

# RFID 미들웨어 표준 아키텍처에 기반한 적응적 부하 분산 방법

박 재 겐<sup>†</sup> · 채 흥 석<sup>††</sup>

## 요 약

최근 RFID(Radio Frequency Identification) 기술은 사물에 대한 자동적인 인식을 가능케 함으로써 물류, 의료, 식품관리 등과 같은 분야에 적용되고 있다. 부하 분산은 과부하 상태인 노드로부터 부하가 적은 노드로 작업 부하를 이동시켜 시스템의 확장성을 향상시키는 기본 기술이다. 시스템의 부하를 예측하기 어렵고 부하량의 편차가 큰 경우에는 적응적 부하 분산이 효과적인 것으로 알려져 있다.

RFID 미들웨어는 많은 수의 리더로부터 수신된 태그 정보를 효율적으로 처리하기 위하여 기존의 부하 분산기술이 도입될 필요가 있다. RFID 시스템이 부하량을 예측하기 힘들고 편차가 큰 환경에 적용될 경우 실행시간에 시스템의 전체 부하량에 따라 적합한 정책으로 변경할 수 있는 적응적 부하 분산 기법을 사용하는 것이 바람직하다. 본 논문에서는 RFID 미들웨어에 적응적 부하 분산 기법을 도입하기 위한 접근 방법과 결과를 제시한다. 먼저 RFID 미들웨어의 작업 부하 모델을 결정한다. 그리고 부하 모델을 바탕으로 다양한 부하 분산 정책을 시스템의 부하 상태 별로 적용하여 시스템의 부하 상태에 적합한 부하 분산 정책을 선택한다.

키워드 : RFID 미들웨어, 적응적 부하 분산

## An adaptive load balancing method for RFID middlewares based on the Standard Architecture

Jaegel Park<sup>†</sup> · Heung Seok Chae<sup>††</sup>

### ABSTRACT

Because of its capability of automatic identification of objects, RFID(Radio Frequency Identification) technologies have extended their application areas to logistics, healthcare, and food management system. Load balancing is a basic technique for improving scalability of systems by moving loads of overloaded middlewares to under loaded ones. Adaptive load balancing has been known to be effective for distributed systems of a large load variance under unpredictable situations.

There are needs for applying load balancing to RFID middlewares because they must efficiently treat vast numbers of RFID tags which are collected from multiple RFID readers. Because there can be a large amount of variance in loads of RFID middlewares which are difficult to predict, it is desirable to consider adaptive load balancing approach for RFID middlewares, which can dynamically choose a proper load balancing strategy depending on the current load. This paper proposes an adaptive load balancing approach for RFID middlewares and presents its design and implementation. First we decide a performance model by a experiment with a real RFID middleware. Then, a set of proper load balancing strategies for high/medium/low system loads is determined from a simulation of various load balancing strategies based on the performance model.

Key Words : RFID Middleware, Adaptive Load Balancing

### 1. 서 론

RFID 태그 및 리더 가격의 안정화와 실제 비즈니스 환경에 적용된 시범사업의 성공적인 결과로 인해 RFID 기술에 대한 관심이 높아지고 있다[1]. RFID 기술은 상품의 식별 뿐만 아니라 이력과 위치에 대한 즉각적이고 정확한 정보를 제공할 수

있다[1, 2]. RFID 기술이 SCM(Supply Chain Management), 대형 마켓, 유통·물류 산업 등과 같은 다양한 분야에 혁신적으로 적용되면서 인프라스트럭처를 뒷받침할 수 있는 안정적인 RFID 시스템의 제공은 필연적인 과제가 되었다. RFID 시스템에 대한 EPC(Electronic Product Code)와 네트워크 표준화를 위한 비영리 국제기구인 EPCglobal[3]에 의해 제시된 EPC Network[4]는 RFID 산업의 표준으로 자리 잡았다. 이러한 EPC Network는 범세계적 상품 식별 네트워크의 표준으로서 개별 벤더의 RFID 시스템들이 인터넷을 통해 서로 연결될 수 있도록 한다.

부하 분산은 네트워크상에 분산되어 있는 자원을 효율적

\* “이 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임” (지방연구중심대학육성사업/차세대물류IT기술연구사업단)

† 정 회 원 : 세방㈜ 정보기술팀

†† 정 회 원 : 부산대학교 컴퓨터공학과 조교수

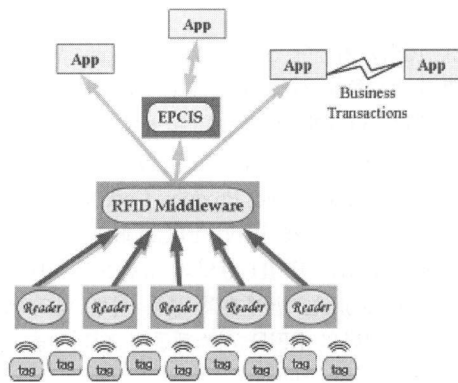
논문접수 : 2007년 6월 18일, 심사완료 : 2007년 8월 22일

으로 사용하여 높은 성능을 얻기 위한 방법이다. 일반적으로 부하 분산을 위해서는 시스템 환경과 각 노드의 자원 및 부하 상태를 고려하여 작업을 적절히 분배하거나 재할당하는 방법을 사용한다[5, 6, 7, 8, 9, 10]. 기존의 여러 연구들이 부하 분산을 통해 확장성을 제공하기 위해 노력하였다[11, 12, 13, 14]. 이러한 연구들은 부하 분산을 위한 기본 메커니즘으로 부하가 높은 노드로부터 부하가 적은 노드로 작업을 이동시키는 방법을 사용하였다. 분산 시스템에서 미들웨어의 부하 분산 기술이 일반적이면서도 중요한 기술이듯 RFID 미들웨어 역시 부하 분산 기술을 도입하여 시스템의 확장성을 향상시켜야 한다. 적응적 부하 분산 기법이란 로드밸런서가 시스템의 상태 및 환경정보의 변화에 따라 부하 분산 정책을 스스로 변경하는 기법을 뜻한다[5, 6]. 적응적 부하 분산 기법을 RFID 시스템에 적용하면 부하량의 변화에 따라 로드밸런서가 스스로 적합한 정책을 선택하게 된다.

(그림 1)은 EPC Network 표준을 따르는 RFID 시스템의 구성을 보여준다.

RFID 리더는 EPC 태그 정보를 읽어들이고, RFID 미들웨어는 리더로부터 수집한 태그 정보를 여과하여 상위 레이어에 위치한 RFID 애플리케이션으로 보고한다. EPCIS(EPC Information Service)는 객체 정보 저장소의 역할을 하며 사용자가 EPC Network를 통해 EPC 정보 검색을 요청할 경우 그에 맞는 정보를 제공한다. RFID 애플리케이션은 RFID 미들웨어를 통해 보고된 정보를 기반으로 자동화된 비즈니스 프로세스를 수행한다. 이와 같이 RFID 미들웨어는 RFID 시스템에서 리더 레이어와 애플리케이션 레이어 사이의 데이터 교환을 가능하게 하는 중추적인 역할을 한다[4]. RFID 미들웨어에 과부하 현상이 지속되면 처리의 효율성이 낮아짐은 물론 처리 지연으로 인한 비즈니스 프로세스의 지연 및 장애가 발생하게 된다. RFID 미들웨어의 과부하 문제는 일반적인 분산 시스템에서의 부하 분산 기술과 마찬가지로 작업 부하가 높은 미들웨어의 작업을 부하가 적은 미들웨어로 분산시킴으로써 해결할 수 있다.

시스템의 부하량을 예측할 수 있다면 정적 부하 분산 기법을 사용하는 것이 바람직하고, 부하량을 예측하기 어렵지만 부하량의 편차가 크지 않은 환경이라면 동적 부하 분산 기법을 적용함으로써 큰 효과를 거둘 수 있다. RFID 시스템



(그림 1) RFID 시스템의 구성

이 주로 배치되는 유통 및 물류 등의 분야는 부하량의 예측이 힘들고 편차가 매우 큰 환경이므로 시스템의 전체 부하 상태에 따라 알맞은 정책으로 변경할 수 있는 적응적 부하 분산 기법을 적용하는 것이 바람직하다.

본 논문의 구성은 다음과 같다. 2장에서는 표준 RFID 미들웨어의 부하 분산을 위해 필요한 기본 개념과 부하 분산 기본 정책을 소개한다. 3장에서는 기존에 연구된 다양한 부하 분산 정책들을 바탕으로 RFID 미들웨어에 부하 분산 기술을 적용시키는 방법을 설명한다. 4장에서는 RFID 미들웨어를 위한 적응적 부하 분산을 지원하는 부하 분산 시스템을 소개한다. 5장에서는 실제 RFID 미들웨어를 바탕으로 작업 부하 모델[15, 16]을 결정한다. 그리고 부하 모델을 바탕으로 적응적 부하 분산을 도입하기 위한 부하 분산 시뮬레이션 수행 결과를 기술한다. 6장에서는 관련 연구를 소개하고, 마지막으로 7 장에서는 결론 및 향후 연구 방향을 기술한다.

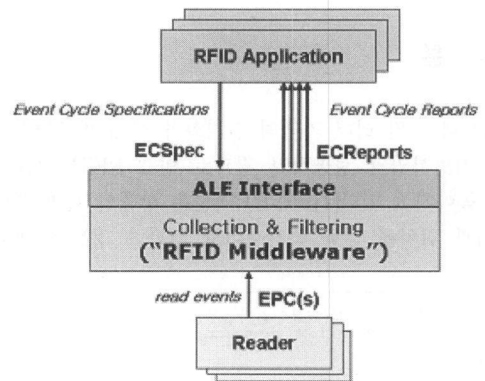
## 2. 연구 배경

본 장에서는 RFID 미들웨어의 부하에 큰 영향을 주는 표준 입력과 출력을 정의하고 있는 EPCglobal ALE(Application Level Events) 인터페이스[16]에 대해 기술하고 RFID 미들웨어의 기본 개념을 설명한다. 그리고 부하 분산 기술을 도입하기 위해 결정해야 하는 기본 정책들을 소개한다.

