한국지리정보학회지 12권 4호 : 113~124(2009)

U-GIS 기반 도시시설물 관리 분야의 그리드(GRID) 아키텍처 적용 연구*

남상관[1※] · 오윤석[1] · 류승기[2] · 권혁종[3]

An Application of GRID Architecture on a Part of Urban Facilities Management Based on U-GIS*

Sang-Kwan NAM[1※] · Yoon-Seuk OH[1] · Seung-Ki RYU[2] · Hyuk-Jong KWON[3]

요 약

다수의 분산된 컴퓨터들을 단일 컴퓨터처럼 활용하여 고성능 대용량의 컴퓨팅 기능을 얻기 위한 그리드 컴퓨팅 기술은 다양한 분야에서 연구가 활발히 진행 중에 있다. 본 연구는 u-GIS 기반 도시시설물 관리 시스템에 그리드 기술을 적용한 연구이다. 시설물의 상태를 실시간 모니터링 하기 위해 도시 시설물에 각종 센서가 설치되는데, 이 센서의 수가 늘어나고, 센서 데이터를 수집하는 게이트웨이의 수가 늘어날 경우 서버에서는 데이터의 처리를 위해 많은 컴퓨팅 자원이 필요하게 된다. 본 연구에서는 이렇게 서버의 부하가 늘어나 임계치에 근접할 경우, 이를 판단하여 서버의 작업 일부를 유휴 게이트웨이로 분산시킬 수 있는 기술을 개발하였다. 본 시스템을 향후 u-City 등 대용량 데이터 처리 분야에 활용할 경우 경제적이고 효율적인 시스템 개발이 가능할 것이다.

주요어 : 그리드 컴퓨팅, 유비쿼터스 센서 네트워크, 시설물 관리, 유비쿼터스 GIS

ABSTRACT

A research of grid computing that is the combination of computer resources from multiple distributed administrative domains to get large amount of computing power is underway. This research is an application of grid technology on a urban facilities management system based on u-GIS. The sensors are set up in the urban facilities to make it monitoring. If an amount

2009년 11월 19일 접수 Received on November 19, 2009 / 2009년 12월 15일 수정 Revised on December 15, 2009 / 2009년 12월 16일 심사완료 Accepted on December 16, 2009

* 본 연구는 국토해양부 첨단도시기술개발사업 - 지능형국토정보기술혁신사업과제의 연구비지원(06국토정보C01)에 의해 수행되었습니다.

1 한국건설기술연구원 U-국토연구실 Ubiquitous Land Implementation Research Division, KICT

2 한국건설기술연구원 도로연구실 Highway Research Division, KICT

3 (주)웨이버스 기술연구소 Research Center, WAVUS

※ 연락저자 E-mail : griffey@kict.re.kr

of sensors and gateways is increased, the server needs more computing resources to process the data. In this study we developed the skills that can distribute jobs to idle gateway, in case of the server capacity had approached threshold. It will be possible to develop of economic and efficient system that will apply to large amount of data processing about u-City.

