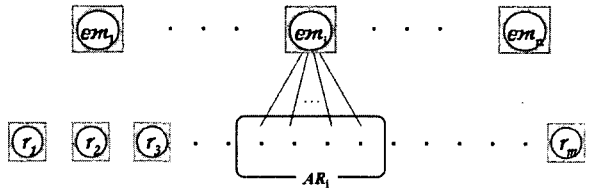


그림 3. 구성요소에 대한 기본 정의

[그림 3]은 정의의 대상이 되는 RFID 미들웨어 시스템에 대한 구성요소와 그와 관련된 관계를 보여준다. 본 논문에서 제안하는 RFID 미들웨어에 적합한 부하 분산 기법은 기본적으로  $em$ 의 총 수  $n$  보다  $r$ 의 총 수  $m$ 이 매우 크다고 가정한다. ( $n \ll m$ )

정의 3.  $em_i$ 에 연결된  $r$ 들의 전체 집합을  $AR_i$  라 하면, 모든  $r$ 은  $em$  중 하나에 연결되어 있으므로, 다음이 성립한다.

$$R = \bigcup_{i=1}^n AR_i(em_i)$$

정의 4. 어떤  $em_i$ 가 가지고 있는 작업 부하가  $WL(em_i)$  (단,  $1 \leq i \leq n$ ) 라 할 때,

$$WL(em_i) = \sum WL(r) \cdot \{r \in AR_i\} \text{ ,}$$

로 정의한다.

정의 5. 어떤  $em_i$ 이 정상적인 수행이 가능한 최대 부하량 한계치인 Maximum threshold를  $WL^U(em_i)$ 라 정의한다. 이와 반대로, 전체적인 시스템의 성능 저하 및 지연을 유발할 수 있는 최소 부하 한계치인 Minimum threshold를  $WL^L(em_i)$ 라고 정의한다. Maximum threshold와 Minimum threshold의 평균  $WL^M(em_i)$ 은 다음과 같이 정의한다.

$$WL^M(em_i) = \frac{WL^U(em_i) + WL^L(em_i)}{2}$$

#### 3.3. 알고리즘

##### 3.3.1. 부하 분산 알고리즘 수행 시점

어떤  $em_i$ 에 대하여

$$WL(em_i) - WL^U(em_i) > 0 \text{ , ( 단, } 1 \leq i \leq n \text{ )}$$

가 성립하면,  $em_i$ 는 과부하 상태로 판별된다. 과부하 상태인  $em_i$ 를  $em_k$ 라고 하자.  $em_k$ 가 결정되면 부하 분산 알고리즘을 수행한다.

3.3.2. 리더 재배치를 위한 알고리즘

```
// sort  $AR_k$  in decreasing order of  $WL$ 
 $S = \langle r_1, r_2, \dots, r_{|AR_k|} \rangle$  such that  $r_i \in AR_k$ 
 $WL(r_i) \geq WL(r_{i+1}), 1 \leq i < |AR_k|$ 

DO
  for each  $r_i$  in  $S$ 
    find  $t (t \neq k)$  such that
       $WL(em_t) + WL(r_i) < WL^U(em_i)$ 
      and  $WL(em_t) + WL(r_i)$  최소의 값

     $AR_k = AR_k - r_i$ 
     $AR_t = AR_t + r_i$ 
     $WL(em_k) = WL(em_k) - WL(r_i)$ 
     $WL(em_t) = WL(em_t) + WL(r_i)$ 

  UNTIL  $WL(em_k) < WL^M(em_k)$ 
```

3.3.3. 부하 분산 기법

- 정보 수집 정책 : RFID 미들웨어의 부하 분산을 위해 수집될 수 있는 부하 정보는 CPU 이용률, 응답시간,  $em$ 의 처리량, 과부하 상태인  $em_k$ 에 연결된  $AR_k$ 에 대한  $\{WL(r_1), WL(r_2), \dots, WL(r_{|AR_k|})\}$  등이 있다. 단일 부하 정보로 정확한 부하량을 판단하기 어렵다고 판단될 경우에는 각 부하 정보에 대한 가중치를 두어 계산하는 방법을 사용할 수 있다.
- 시작 정책 : 본 논문에서 제시한 알고리즘은 과부하 상태로 판명된  $em_k$ 에 의해 부하 분산 정책이 시작된다. 부하 관리를 위한 마스터를 둘 경우에는 중앙 집중식(Centralized Initiation)방법을 채택할 수 있다. 반대로 분산 방식을 채택한다면  $em_k$ 가 Initiator로서의 역할을 한다.
- 작업 전송 정책 : Initiator가 부하 재배치를 고려하는 시점은 3.3.1. 절에 제시된 바와 같이 부하량이 Maximum threshold를 초과한  $em$ 이 발생하는 시점이다.
- 선택 정책 : 어떤  $em_i$ 의 어떤  $WL(r_i)$ 를 재배치 시켜야 할 작업으로 선정할 것인가를 뜻한다. 재배치를 위해 선정되는 작업은 하나 이상일 수 있다. 3.3.2. 절에 제시된 바와 같이 부하량이 Maximum threshold를 초과한  $em_k$ 에서 가장 큰 부하를 가지는  $WL(r_i)$  (단,  $r_i \in AR_k$ )이 우선적으로 재배치 되어야 할 작업으로 선정된다.  $WL(r_i)$ 를 선택하는 과정은  $WL(em_k)$ 가  $WL^M(em_k)$  보다 작아질 때 까지 계속 진행된다.
- 위치 선정 정책 : 선택 정책에 의해 이동될 부하로 선정된  $WL(r_i)$ 를 어떤  $em_t$ 으로 재배치할 것인가에 대한 결정이다.  $WL(em_t) + WL(r_i)$ 가 Maximum threshold를 넘지 않으면서 최소가 되는  $em$ 이 목적지 노드가 된다.