### 2.1 ALE 인터페이스

ALE 인터페이스는 RFID 미들웨어의 핵심 기능인 태그 정보 수집, 여과, 상위 레이어로의 전달 등에 대한 EPC Network 표준 인터페이스이다. (그림 2)는 ALE 표준을 따르는 RFID 미들웨어의 동작 방법을 보여준다.

RFID 미들웨어는 ALE 인터페이스를 통해 작업을 부여받고 그에 따른 처리 결과를 보고한다. 이벤트 사이클이란 미들웨어가 RFID 애플리케이션에게 보고하기 위한 정보를 만드는 주기이다[17]. ECSpec(Event Cycle Specification)[17]은 RFID 미들웨어가 이벤트 사이클마다 어떠한 처리 과정을 거쳐야 하는지를 기술하는 요구사항 명세이다. ECRReport[17]는 RFID 미들웨어가 ECSpec에 대한 처리 결과로 생성하는 보고 표준 형식이다.



(그림 2) ALE 인터페이스의 역할

(그림 3)은 RFID 애플리케이션의 작업 요청에 따른 각 요소들의 관계를 보여주는 개념도이다.

RFID 미들웨어는 ECSpec에 기술된 논리적 리더 정보와 미들웨어가 가지고 있는 물리적 리더 디바이스 정보를 맵핑하여 필요한 물리적 리더들을 선택하게 된다. 이와 같이 RFID 미들웨어는 ECSpec을 기본으로 하여 지정된 리더로부터 EPC 정보를 수집하고 ECSpec에 기술된 각종 사이클 정보와 필터링 정보 등을 고려하여 ECRReport 형식으로 RFID 애플리케이션에게 결과를 보고한다. RFID 미들웨어는 RFID 애플리케이션의 작업 요청에 대해 동일 노드에 배치된 자원만으로 작업을 처리하는 것이 아니라 다른 노드에 존재하는 물리적 리더와 의존 관계를 갖게 된다.

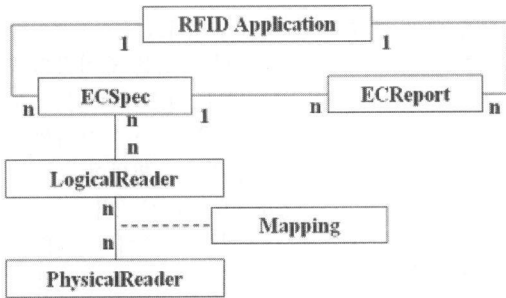
(그림 4)는 RFID 미들웨어에게 작업 처리를 요청하기 위한 ECSpec의 예를 보여준다.

• logicalReaders 매개변수 [17]

태그 정보 수집을 위해 사용될 논리적 리더 설정을 위해 사용된다. logicalReader 매개변수를 이용하여 하나 이상의 논리적 리더를 선택할 수 있다.

• boundarySpec 매개변수 [18]

RFID 미들웨어의 태그 정보 수집과 보고에 관련된 각종



(그림 3) ECSpec과 각 요소들 간의 관계

```

<ale:ECSpec xmlns:ale="urn:epcglobal:ale:xsd:1"
  <logicalReaders>
    <logicalReader>dock_1a</logicalReader>
    ...
  </logicalReaders>
  <boundarySpec>
    <repeatPeriod unit="MS">20000</repeatPeriod>
    <duration unit="MS">3000</duration>
    ...
  </boundarySpec>
  <reportSpecs>
    <reportSpec reportName="report3">
      <reportSet set="CURRENT"/>
      <filterSpec>
        <includePatterns>
          <includePattern>urn:epc:pat:gid-96:20.*.[5000-9999]</includePattern>
        </includePatterns>
      </filterSpec>
      <output includeCount="true"/>
    </reportSpec>
    ...
  </reportSpecs>
  ...
</ale:ECSpec>
    
```

(그림 4) ECSpec의 예[17]

시간 및 주기를 결정할 수 있다. repeatPeriod 매개변수를 통해 이벤트 사이클의 시작 시점부터 다음 시작 시점까지의 시간 간격을 정의하여 RFID 미들웨어의 보고 주기를 결정한다. duration 매개변수는 이벤트 사이클의 주기를 결정한다.

• reportSpecs 매개변수 [17]

reportSpec 매개변수를 통해 ECRReport 생성을 위한 필터링 방법, 태그 수 기재 여부, 보고 방법, 보고 단위 등을 설정할 수 있다. reportSet 매개변수의 set속성의 값이 ADDITION인 경우 이전 이벤트 사이클의 결과와 비교하여 새로 발견된 태그 정보에 대해서만 보고한다. 속성값이 DELETION인 경우에는 이전 이벤트 사이클의 결과와 비교하여 사라진 태그 정보만을 보고한다. 속성값이 CURRENT인 경우에는 이전 이벤트 사이클과는 무관하게 현재 이벤트 사이클에서 중복된 태그를 제거한 모든 태그 정보를 보고한다. includePattern 매개변수를 사용하여 EPC 패턴이 일치하는 태그 정보만을 추출할 수 있다. (그림 4)에 기술된 필터의 예는 GID 타입 96비트 포맷[17]의 모든 태그 중에 회사 코드가 20이고 모든 상품 코드를 포함하며 시리얼 번호가 5000에서 9999 사이인 태그 정보를 추출하도록 한다.

2.2 동적·적응적 부하 분산을 위한 기본 정책

분산 시스템의 부하 분산을 위해서는 고려해야 할 5가지 기본 정책이 있다. 본 절에서는 동적 부하 분산을 중심으로 각 정책에 대해 소개한다.

• 정보 정책(Information policy)

클러스터 내의 노드들로부터 작업 부하 정보를 수집하고 관리하는 방법을 결정하는 것이다. 구체적으로 설명하면 노드로부터 수집해야 할 정보, 평가 지표, 로드밸런서의 위치, 정보 수집 방법, 정보 수집 시점, 수집된 상태정보의 관리 방법 등을 결정하는 것이다. 가장 일반적으로 수집 대상이 되는 부하 정보로는 CPU 사용률(CPU usage rate), 프로세스 큐 길이(CPU run queue length), 초당 요청 수(Request per second), 처리량(Throughput), 응답 시간(Response Time) 등이 있다[6, 19]. 대표적인 정보 수집 방법으로는 브로드캐스트, 멀티캐스트, 개별 노드에 대한 폴링, 에이전트 기법, 이웃 노드와의 정보교환 등이 있다[7, 20]. 로드밸런서의 위치에 따라 분산형과 집중형 부하 분산으로 구분할 수 있다. 동일 클러스터 내에 노드의 수가 많아질 경우 병목현상에 의한 시스템의 성능 저하를 막기 위해 분산형 방식을 사용하는 것이 바람직하다. 하지만 참여 노드의 수가 많지 않은 경우와 부하 이동의 비용이 적은 경우에는 모든 노드에 대한 정확한 부하 정보를 수집하여 부하 분산 결정을 내릴 수 있고 쉽게 부하 정보를 모니터링할 수 있는 집중형 방식을 사용하는 것이 바람직하다[21, 22].

- Demand Driven 정책[6, 9, 12]

분산형 부하 분산 기법에서 사용된다. 개별 노드에 로

드벨런서가 배치되어 스스로 과부하 상태 여부를 판단한다. 각 로드벨런서가 부하 분산이 필요한 상태임을 판단하고 부하 분산 대상 노드를 찾아 분산 프로세스를 수행한다.

- Periodic 정책[6, 9, 12]

분산형과 집중형에 모두 적용할 수 있는 기법이다. 로드벨런서가 각 노드의 부하 정보를 주기적으로 수집하여 수집된 부하 정보를 바탕으로 부하 분산 메커니즘을 수행한다.

- State Change 정책[6, 9, 11, 12]

분산형과 집중형에 모두 사용할 수 있는 기법이다. 각 노드에서 노드의 부하 상태를 판별하여 상태의 변화가 있을 때만 로드벨런서에게 정보를 전송한다. 분산형에서는 정책에 따라 이웃 노드 또는 모든 노드에게 상태 정보를 전송하고 집중형의 경우에는 중앙 로드벨런서에게 상태 정보를 전송한다.

• 전송 정책(Transfer policy)

부하 분산이 필요한 시점을 판단하는 것이며 이러한 결정을 위해 로드벨런서는 수집된 노드의 부하 정보를 이용한다. [6, 9]

- Threshold 정책[6, 11, 12]

노드의 부하 상태 판단 지표에 대한 임계값을 사용하여 부하 상태를 결정한다. 과부하와 저부하의 2가지 상태를 구분하기 위해서는 하나의 임계값을 사용하는 1 threshold 정책을 적용한다. 그리고 과부하, 정상, 저부하의 3상태를 구분하기 위해서는 2개의 임계값을 사용하는 2 threshold 정책을 사용한다.

- Relative 정책[6, 19]

노드간의 부하량의 차이를 기준으로 부하 분산 시점을 판단한다. 부하가 가장 적은 노드와 비교 대상인 노드간의 부하량의 차가 기준 이상이 되면 부하 분산 프로세스가 수행된다.

• 시작 정책(Initiation policy)

부하 분산 프로세스를 누가 시작 시킬 것인가를 결정하는 것이다. Initiator는 과부하 상태인 노드 또는 부하가 적은 노드가 될 수 있으며, 두 노드 모두 Initiator가 될 수도 있다. 과부하 상태인 노드와 저부하 상태인 노드를 판단하는 기준은 전송 정책을 근거로 한다.

- Sender 정책[6, 9, 12, 23]

로드벨런서가 과부하 상태인 노드를 탐지하였을 경우 부하 분산 프로세스를 시작시킨다. 시스템의 전체 부하량이 낮은 환경에서 안정적으로 동작한다.

- Receiver 정책[6, 9, 12, 23]

로드벨런서가 저부하 상태인 노드를 탐지하였을 경우 부하 분산 프로세스가 시작된다. 시스템의 전체 부하량이 높은 환경에서 안정적으로 동작한다.

- Symmetrically 정책[6, 9, 12]

로드벨런서가 과부하 상태인 노드 또는 저부하 상태인 노드를 감지하였을 경우 부하 분산 프로세스가 시작된다. 시스템의 부하가 전반적으로 높거나 전반적으로 낮은 환경에서는 불필요한 부하 분산 시도 횟수가 증가하여 성능을 저하시킬 수 있다.