KEYWORDS : Grid Computing, USN, Facility Management, U-GIS

서 론

1. 연구배경

최근 전국 지자체에서 경쟁적으로 추진 중인 u-City 건설 사업이나 국토해양부의 연구개발 사업인 '지능형국토정보기술혁신사업'의 '도시시설물 지능화 기술개발'과제 등은 기존 재래 기술 기반 도시 관리기법에 첨단 유비쿼터스 컴퓨팅 관련 기술을 융합하여, 실시간으로 시설물의 상태를 모니터링 하기 위한 대표적인 사업이다. 기존에도 다양한 분야에 분야에 USN과 GIS 기술을 접목한 u-GIS 관련 연구가 다수 진행되었다. 김의명(2006)등은 무선인식 기술과 지형공간정보체계를 이용하여 가로수를 관리하기 위한 방안을 연구하였고, 전문장(2009) 등은 과수의 생육환경을 모니터링 하기 위해 GIS와 USN 기술을 활용하였다. 이러한 기술이 보편화 될 경우 수많은 센서가 설치될 것이며, 센서로부터 실시간으로 수집되는 정보를 처리하기 위해 막대한 컴퓨팅 파워가 필요하게 될 것으로 예상된다. 현재의 컴퓨터 성능으로 이렇게 많은 양의 데이터를 처리하기 위해서는 수퍼 컴퓨터와 같은 초고성능 컴퓨터가 필요하다. 그러나 이러한 컴퓨터를 구입하고 운영하기 위해서는 막대한 비용이 소요되고, 기술적으로 고도의 전문가가 필요하는 등 일반 지자체에서 사용하기에는 어려움이 있다. 이러한 문제를 해결할 수 있는 방법 중 대표적인 기술이 그리드 컴퓨팅이다. 본 연구에서는 그리드 컴퓨팅 기술을 시설물 관리 분야 중 센서 네트워크 데이터 처리 분야에 적용하여 그 유용성을 확인하였다.

2. 그리드 컴퓨팅 개요 및 동향

그리드 컴퓨팅은 1990년대 미국 시카고 대학의 이안포스터(Ian Foster) 교수에 의해 처음 제안되었다. 이후 개념적 발전을 거듭하여 최근에는 인터넷을 통한 분산 컴퓨팅 자원의 공유 방식의 대용량 계산, 병렬 컴퓨팅을 활용한 컴퓨팅 능력 향상, IT 기반의 각종 기술 등 다양한 분야로 발전하고 있다(이성현 등, 2007). 그리드(GRID)의 원래 의미는 격자, 그물망, 방송망, 네트워크 등의 의미를 가지는 단어이지만, 물리적이나 지리적으로 분산된 컴퓨터 자원을 조직과 지역에 관계없이 단일 컴퓨터처럼 사용할 수 있는 기술을 의미하기도 한다. (wikipedia, 2009). 미국, 유럽, 일본, 호주 등 해외의 그리드 기술은 민간업체와 정부출연연구소를 중심으로 매우 다양한 분야에서 적용되고 있다. 미국은 1998년부터 다양한 그리드 프로젝트를 추진 하고 있으며, 특히 인간게놈지도 작성사업, 항공기 통합설계 등 첨단기술 개발을 분야에서 활발하게 사용하고 있다. 유럽에 경우, 유럽데이터 그리드, 유로 그리드 등 연구 시험망을 구축하고 있고, 일본은 APGrid, 글로벌 컴퓨팅 그리드 등 다양한 그리드 프로젝트가 진행 중에 있다. (김완석 등, 2004).

국내의 그리드 컴퓨팅 관련 연구는 다음과 같다. 2001년 5월 정보통신부와 한국과학기술정보연구원(KISTI)이 주관하여 국가 그리드 기본계획을 수립하였고, 2002년부터 국가 그리드 기반구축 사업과 그리드 미들웨어 연구가 진행 중이다(한국정보산업연합회, 2003). 또한 2002년 시작된 Korea@home 프로젝트가 국내의 대표적인 그리드 컴퓨팅 사례로 볼 수 있

다. 이러한 그리드는 컴퓨팅 분야 뿐만 아니라 타 분야에서도 연구가 진행 중이며, 대표적인 것이 스마트 그리드라고 불리는 지능형 전력망 사업이다. 스마트 그리드는 전력, 컴퓨팅, 통신, 소프트웨어, 가전, 반도체 등 다양한 기술요소가 복합적으로 얽혀있는 첨단 분야로서, 국내에서도 2009년 말까지 스마트 그리드 추진 로드맵을 만들고 있다(박찬국 등, 2009).