4. 결론

본 논문에서 제시한 RFID 미들웨어에 적합한 부하 분산 기법은 RFID 미들웨어 시스템의 특성에 부합한다. 부하 분산 정책에 따른 리더의 이동 횟수를 최소화하고 빈번한 과부하 상태의 발생을 방지하기 위한 알고리즘을 제시하였다. 향후 연구 과제는 시뮬레이션을 통해 제시된 알고리즘의 효과를 검증하는 것이다. 그리고 다른 알고리즘들과의 비교를 통해 실효성을 입증하고 RFID 미들웨어 시스템에 직접 적용할 것이다. 앞선 연구에서는 RFID 미들웨어 시스템의 가용성 관리를 위한 가용성 관리 프레임워크와 프레임워크를 기반으로 구현된 가용성 관리 시스템을 제시하였다. 본 논문에서 제시한 부하 분산 기법을 가용성 관리 시스템과 통합하여 확장성과 가용성을 동시에 제공할 수 있도록 하는 연구가 필요하다.

5. 참고문헌

- [1] Richard Korry, "A load sharing algorithm for a workstation environment", *Department Computer Science, FR 35, University Washington Seattle, Washington 98195, Tech. Report 86 10 3, oct. 1986*
- [2] HSUCi YIN Huang and Jane W. S. Liu., "Dynamic Load balancing Algorithms in Homogeneous Distributed System.", *U.S.Army, Report, no. UIUCDCS-R-86-1261, 1986*
- [3] Phillip Kraeger and Raphael Finkel, "An adaptive load sharing algorithm for a multicomputer", *Wisconsin univ. Tech. Report #539, April. 1984*
- [4] EPCglobal, The ID Systems(Tags and Readers) [http://www.epcglobalus.org/Network/ID\\_Systems.html](http://www.epcglobalus.org/Network/ID_Systems.html)
- [5] EPCglobal, EPC Middleware [http://www.epcglobalus.org/Network/EPC\\_Middleware.html](http://www.epcglobalus.org/Network/EPC_Middleware.html)
- [6] EPCglobal, EPC Information Services [http://www.epcglobalus.org/Network/EPC\\_Information\\_Services.html](http://www.epcglobalus.org/Network/EPC_Information_Services.html)
- [7] EPCglobal, The Network Components, [http://www.epcglobalus.org/Network/network\\_components.html](http://www.epcglobalus.org/Network/network_components.html)
- [8] Wensong Zhang, Shiyao Jin and Quanyuan Wu, "Creating Linux virtual Servers", *LinuxExpo Conf.*, 1999
- [9] Jaiganesh Balasubramanian, Douglas C.Schmidt, Lawrence Dowdy, and Ossama Othman, "Evaluating the performance of middleware load balancing strategies", *Proc. 8th IEEE Conf. on Enterprise Distributed Object Computing, 2004*
- [10] Cho Cho Myint, Khin Mar Lar Tun, "A Framework of Using Mobile Agent to Achieve Efficient Load Balancing in Cluster", *University of Computer Studies Yangon, Myanmar*
- [11] N. G. Shivaratri, P. Krueger, and M. Singhal, "Load Distributing for Locally Distributed Systems", *IEEE Int. Conf. Dist. Computer Systems, pp. 502-509, 1990*
- [12] S. Zhou, "A Trace-Driven Simulation Study of Dynamic Load Balancing", *IEEE Trans. Software Engineering, vol. SE-14, no. 9, pp. 1327-1341, September 1988*
- [13] M. M. Theimer and K. A. Lantz "Two adaptive Location Policies for Global Scheduling Algorithms.", *IEEE Int. Conf. Dist. Computer Systems, pp. 502-509, 1990*
- [14] S. Zhou, X. Zheng, J. Wang and P. Delisle, "Utopia: a Load Sharing Facility for Large, Heterogeneous Distributed Computer Systems.", *Software-Practice and Experience, vol. 23, no. 12, pp. 1305-1336, December 1993*
- [15] D.L. Eager, E. D. Lazowska, and J. Zahorjan, "A Comparison of Receiver-Initiated and Sender-Initiated Adaptive Load Sharing", *Performance Evaluation, Vol.6, pp.53-68 March 1986*