- Adaptive Symmetrically 정책[6, 9, 12]

Sender 노드가 탐지되었을 때 Receiver 노드 리스트가 비어있지 않을 경우 부하 분산 프로세스가 수행된다. 부하가 높은 환경에서 Receiver 후보 노드가 없음에도 Sender노드가 빈번하게 발생하여 불필요하게 지속적으로 부하 분산을 시도하는 것을 방지한다. 그리고 시스템의 부하가 낮은 환경에서 Sender 노드가 존재하지 않음에도 Receiver 노드가 탐지될 때 마다 부하 분산을 시도하는 것을 막는다. 이와 같이 불필요한 부하 분산 시도 횟수가 많은 Symmetrically 시작 정책을 보완한 정책으로서 우수한 성능을 보인다.

• 작업 선택 정책(Job Selection policy)

부하 분산 프로세스의 수행을 위해 어떤 작업을 재할당할 것인지를 결정하는 것이다[11, 12]. 비 선점 정책에서는 처리중인 작업은 작업 이동의 대상이 되지 않는다. 그러나 선점 정책에서는 처리중인 작업이라도 재할당될 수 있다[24].

- 작업 대기 큐의 작업 대기 순서[11]

작업 대기 큐를 두어 도착한 작업을 순차적으로 처리하는 환경에서 주로 사용된다. 가장 최근에 도착한 작업을 이동 대상으로 하거나 작업 대기 큐에서의 대기 순서를 응용하여 작업을 선택하는 방법이다.

- 작업의 가중치 및 비용[8, 10, 18, 25, 26]

도메인의 특성에 맞는 작업의 가중치와 비용을 고려하여 작업을 선택하는 방법이다. 그리고 작업의 이동 비용 및 부하를 예측하여 적합한 작업을 선택할 수 있다. 기존의 많은 연구들에서는 주로 특정 도메인에 특화된 계산 방법을 사용하였다.

• 위치 정책(Location policy)

전송 정책과 시작 정책에 의해 부하 분산을 시작시키는 노드가 결정되었을 경우 부하 분산 프로세스에 참여할 상대 노드를 선택하는 것이다. 즉 알맞은 작업 이동 파트너를 찾는 것이다[8, 14].

- LeastLoaded / MostLoaded 정책[6, 9, 19]

시작 정책을 기준으로 전송 정책에 의해 선택된 노드가 과부하 상태인 노드라면 LeastLoaded 정책을 사용하여 가장 부하가 적은 노드를 참여 대상으로 선택한다. 반대로 저부하 상태인 노드가 부하 분산 프로세스를 시작시킨 경우에는 MostLoaded 정책을 사용하여 가장 부하가 큰 노드를 참여 대상 노드로 선택한다.

- Random[6, 9]

무작위로 부하 분산에 참여할 대상 노드를 선택한다.



– Adjacent[6]

분산형 기법에서 사용되며 인접 노드를 부하 분산 대상 노드로 선택한다.

### 3. RFID 미들웨어를 위한 부하 분산 정책

본 장에서는 RFID 미들웨어에 부하 분산 기술을 적용하기 위한 부하 분산 정책을 소개한다. 정책 결정을 위해서 RFID 미들웨어의 특성과 배치되는 환경을 고려하였다. 5가지 기본 정책의 세부정책들을 조합한 18가지 정책 유형을 RFID 미들웨어의 동적 부하 분산에 적용한다.

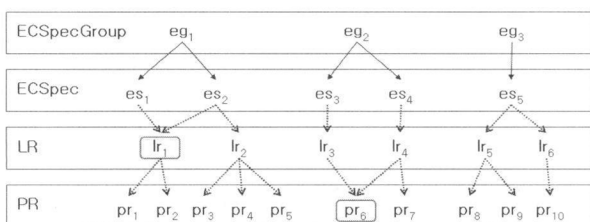
#### 3.1 작업 선택 정책

RFID 미들웨어는 ECSpec에 의해 생성된 작업들을 처리하기 위한 스케줄링 기법으로 FCFS(First-Come, First-Served) 방식을 사용하지 않고 운영체제의 프로세스 스케줄링 정책에 따라 경쟁적으로 처리한다. 그리고 ECSpec은 요청명세서의 역할을 할 뿐 ECSpec을 이동시키는 것이 실행중인 작업의 중단을 야기시키는 것은 아니다. 그러므로 ECSpec을 이동시켜 부하를 분산시킬 경우 선점형 기법과 같이 동작하지만 PCB 정보 및 I/O 정보[27] 등을 고려하지 않고 작업 이동이 가능하므로 작업 이동 비용 측면에서 매우 우수하다.

일반적인 RFID 리더는 여러 RFID 미들웨어의 동시 접근을 허용하지 않는다. 그러므로 RFID 미들웨어에서 동일한 리더를 공유하는 ECSpec들 중에 일부만을 다른 RFID 미들웨어로 이동시킬 경우 정상적인 처리가 이루어질 수 없다. 이러한 문제를 해결하기 위해 ECSpec과 RFID 리더의 의존 관계를 분석하여 이동이 가능한 작업 단위로 ECSpec을 그룹핑하는 방법을 사용한다.

(그림 5)는 ECSpec과 리더와의 의존 관계를 고려하여 ECSpec을 그룹핑하는 예를 보여준다.

ECSpec에 기술된 논리적 리더(LR)와 RFID 미들웨어가 가지고 있는 물리적 리더(PR) 정보를 맵핑시키면 이동 가능한 작업 단위인 ECSpecGroup을 결정할 수 있다. 논리적 리더 lr<sub>1</sub>의 경우 ECSpec es<sub>1</sub>과 es<sub>2</sub>에 의해 공유된다. 그리고 물리적 리더 pr<sub>6</sub>의 경우 논리적 리더 lr<sub>3</sub>와 lr<sub>4</sub>에 의해 공유된다. (그림 5)에 제시된 5개의 ECSpec들은 이러한 의존 관계에 따라 3개의 ECSpecGroup으로 그룹핑 될 수 있으며 이러한 ECSpecGroup은 작업 이동의 단위가 된다. ECSpec을 그룹핑하는 기능은 4장에서 소개하는 RFID 미들웨어 부하 분산 시스템을 통해 지원된다.



(그림 5) ECSpec과 리더와의 의존 관계를 고려한 그룹핑

본 논문에서는 RFID 미들웨어의 작업 이동 단위를 ECSpecGroup으로 정의하여 작업 선택 정책에서는 이동시킬 ECSpecGroup을 선택하게 된다. 그러므로 각 ECSpecGroup의 부하 정보를 수집하여 작업 선택을 위한 데이터로 사용한다. 각 ECSpecGroup의 부하량은 해당 ECSpecGroup에 속하는 처리중인 ECSpec의 수와 단위시간당 수집된 태그 수를 기준으로 한다. 각 ECSpecGroup은 ECSpec 수와 태그 수에 따라 부하량의 편차가 클 수 있기 때문에 작업의 선택 기준을 Maximal size, Middle size, Minimal size로 구분하여 선택의 폭을 넓혔다.

#### 3.2 정보 정책

모든 RFID 미들웨어의 상태를 실시간 모니터링 함은 물론 정확하고 신속한 부하 분산이 필요한 환경을 고려하여 집중형 부하 분산 기법을 도입하였다. RFID 시스템이 배치되는 환경은 많은 수의 미들웨어 보다는 정확한 부하 정보를 수집하여 신속히 부하 분산 프로세스를 수행시키는 것이 중요하다. 그러므로 집중형 부하 분산 기법에 적용 가능한 Periodic과 StateChange 정보 정책을 사용한다.

<표 1>은 RFID 미들웨어의 부하 상태를 판단하고 ECSpecGroup의 부하량을 판단하기 위해 수집해야 할 부하 정보를 보여준다.

RFID 미들웨어의 부하 상태를 판별하기 위해 평균 보고 시간과 평균 CPU 사용률이 사용되며 RFID 미들웨어 부하 분산 시스템 관리자는 판단 지표를 선택할 수 있다. 그리고 각 ECSpecGroup에 속하는 처리중인 ECSpec의 수와 초당 수집된 태그 수는 작업 선택 정책을 위해 수집되는 정보이다. 정보 정책으로 Periodic 정책을 사용할 경우 RFID 미들웨어는 RFID 미들웨어와 ECSpecGroup의 부하 정보를 주기적으로 로드밸런서에게 전송한다. 반면에 StateChange 정책이 선택될 경우 RFID 미들웨어가 배치된 클라이언트 측에서는 전송 정책에 근거하여 자신의 부하 상태에 변화가 생겼을 경우에만 부하 상태 정보와 ECSpecGroup 정보를 로드밸런서에게 전송한다.

#### 3.3 전송 정책

정보 정책에 의해 결정된 수집 정보를 바탕으로 언제 부하 분산 프로세스를 시작시킬 것인지를 결정한다. 구체적으로 설명하면 평균 보고 시간과 평균 CPU 사용률을 기준으로 RFID 미들웨어의 세 가지 상태를 구분한다.

<표 2>는 RFID 미들웨어의 부하 상태를 판단하기 위해

<표 1> RFID 미들웨어의 부하 분산을 위해 수집되는 부하 정보

부하 정보 구분	설명
RFID 미들웨어의 부하 정보	RFID 미들웨어의 평균 보고시간
	RFID 미들웨어의 평균 CPU 사용률
ECSpecGroup의 부하 정보	ECSpecGroup에 속한 처리중인 ECSpec의 수
	ECSpecGroup의 ECSpec에 설정된 리더들로부터 수집된 태그 수

〈표 2〉 RFID 미들웨어를 위한 threshold 정책

부하 상태	상태 판단 기준
저부하 상태	RFID 미들웨어의 부하 < 하위 임계치
정상적인 부하 상태	하위 임계치 ≤ RFID 미들웨어의 부하 ≤ 상위 임계치
과부하 상태	상위 임계치 < RFID 미들웨어의 부하

2 threshold 정책을 선택하였을 때 부하 상태와 상태 판단 기준을 보여준다.

2 Threshold 정책을 사용하여 RFID 미들웨어의 세 가지 상태를 정의하였다. 이와 같이 전송 정책에서 정의한 RFID 미들웨어의 부하 상태 결정 기준은 Sender 노드, 정상적인 노드, Receiver 노드 탐지의 기준이 된다.

3.4 시작 정책

RFID 미들웨어를 위한 부하 분산 프로세스를 시작시키는 주체를 결정한다. 전송 정책과 함께 RFID 미들웨어의 부하 분산 프로세스를 시작시키는 역할을 한다.

- Sender 시작 정책 : 특정 RFID 미들웨어가 과부하 상태인 것으로 판단되었을 경우 로드밸런서가 부하 분산 프로세스를 시작시킨다.
- Receiver 시작 정책 : 특정 RFID 미들웨어가 저부하 상태인 것으로 판단되었을 경우 부하 분산 프로세스가 시작된다.
- Ad-Symmetrically 시작 정책 : 저부하 상태인 RFID 미들웨어 리스트가 존재할 때 과부하 상태인 RFID 미들웨어가 탐지되면 부하 분산 프로세스가 시작된다. Symmetrically 정책은 빈번하게 불필요한 부하 분산을 시도하므로 시작 정책에서 제외하였다.

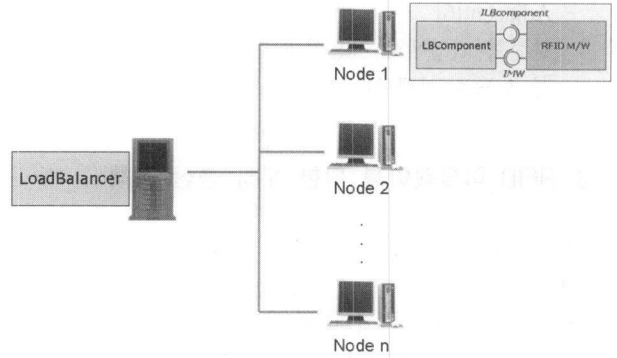
3.5 위치 정책

특정 RFID 미들웨어의 과부하가 감지되어 부하 분산 프로세스가 시작될 경우 가장 부하가 적은 RFID 미들웨어를 부하 분산 프로세스에 참여시키기 위해 LeastLoaded 정책을 사용한다. 그리고 저부하 상태인 RFID 미들웨어가 탐지되어 부하 분산 프로세스가 시작될 경우 부하가 가장 큰 RFID 미들웨어를 부하 분산 프로세스에 참여시키기 위해 MostLoaded 정책을 사용한다.

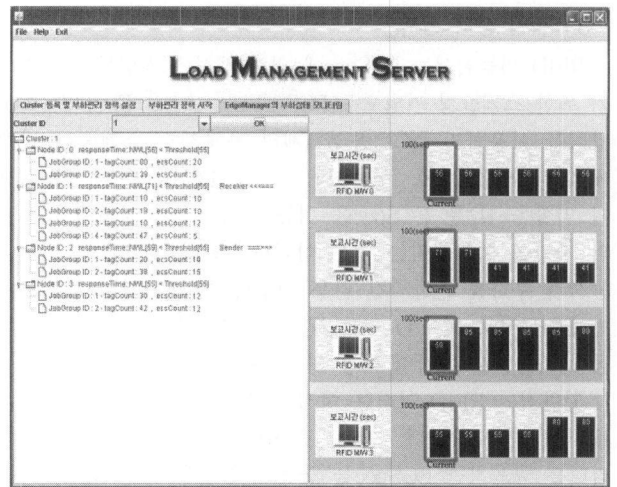
4. 부하 관리 시스템의 설계 및 구현

본 장에서는 3장에서 RFID 미들웨어를 위해 제시한 부하 분산 기본 정책을 지원하기 위해 구현한 RFID 미들웨어 부하 관리 시스템을 소개한다. (그림 6)은 부하 관리 시스템의 구성을 보여준다.

RFID 미들웨어 부하 관리 시스템은 집중형 부하 분산을 지원하기 위해 서버/클라이언트 모델을 채택하였다. 로드밸런서는 부하 분산 수행을 위한 정책을 결정하고 지시하는 역할



(그림 6) RFID 미들웨어 부하 관리 시스템

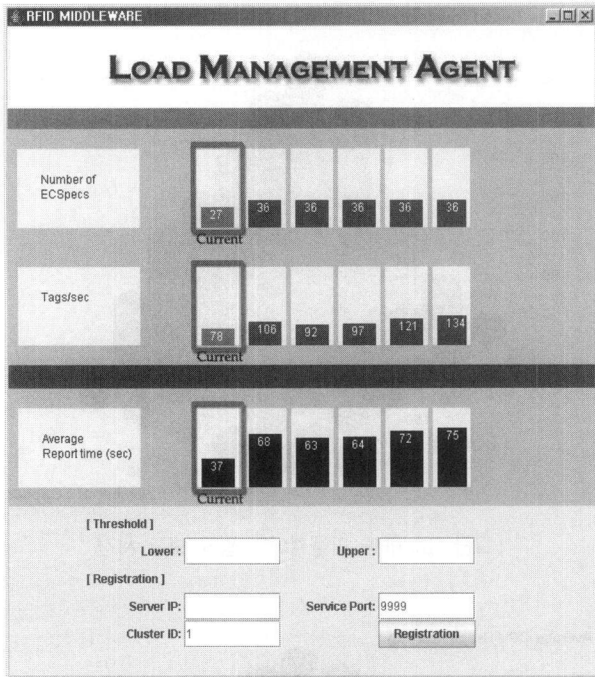


(그림 7) RFID 미들웨어 부하 관리 서버 : Load Balancer

을 한다. 그리고 LBComponent는 인터페이스를 통해 RFID 미들웨어와 상호 작용한다. LBComponent의 ILBComponent 인터페이스는 RFID 미들웨어의 기본 정보, ECSpec의 추가 및 삭제 정보, RFID 리더의 추가 및 삭제 정보 등을 보고 받기 위해 사용된다. 그리고 RFID 미들웨어의 IMW 인터페이스는 LBComponent가 직접 계산할 수 없는 RFID 미들웨어의 부하 정보를 수집하고 로드밸런서의 부하 분산 정책에 의해 결정된 사항을 RFID 미들웨어에게 명령하기 위해 사용된다.

(그림 7)은 로드밸런서의 역할을 하는 RFID 미들웨어 부하 관리 서버의 실행 화면을 보여준다.

RFID 미들웨어 부하 관리 서버를 통해 관리 대상인 RFID 미들웨어와 미들웨어에서 처리중인 ECSpecGroup의 부하 정보를 모니터링할 수 있다. 그리고 부하 관리 서버의 사용자 인터페이스와 로그 정보를 통해 작업 이동 현황 및 이동 이력을 확인할 수 있다. 좌측 화면은 4대의 RFID 미들웨어가 로드밸런서에 등록되어 있으며 총 111개의 ECSpec이 10개의 ECSpecGroup으로 그룹핑되어 있음을 보여준다. 그리고 평균 보고 시간을 성능 평가 지표로 부하 분산이 이루어지고 있음을 알 수 있다. 각 ECSpecGroup에 포함된 처리중인 ECSpec의 수와 RFID 리더들로부터 수신한 초당 태그의 수 역시 모니터링할 수 있다. 우측 화면은 각 RFID 미들웨어의 평균 보고 시간을 보여준다. 관리자가 부하 상태 판단



(그림 8) RFID 미들웨어 부하 관리 클라이언트

지표로 평균 CPU 사용률을 선택한다면 평균 CPU 사용률을 모니터링 할 수 있다.

(그림 8)은 평균 보고 시간이 부하 상태 판단 지표로 선택되었을 경우 RFID 미들웨어 부하 관리 클라이언트의 실행 화면을 보여준다.

관리자는 RFID 미들웨어와 동일 노드에 배치되는 LBComponent를 통해 RFID 미들웨어를 부하 관리 서버에 등록시켜 부하 관리 기능을 제공받을 수 있다. 로드밸런서에 등록하기 위해 부하 관리 서버의 IP와 자신이 속하게 될 클러스터를 선택하고 전송 정책을 위해 필요한 임계치를 입력한다. 그리고 화면을 통해 RFID 미들웨어가 처리중인 전체 ECSpec의 수, 리더들로부터 수신된 초당 태그의 수, 평균 보고 시간 등의 정보를 확인할 수 있다. 모니터링 되는 정보는 로드밸런서에서 결정한 정보 정책에 의해 결정된다.

## 5. RFID 미들웨어를 위한 적응적 부하 분산

RFID 미들웨어에 적응적 부하 분산 기법을 도입하기 위한 접근 방법과 결과를 소개한다. 먼저 RFID 미들웨어의 작업 부하 모델을 생성한다. 그리고 부하 모델을 기반으로 시스템의 부하 상태에 따른 각 정책 유형별 성능 시뮬레이션 결과를 제시한다. 시스템의 부하 상태에 따른 시뮬레이션 결과를 비교하여 RFID 미들웨어의 적응적 부하 분산에 적합한 부하 분산 정책 유형을 선택한다.

### 5.1 RFID 미들웨어의 작업 부하 모델

RFID 미들웨어의 작업 부하 모델 결정을 위해 부하에 많은 영향을 주는 부하 요소와 부하 상태를 판단할 수 있는 평

가 지표를 정의하였다. 그리고 RFID 미들웨어의 부하 요소에 따른 평가 지표를 측정하여 작업 부하 모델을 결정하였다.

#### 5.1.1 부하 요소 및 평가 지표

RFID 미들웨어의 부하에 많은 영향을 미치는 부하 요소로 ECSpec의 수와 초당 수집된 태그의 수를 사용하였다.

- ECSpec의 수 : ECSpec에 의해 결정되는 리더 설정, 태그 정보 수집 및 보고 주기 설정, 필터링 설정 등은 RFID 미들웨어의 부하를 발생시킨다. 이와 같이 RFID 미들웨어의 처리 과정에 큰 영향을 주는 ECSpec은 경쟁적으로 컴퓨팅 자원을 사용하므로 동시에 처리되는 ECSpec의 수가 많아질수록 RFID 미들웨어의 부하량은 높아진다.
- 초당 태그 수 : ECSpec에 명시된 태그 정보의 필터링 설정 및 보고 포맷 생성 방법 설정 등은 RFID 미들웨어가 처리하는 태그 정보의 수가 많아 질수록 연산의 양이 부하에 큰 영향을 준다.

RFID 미들웨어의 부하 상태를 판단할 수 있는 지표로 평균 보고 시간과 평균 CPU 사용률을 사용하였다.

- 평균 보고 시간 : RFID 미들웨어가 ECSpec에 기술된 repeatPeriod 매개변수의 설정 값에 따라 RFID 애플리케이션으로 처리 결과를 보고할 때까지 소요되는 평균 시간이다. 실시간 처리를 요하는 RFID 애플리케이션의 지정된 보고 주기를 만족시키지 못하면 RFID 애플리케이션의 자동화된 비즈니스로직의 오류 및 처리 지연을 야기하게 된다. 많은 기존의 연구에서 성능 평가 지표로 사용하는 평균 응답 시간과 유사한 개념이다.
- 평균 CPU 사용률 : 서버 및 미들웨어의 성능 지표로 널리 사용되는 평균 CPU 사용률은 RFID 미들웨어에서도 중요한 성능 평가 지표가 된다.

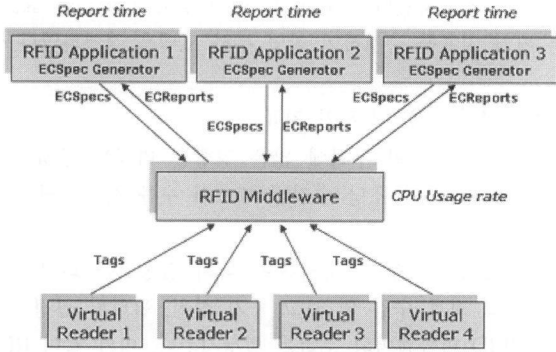
#### 5.1.2 작업 부하 모델의 실험 환경

‘A’사의 RFID 미들웨어를 대상으로 작업 부하 모델을 결정하였다. (그림 9)는 작업 부하 모델 결정을 위한 아키텍처를 보여준다.

부하 모델 결정을 위해 1개의 RFID 미들웨어, 3개의 RFID 애플리케이션, 4개의 가상리더를 배치하였다. 실제 RFID 미들웨어를 설치한 워크스테이션의 프로세서는 Intel Pentium4 3.02 GHz이고 메모리는 2GB이며 운영체제는 Windows Server 2003을 사용하였다.

- 사용된 ECSpec의 속성 값

20초마다 RFID 애플리케이션으로 보고하도록 하였으며 reportSet 매개변수의 Set 속성값을 CURRENT로 설정하였다. 이와 같은 ECSpec의 설정은 ALE Specification 1.0[17]에 기술된 ECSpec의 예제를 바탕으로 하였다. EPC 패턴 필터를



(그림 9) RFID 미들웨어 부하 모델 결정을 위한 아키텍처

사용하여 EPC 정보의 50%가 필터링 되도록 설정하였다.

• RFID 애플리케이션

RFID 애플리케이션은 ECSpec을 생성하고 ECRReport를 보고 받도록 하였다. 처리중인 ECSpec의 수는 1, 10, 30, 50, 70개로 설정하였다.

• 가상 리더와 태그

많은 양의 태그를 생성하기 위해 표준 태그 포맷을 따르는 가상 리더를 사용하였다. 사용된 EPC 포맷은 GID-96 (96bits tag)를 기준으로 하였다. 초당 태그 수는 포아송 분포를 따르도록 하고 평균값을 1, 50, 100, 150, 200, 250, 330개로 설정하였다.

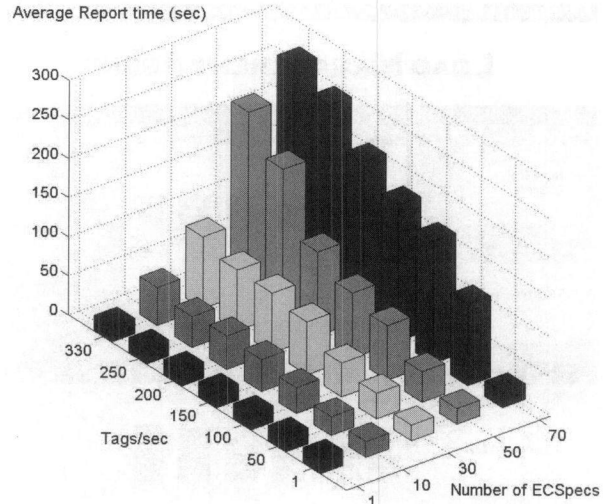
5.1.3 작업 부하 모델의 실험 결과

RFID 미들웨어의 부하 요소를 ECSpec의 수와 초당 태그 수로 두고 RFID 미들웨어의 평균 보고 시간과 평균 CPU 사용률을 측정하였다. (그림 10)은 ECSpec의 수와 초당 처리 태그 수의 변화에 따른 RFID 미들웨어의 평균 보고 시간을 보여준다.

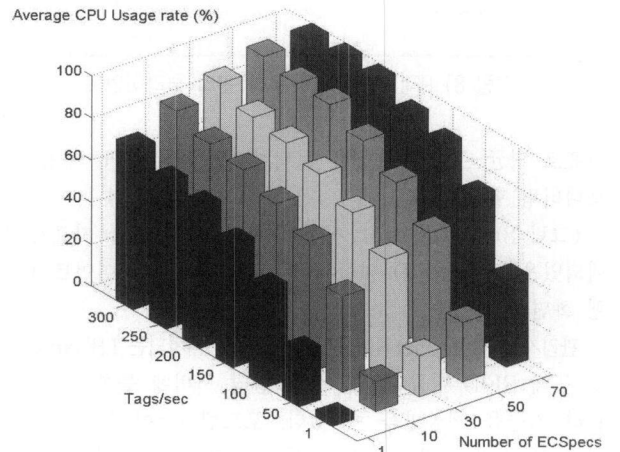
ECSpec의 수가 작을 경우 RFID 미들웨어는 태그 수에 큰 영향을 받지 않고 우수한 평균 보고 시간을 보인다. 그러나 동시에 처리되는 ECSpec의 수가 늘어남에 따라 태그 수의 증가에 따른 평균 보고 시간이 급격하게 증가하는 것을 확인할 수 있다.

(그림 11)은 ECSpec의 수와 초당 수집 태그 수의 변화에 따른 RFID 미들웨어의 평균 CPU 사용률을 보여준다.

평균 CPU 사용률에 대해서는 처리중인 ECSpec의 수가 초당 수집 태그수에 비해 보다 많은 영향을 주고 있음을 확인할 수 있다. 이와 같이 측정된 평균 보고 시간과 평균 CPU 사용률은 다중회귀분석을 위한 데이터로 사용되었다. 5회에



(그림 10) RFID 미들웨어의 평균 보고 시간



(그림 11) RFID 미들웨어의 평균 CPU 사용률

걸쳐 데이터를 수집하였으며 이를 기반으로 회귀분석 식을 도출하였다.

<표 3>은 다중회귀분석을 통한 RFID 미들웨어의 작업 부하 모델 결정 결과를 보여준다.

평균 보고 시간의 경우 모형은 결정계수( $R^2$ ) = .830, 조정 결정계수(Adj.  $R^2$ ) = .827, F값( $F$ ) = 334.313\*\*\*을 보여 변수의 설명력이 매우 높고 통계적으로 유의미하다. 그리고 평균 CPU 사용률에 대한 모형 역시  $R^2$  = .915, Adj.  $R^2$  = .914,  $F$  = 741.746\*\*\*로 변수의 설명력이 높다. 그리고 t검증 결과 Sig.역시 .000으로 추정되어 회귀계수가 유의미한 것을 확인하였다.

<표 3> RFID 미들웨어의 부하 모델

회귀 식	$R^2$	Adj. $R^2$	F
$\widehat{\text{평균보고시간}} = -51.987 + 2.008\text{ECSpec} + 0.456\text{tag}$	.830	.827	334.313***
$\widehat{\text{평균CPU사용률}} = 11.568 + 0.387\text{ECSpec} + 10.974\text{Intag}$	.915	.914	741.746***

5.2 적응적 부하 분산 정책

시스템의 부하량에 따른 적합한 부하 분산 정책을 선택하기 위해 부하 모델을 바탕으로 부하 분산 정책 유형별 성능 시뮬레이션을 수행하였다. 적용한 정책 유형은 3장에서 소개한 RFID 미들웨어를 위한 부하 분산 정책을 기초로 18가지 정책 유형을 시뮬레이션 하였다.

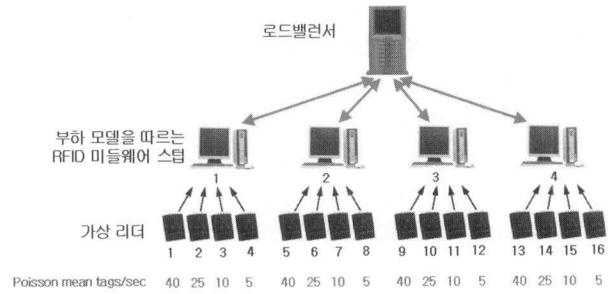
5.2.1 실험 환경

(그림 12)는 RFID 미들웨어의 작업 부하 모델을 바탕으로 시뮬레이션을 수행하기 위한 시스템 아키텍처이다.

로드밸런서는 4장에서 소개한 로드밸런서를 사용하였으며 실제 RFID 미들웨어 대신에 5.1절의 작업 부하 모델을 따르는 미들웨어 스텝을 적용하였다. 시뮬레이션을 위해 1개의 로드밸런서, 4개의 미들웨어 스텝, 16개의 가상리더를 사용하였다. 초기 구성은 (그림 12)와 같이 4개 미들웨어 스텝마다 4개의 가상 리더를 할당하였다.

<표 4>는 시스템의 부하 상태를 ‘상’, ‘중’, ‘하’로 나누기 위한 초기 설정을 보여준다.

부하 모델 정립 과정에 근거하여 ECSpec의 수를 45개, 30개, 15개를 기준으로 시스템의 부하 상태를 ‘상’, ‘중’, ‘하’로 구분하였다. 작업 부하 모델에서 확인할 수 있는 것과 같이 평균 보고 시간을 평가 지표로 설정할 경우 Tag 수 보다는 ECSpec의 수가 시스템의 부하에 큰 영향을 준다. 그러므로 실험의 편의성을 위해 tag 수 보다는 ECSpec의



(그림 12) 시뮬레이션을 위한 환경 구성

<표 4> 시스템의 부하 상태 구분을 위한 초기 설정

시스템의 부하상태	M/W당 평균ECSpec수	M/W 1	M/W 2	M/W 3	M/W 4
		ECSpec수	ECSpec수	ECSpec수	ECSpec수
상	45	60	55	50	15
중	30	55	35	20	10
하	15	40	10	5	5

수를 조절하여 시스템의 부하 상태를 조절하였다. 그리고 각 가상 리더에는 (그림 12)와 같이 포아송 분포에 따른 평균값을 설정하였다.

<표 5>는 <표 4>에서 정의한 시스템의 부하 상태 마다 시뮬레이션에 적용한 정책 유형을 보여준다.

<표 5> 세부 정책 설정에 따른 정책 유형

전략	정보 정책	전송 정책			시작 정책	작업 선택 정책	위치 정책
		부하 정보	t1	t2			
1	Periodic	평균보고시간	50	×	Sender	minimal size	LeastLoaded
2	Periodic	평균보고시간	50	×	Sender	middle size	LeastLoaded
3	Periodic	평균보고시간	50	×	Sender	maximal size	LeastLoaded
4	Periodic	평균보고시간	35	×	Receiver	minimal size	MostLoaded
5	Periodic	평균보고시간	35	×	Receiver	middle size	MostLoaded
6	Periodic	평균보고시간	35	×	Receiver	maximal size	MostLoaded
7	Periodic	평균보고시간	35	50	Ad-Sym	minimal size	LeastLoaded
8	Periodic	평균보고시간	35	50	Ad-Sym	middle size	LeastLoaded
9	Periodic	평균보고시간	35	50	Ad-Sym	maximal size	LeastLoaded
10	StateChange	평균보고시간	50	×	Sender	minimal size	LeastLoaded
11	StateChange	평균보고시간	50	×	Sender	middle size	LeastLoaded
12	StateChange	평균보고시간	50	×	Sender	maximal size	LeastLoaded
13	StateChange	평균보고시간	35	×	Receiver	minimal size	MostLoaded
14	StateChange	평균보고시간	35	×	Receiver	middle size	MostLoaded
15	StateChange	평균보고시간	35	×	Receiver	maximal size	MostLoaded
16	StateChange	평균보고시간	35	50	Ad-Sym	minimal size	LeastLoaded
17	StateChange	평균보고시간	35	50	Ad-Sym	middle size	LeastLoaded
18	StateChange	평균보고시간	35	50	Ad-Sym	maximal size	LeastLoaded



정보 정책에서 Periodic을 사용할 경우 RFID 미들웨어 스텝이 로드밸런서에게 3초 주기로 부하 상태 정보를 전송하고 로드밸런서는 10초 주기로 부하 분산 프로세스의 시작 여부를 결정하도록 하였다. 전송 정책은 평균 보고 시간을 기준으로 2 threshold 정책을 사용하여 ‘저부하’, ‘정상’, ‘과부하’ 상태로 나누었다. 위치 정책은 과부하 상태인 RFID 미들웨어 탐지 시 부하 분산 프로세스가 시작되는 경우는 LeastLoaded를 적용하고 저부하 상태인 RFID 미들웨어 감지 시 부하 분산 프로세스가 시작되는 경우에는 MostLoaded 정책을 사용하였다. <표 5>와 같이 부하 분산 정책의 조합에 따라 총 18가지의 전략을 결정하였다. 그리고 시스템의 부하 상태별로 18가지의 전략에 대한 시뮬레이션을 수행하여 결과를 수집하였다. 각 전략에 따른 시뮬레이션 수행 시간은 20분을 기준으로 하였다.

5.2.2 실험 결과

시스템의 부하 상태별 실험 결과는 다음과 같다.

• 시스템 부하 상태 ‘상’

(그림 13)은 시스템의 부하 상태가 매우 높은 환경에서 <표 5>에 기술된 정책 유형별로 부하 분산 시뮬레이션을 수행한 결과이다. (가)는 정책에 따른 평균 보고 시간을 측정한 것이며 (나)는 작업 이동횟수를 측정한 결과이다.

- 평균 보고시간 측면에서는 Periodic정책이 StateChange 정책보다 우수한 성능을 보였다. StateChange 정보 정책은 노드의 3상태만을 구분할 뿐 부하 정보를 수치화 시키지 않아 Periodic 정책에 비해 적합한 부하 분산 참여 미들웨어를 선택하는데 한계가 있다.
- 작업 이동횟수 측면에서는 Periodic 정보 정책과 Receiver

시작 정책을 선택하였을 경우 작업 이동 비용이 가장 낮게 측정되었다.

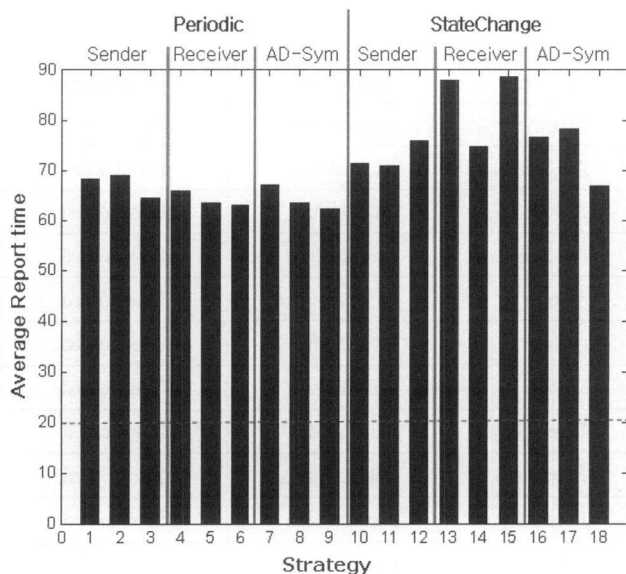
전략 13, 15, 16, 17, 18번의 경우 작업 이동 비용 측면에서는 우수하지만 수치화된 정확한 부하 정보를 수집하지 못한 결과 평균 보고시간의 경우 낮은 성능을 보였다. Sender 시작 정책을 선택하였을 경우 상위 임계치를 초과하는 RFID 미들웨어가 지속적으로 탐지되기 때문에 빈번한 작업 이동이 발생하였다. 이와 같이 부하가 심한 환경에서는 Receiver 시작 정책을 사용하여 하위 임계치 이하의 RFID 미들웨어가 탐지 되었을 때 ECSSpecGroup을 재할당하는 방법을 사용할 경우 우수한 성능을 보인다.

• 시스템 부하 상태 ‘중’

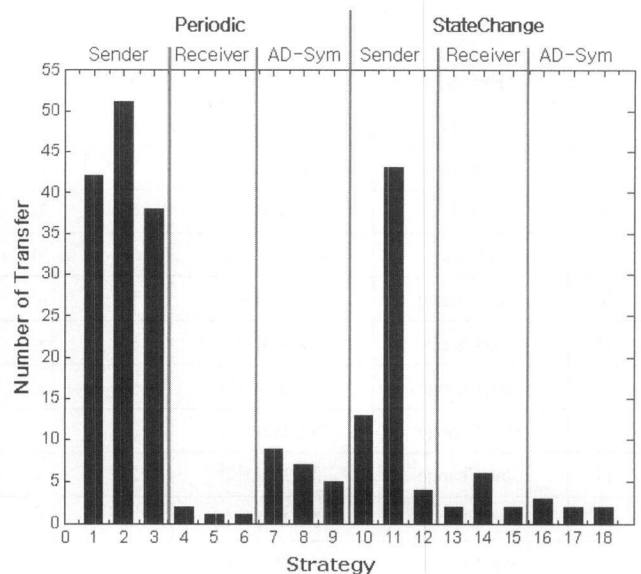
(그림 14)는 시스템의 부하 상태가 ‘중’인 환경에서 부하 분산 평가지표를 측정한 결과이다.

- 평균 보고시간 측면에서는 정확한 부하 정보를 수집할 수 있는 Periodic 시작 정책이 StateChange 시작 정책 보다 우수한 성능을 보였다. 그리고 Ad-Symmetrically 시작 정책을 선택할 경우 다른 선택 정책들에 비해 상대적으로 우수함을 확인할 수 있다.
- 작업 이동횟수 측면에서는 StateChange 정보 정책이 Periodic 정보 정책 보다 상대적으로 우수한 성능을 보였다.

StateChange 정책은 작업 이동횟수 측면에서는 우수하지만 평균 보고시간 측면에서는 낮은 성능을 보임을 확인할 수 있다. 부하 분산 프로세스에 참여하는 RFID 미들웨어를 선택하는데 있어서 부정확한 판단을 내렸다고 볼 수 있다. 최신 부하 정보를 바탕으로 정확한 참여 노드를 선택하도록

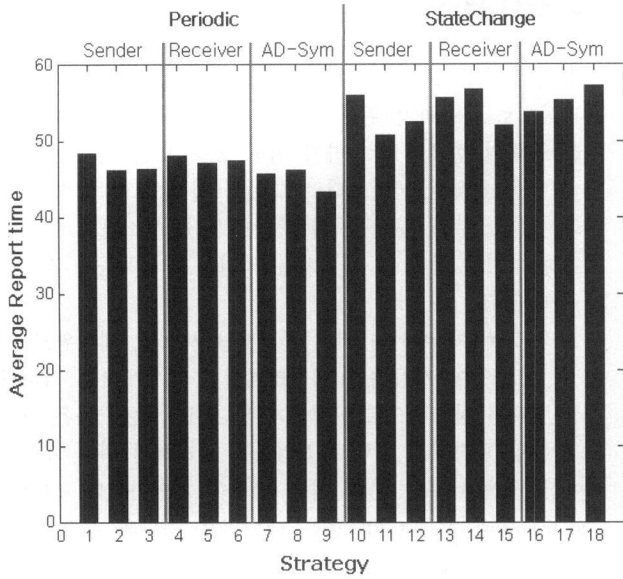


(가) 평균 보고시간

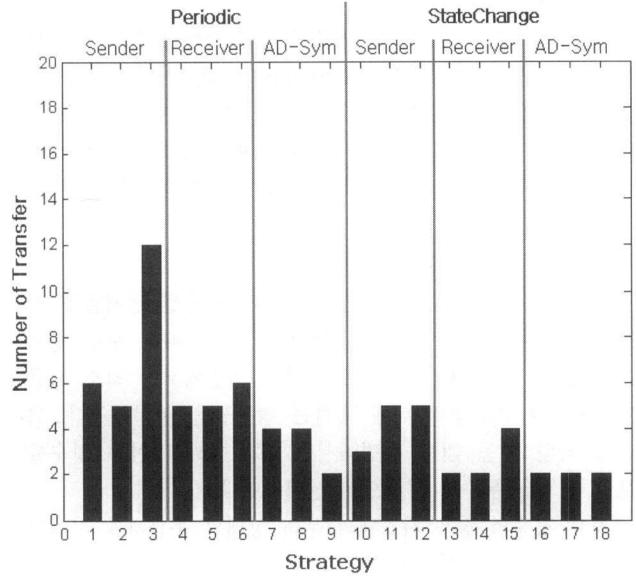


(나) 작업 이동횟수

(그림 13) 전략별 동적 부하 분산 결과 : 시스템 부하 상태 - 상

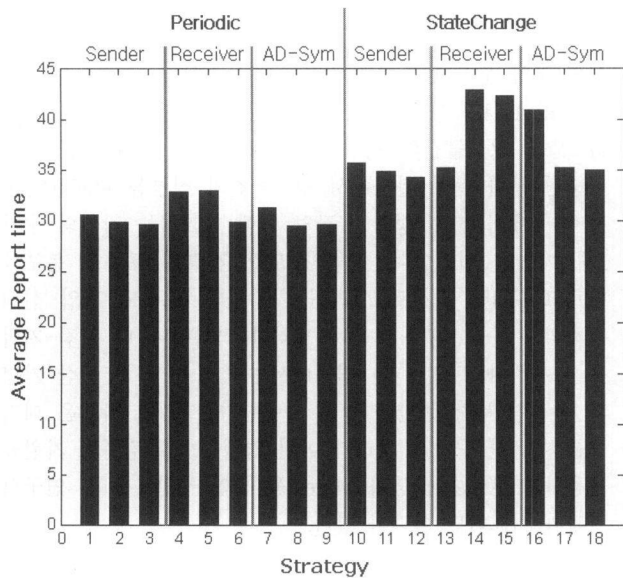


(가) 평균 보고시간

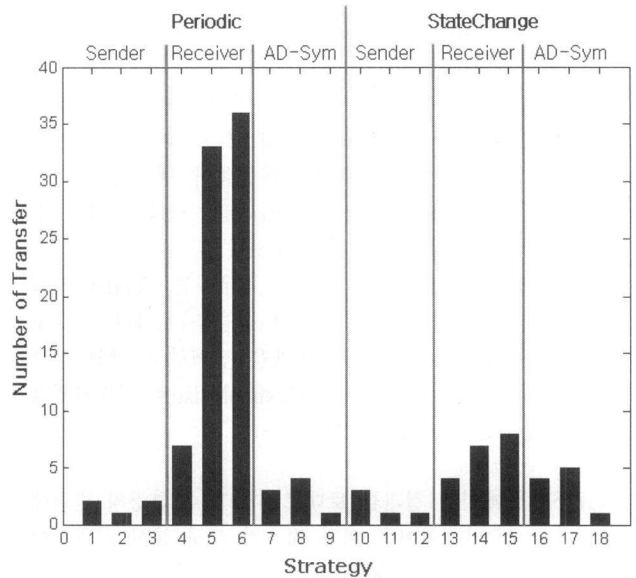


(나) 작업 이동횟수

(그림 14) 전략별 동적 부하 분산 결과 : 시스템 부하 상태 - 중



(가) 평균 보고시간



(나) 작업 이동횟수

(그림 15) 전략별 동적 부하 분산 결과 : 시스템 부하 상태 - 하

하는 Periodic 정보 정책을 사용하고 불필요한 부하 분산 시도 횟수를 줄이는 AD-Symmetrically 시작 정책을 사용하여 오히려 우수한 성능을 보임을 확인할 수 있다.

• 시스템 부하 상태 ‘하’

(그림 15)는 시스템의 부하 상태가 ‘하’인 환경에서 정책별 부하 분산 시뮬레이션을 수행한 결과이다.

- 평균 보고시간 측면에서는 시스템의 부하 상태가 ‘상’과 ‘중’인 경우와 마찬가지로 Periodic 정보 정책이 우수한 평균 보고 시간을 보였다. 그리고 Sender와 Ad-

Symmetrically 시작 정책이 Receiver 시작 정책보다 상대적으로 우수한 성능을 보임을 확인할 수 있다.

- 작업 이동 횟수 측면에서는 Sender 시작 정책과 Ad-Symmetrically 시작 정책이 우수한 결과를 보인다. Receiver 시작 정책을 적용할 경우 특정 RFID 미들웨어의 부하가 하위 임계치 미만으로 내려가면 과부하 상태인 미들웨어의 유무와 관계없이 전송 정책이 수행되어 불필요한 작업 이동횟수가 많아짐을 알 수 있다.

평균 보고시간과 작업 이동횟수를 모두 고려할 경우 Periodic 정보정책을 사용하고 Sender 시작 정책 또는 Ad-

Symmetrically 시작 정책을 사용하는 경우 평균 보고시간과 작업 이동횟수 측면에서 모두 우수한 성능을 보인다. 시스템의 부하가 낮은 환경에서 Ad-Symmetrically 시작 정책을 사용할 경우 Sender 탐지 이벤트가 발생하였을 때 Receiver 큐에 RFID 미들웨어가 존재할 확률이 매우 높으므로 Sender 시작 정책과 매우 유사하게 동작하여 비슷한 성능을 보인다.

5.2.3 적응적 부하 분산 정책

<표 6>은 적응적 부하 분산 정책에 사용될 정책들을 선정하기 위해 전체 시스템의 부하의 정도에 따른 우수 정책을 선별한 결과이다. 평균 보고 시간 측면에서는 최소 평균 보고시간의 7% 범위 내에 속하는 정책들을 선정하였다. 그리고 작업 이동 횟수 측면에서는 4회 이하의 이동 횟수를 보인 정책을 우수한 정책으로 선택하였다.

정보 정책의 경우 시스템의 부하 상태와 무관하게 정확한 정보 수집이 가능한 Periodic 정보 정책을 사용할 경우 높은 성능을 보였다.

- 시스템의 부하 상태가 ‘상’인 경우 저부하 상태인 RFID 미들웨어가 탐지될 경우 무조건 부하 분산을 시도하는 Receiver 시작 정책을 사용할 때 우수한 성능을 보였다.
- 시스템의 부하 상태가 ‘중’인 환경에서는 저부하 상태인 RFID 미들웨어에 대한 리스트가 비어있지 않은 경우 과부하 상태인 RFID 미들웨어 탐지되면 부하 분산을 시도하는 Ad-Symmetrically 시작 정책을 선택하였을 때 우수한 성능을 보였다.
- 시스템의 부하 상태가 ‘하’인 경우에는 Sender 시작 정책을 사용하는 경우 우수한 성능을 보였다. 그리고 Ad-Symmetrically 시작 정책을 선택하고 작업 선택 정책을 Middle size 또는 Maximal size로 설정하였을 때 우수한 결과를 보였다.

<표 6>에 제시된 결과를 토대로 시스템 부하량의 변화에 적응하는 RFID미들웨어를 위한 적응적 부하 분산 시스템에 사용될 전략을 선정하였다.

6. 관련 연구

분산 시스템 환경에서 부하 분산은 가용한 컴퓨팅 자원을 효과적으로 사용하기 위한 기술이다. 그리고 단일 시스템의 처리 능력의 한계를 극복하기 위해 동종 또는 이기종 컴퓨팅 자원을 하나의 클러스터로 그룹화 하여 처리 능력을 극대화시킨다. 부하 분산을 위해서는 분산 시스템의 특징에 기초하여 기본 부하 분산 절차를 정의하고 정책들을 결정해야 한다[5, 6].

그리드 컴퓨팅에서는 서버 풀, 저장 시스템, 네트워크 등을 하나로 묶어 사용자와 애플리케이션 레벨에서는 커다란 가상 컴퓨팅 시스템으로 인식하게 된다. Paylii[28] 등은 그리드 컴퓨팅 환경에서 병렬로 작업들을 처리하기 위한 애플리케이션 레벨의 부하 분산 접근 방법을 제안하였다. 그리고 Cao[13] 등과 Myint[20] 등은 에이전트 기반의 부하 분산 기법을 시도하였다. 부하 분산에 대한 많은 연구결과가 있지만 대부분의 연구들이 동종의 노드를 가정하고 작업 대기 큐를 두어 FCFS(First-Come, First-Served) 형태의 스케줄링 정책을 사용한다. 그렇지 않다 하더라도 배치된 작업들 혹은 장치들 간의 의존관계가 없는 독립적인 작업을 가정하고 있다. 그러므로 이러한 기존의 연구결과를 RFID 미들웨어의 부하 분산에 그대로 적용할 수는 없다.

Krueger[12] 등은 분산형 부하 분산 기법을 기준으로 부하 분산 정책들간의 성능을 비교하였으며 Marei[11] 등은 집중형 부하 분산 환경에서 정책들간의 성능을 비교하였다. 두 연구에서 전송 정책은 작업 큐에 대기하는 작업의 수가 2개 이상인 시점이다. 작업의 도착 간격은 포아송 분포를 따르고 FCFS 순서로 처리될 가정한다. 그리고 작업 선택 정책의 기준은 가장 최근에 도착한 작업이며 비 선점 기법을 사용한다. 반면에 본 연구에서는 집중형 부하 분산 기법과 선점형 부하 분산 기법을 사용한다. 그리고 작업은 대기 큐를 이용한 순차적 처리가 아닌 경쟁적으로 컴퓨팅 자원을 사용한다. Krueger 등과 Marei 등의 연구 결과를 본 연구의 결과와 비교하면 아래와 같다.

<표 6> 적응적 부하 분산을 위한 정책 선정

시스템 부하 상태	정보정책	Periodic									StateChange								
	시작정책	Sender			Receiver			Ad - Sym			Sender			Receiver			Ad - Sym		
	작업정책	min	mid	max	min	mid	max	min	mid	max	min	mid	max	min	mid	max	min	mid	max
	유형 ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
상	평균 보고시간			0	0	0	0		0	0									
	작업 이동횟수				0	0	0					0	0		0	0	0	0	0
중	평균 보고시간		0	0				0	0	0									
	작업 이동횟수							0	0	0	0			0	0	0	0	0	0
하	평균 보고시간	0	0	0			0		0	0									
	작업 이동횟수	0	0	0				0	0	0	0	0	0	0				0	0

- Krueger 등의 연구에서는 분산형 부하 분산 기법과 작업 대기 큐를 전제로 시스템의 부하를 ‘중’에서 ‘상’까지 변화시켜 가면서 Receiver, Sender, Symmetrically, Ad-Symmetrically 시작 정책의 평균 응답 시간을 비교하였다. ‘중’과 ‘상’의 환경에서 Ad-Symmetrically 정책이 가장 우수한 성능을 보였으며 ‘상’인 경우에는 Receiver 정책과 Ad-Symmetrically 정책이 비슷한 성능을 보였다. 반면에 본 논문에서는 집중형 부하 분산 기법을 사용하고 작업들 간의 경쟁에 의해 자원을 사용하는 것을 전제로 한다. 시스템의 부하가 ‘중’인 환경에서는 Krueger 등과 같은 결과를 보였으나 시스템의 부하가 ‘상’인 환경에서는 Receiver 시작 정책이 Ad-Symmetrically 시작 정책 보다 우수한 성능을 보였다.
- Marei 등의 연구에서는 집중형 부하 분산 기법을 전제로 하여 시스템의 부하가 ‘하’인 상태에서 ‘상’인 상태로 증가시켜 가면서 Sender 시작 정책과 Ad-Symmetrically 정책을 비교 하였다. 시스템 전체의 부하가 ‘하’인 환경에서 Sender 시작 정책과 Ad-Symmetrically 시작 정책이 모두 우수한 성능을 보여 본 본문과 같은 결과를 보였다. 부하 상태가 ‘중’인 경우에 Marei 등의 연구에서는 Sender 시작 정책과 Ad-Symmetrically 시작 정책의 차이가 없었으나 본 연구에서는 Ad-Symmetrically 시작 정책이 Sender 시작 정책보다 우수한 결과를 보였다.

## 7. 결론 및 향후 연구 방향

본 논문에서는 표준 RFID 미들웨어의 확장성 향상을 위한 적응적 부하 분산 기술 도입 방법과 적용 사례를 기술하였다. 첫째, 기존의 부하 분산 기본 기법과 정책들을 RFID 미들웨어에 적용하기 위한 방안과 RFID 미들웨어에 특화된 정책을 제시하였다. 둘째, EPCglobal ALE 표준 인터페이스를 만족하는 RFID 미들웨어의 적응적 부하 분산을 위한 시스템을 제시하였다. 셋째, RFID 미들웨어의 부하에 많은 영향을 주는 부하 요소와 성능 및 부하 상태 판단을 위한 지표를 제시하고 작업 부하 모델을 결정하였다. 넷째, 전체 시스템 부하량의 변화에 따른 적응적 부하 분산 기법을 적용하기 위한 시뮬레이션 결과를 제시하였다.

향후 연구에서는 여러 벤더들의 RFID 미들웨어를 대상으로 작업 부하 모델을 결정할 예정이다. 그리고 RFID 미들웨어 수의 변화와 임계값의 변화에 따른 성능 시뮬레이션 결과를 도출할 것이다. 이와 같은 과정을 거쳐 RFID 미들웨어를 위한 적응적 부하 분산 기법을 개선할 것이다.

## 참 고 문 헌

- [1] Ron Weinstein, "RFID: A Technical Overview and Its Application to the Enterprise," *IT Professional*, Vol.07, No.3, pp.27-33, 2005.
- [2] 김현, "컨테이너터미널의 RFID 효과 분석," 한국해양대학교 대학원 박사학위논문, 2007.
- [3] EPCglobal Inc, <http://www.epcglobalinc.com>
- [4] Ken Traub, et al., "The EPCglobal Architecture Framework," EPCglobal, 2005.
- [5] T.L. Casavant and J.G. Kuhl, "A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems," *IEEE Transactions on Software Engineering*, Vol.14, No.2, pp.141-154, 1988.
- [6] N.G. Shivaratri, P. Krueger, and M. Singhal, "Load Distributing for Locally Distributed Systems," *Computer*, Vol.25, No.12, pp.33-44, 1992.
- [7] O. Kremien and J. Kramer, "Methodical Analysis of Adaptive Load Sharing Algorithms," *IEEE Trans. Parallel and Distributed Systems*, Vol.3, No.11, pp.747-760, 1992.
- [8] Chin Lu and Sau Ming Lau, "An adaptive load balancing algorithm for heterogeneous distributed systems with multiple task classes," *Distributed Computing Systems*, pp.629-636, 1996.
- [9] Orly Kremien, Jeff Kramer, and Jeff Magee, "Scalable, adaptive load sharing for distributed systems," *IEEE Parallel and Distributed Technology*, Vol.1, No.3, pp.62-70, 1993.
- [10] Chang-Jia Wang, Krueger, P., and Liu, M.T., "Intelligent job selection for distributed scheduling," *Proceedings the 13th International Conference on Distributed Computing Systems*, pp.517-524, 1993.
- [11] M. S. Al-Amri and R. E. Ahmed, "New job selection and location policies for load-distributing algorithms," *International Journal of Network Management*, Vol.12, No.3, pp.165-178, 2002.
- [12] Krueger, P. and Shivaratri, N.G., "Adaptive location policies for global scheduling," *IEEE Transactions on Software Engineering*, Vol.20, No.6, pp.432-444, 1994.
- [13] J. Cao, D. P. Spooner, S. A. Jarvis, S. Saini, and G. R. Nudd, "Agent-based Grid Load Balancing using Performance driven Task Scheduling," *In: IPDPS*, pp.49, 2003.
- [14] Corradi, A., Leonardi, L., and Zambonelli, F., "Diffusive load-balancing policies for dynamic applications," *IEEE Parallel and Distributed Technology*, Vol.7, No.1, pp.22-31, 1999.
- [15] M. Calzarossa and G. Serazzi, "Workload Characterization: A Survey," *Proc. IEEE*, Vol.81, No.8, pp.1136-1150, 1993.
- [16] S.P. Dandamudi, "Sensitivity evaluation of dynamic load sharing in distributed systems," *IEEE Parallel and Distributed Technology*, Vol.6, No.3, pp.62-72, 1998.
- [17] EPCglobal Inc. The application level events(ALE) specification, version 1.0.
- [18] Neil Garde, et al., "Application Level Event(ALE) 1.02 Conformance Requirements Document," 2005.
- [19] Jaiganesh Balasubramanian; Schmidt, D.C.; Dowdy, L.; Othman, O., "Evaluating the performance of middleware load balancing strategies," *Enterprise Distributed Object*

*Computing Conference*, pp.135-146, 2004.

[20] C. C. Myint and K. M. L. Tun., "A framework of using mobile agent to achieve efficient load balancing in cluster," *In: Proc of APSITT*, pp.66 - 70, 2005.

[21] Marvin M. Theimer and Keith A. Lantz, "Finding idle machines in a workstation based distributed system," *IEEE Transactions on Software Engineering*, Vol.15, No.11, pp. 1444-1458, 1989.

[22] W. Zhu and C. Steketee, "An experimental study of load balancing on Amoeba," *In First Aizu International Symposium on Parallel Algorithms/Architecture Synthesis. IEEE*, pp. 220-226, 1995.

[23] D. L. Eager, E. D. Lazowska, and J. Zahorjan, "Adaptive Load Sharing in Homogeneous Distributed Systems," *IEEE Transactions on Software Engineering*, Vol.12, No.5, pp.662 -675, 1986.

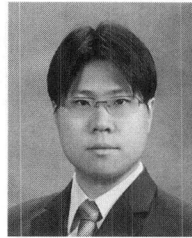
[24] P. Krueger and M. Livny, "A Comparison of Preemptive and Non-Preemptive Load Distributing," *Proc. IEEE Int'l Conf. Distributed Computing Systems*, pp.123-130, 1988.

[25] M. Kafil and I. Ahmad, "Optimal Task Assignment in Heterogeneous Distributed Computing Systems," *IEEE Concurrency*, Vol.6, No.3, pp.42-51, 1998.

[26] C. J. Hou and K.G. Shin, "Load Sharing with Consideration of Future Task Arrivals in Heterogeneous Distributed Real-Time Systems," *IEEE Trans. Computers*, Vol.44, No.9, pp. 1076-1090, 1994.

[27] Dejan S. Milošević, Fred Douglass, Yves Paindaveine, Richard Wheeler, and Songnian Zhou, "Process migration," *ACM Computing Surveys*, Vol.32, No.3, pp.241-299, 2000.

[28] R. Payli, E. Yilmaz, A. Ecer, H. Akay, and S. Chien, "A dynamic load balancing tool for grid computing," *In: Proc of Parallel CFD*, 2004.



### 박재길

e-mail : jgpark@sebang.com

2004년 신라대학교 컴퓨터교육과(학사)

2007년 부산대학교 대학원 컴퓨터공학과  
(공학석사)

2007년~현재 세방(주) 정보기술팀

관심분야: 객체지향 방법론, CBD, Framework, RFID 미들웨어, 가용성, 확장성



### 채흥석

e-mail : hschae@pusan.ac.kr

1994년 서울대학교 원자핵공학과(학사)

1996년 한국과학기술원 대학원 전산학과  
(공학석사)

2000년 한국과학기술원 대학원 전산학과  
(공학박사)

2000년~2003년 동양시스템즈(주) 기술연구소 선임연구원

2003년~2004년 한국과학기술원 전산학과 초빙교수

2004년~현재 부산대학교 컴퓨터공학과 조교수

관심분야: 객체지향 방법론, 소프트웨어 테스트, 소프트웨어 메트릭, 소프트웨어 유지보수, 미들웨어 설계, 프로덕트라인 공학