최근 유비쿼터스 컴퓨팅 기술이 활발하게 논의되면서, 유비쿼터스 센서 네트워크(USN) 분야에도 그리드 컴퓨팅을 적용하기 위한 연구가 진행되고 있는데, 그 중 하나가 본 연구에서 진행중인 센서 그리드 분야이다. 센서 그리드는 소형 센서노드들 간의 통신그룹인 센서 네트워크와 그리드 컴퓨팅이 결합된 분야로써, 두 요소 기술이 하나의 통합 플랫폼에 작용하여, 대용량 자료의 처리나 데이터 신뢰성 향상 분야에 적용하고자 하는 연구이다(Cben-Kbong Tbam 등, 2005; Rajkumar Buyya 등, 2005).

센서 네트워크 분야에 그리드 컴퓨팅을 적용한 대표적인 연구는 Cben-Kbong Tbam(2006), Aqeel-ur-Rehman(2008) 등이 있는데, 이는 센서 네트워크에서 수집된 데이터를 여러 대로 구성된 그리드 컴퓨팅을 이용하여 빠른 시간에 해석하는 연구로서, 본 연구와는 약간 차이를 보이고 있다. 또 국내에서도 오세진(2006) 등이 u-healthcare 분야의 센싱 데이터를 처리하기 위해 그리드 컴퓨팅 기술을 적용한 사례가 있으나 이 경우에도 데이터 처리를 다수의 그리드 컴퓨터를 이용하는 연구로 위의 Cben-Kbong Tbam(2006)사례와 비슷한 경우이다.

본 연구에서 제시한 그리드 컴퓨팅의 개념은 유비쿼터스 센서네트워크(USN)에서 한정된 컴퓨터 자원을 효율적으로 활용하기 위하여, 서버 컴퓨터와 게이트웨이간에 집중되는 부하를 효율적으로 분산하여 작업을 진행할 수 있게 하는 서버-USN 게이트웨이간 그리드 컴퓨팅이다.

도시시설물 관리 분야 그리드 아키텍처

본 연구는 도시정보시스템(UIS)에서 관리하고 있는 도시의 지상시설물을 효율적으로 관리하기 위해, 시설물에 각종 센서를 설치하여 이들의 상태를 실시간 모니터링 하기 위한 과제에서 출발하였다. 본 시스템은 U-GIS 기반의 도시시설물 관리시스템으로 센서에서 실시간 수집되는 정보를 GIS 데이터와 결합하여 시설물을 실시간 관리할 수 있는 시스템으로 전체 시스템 구성은 그림 1과 같다. 본 시스템은 오픈소스 GIS소프트웨어를 활용하여 도시시설물 관리 시스템을 구축하였다.

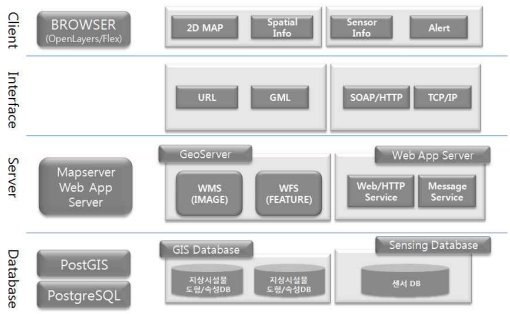

FIGURE 1. U-GIS 기반 시설물 관리시스템

그림 1과 같이 시스템을 구축한 후 센서에서 수집되는 센싱 데이터를 처리하기 위해 시스템을 운영할 경우, 센서의 수가 늘어날수록 서버에 집중되는 부하도 늘어나게 된다. 따라서 본 연구에서는 그리드 개념을 도입하여 서버에 집중되는 부하를 효율적으로 분산할 수 있는 기술을 적용해 보고자 연구를 진행하였다.

USN을 이용한 도시 시설물 관리 시스템 분야에 그리드 컴퓨팅 기술을 적용하는 단계는 그림 2와 같이 센서노드 단계, 미들웨어 단계, 통합 애플리케이션 단계, 서비스 레벨 단계 등 크게 4단계로 구분할 수 있다.

센서노드 단계에서는 센서 네트워크간 메쉬 네트워크(mesh-network)간 그리드 기술 적용

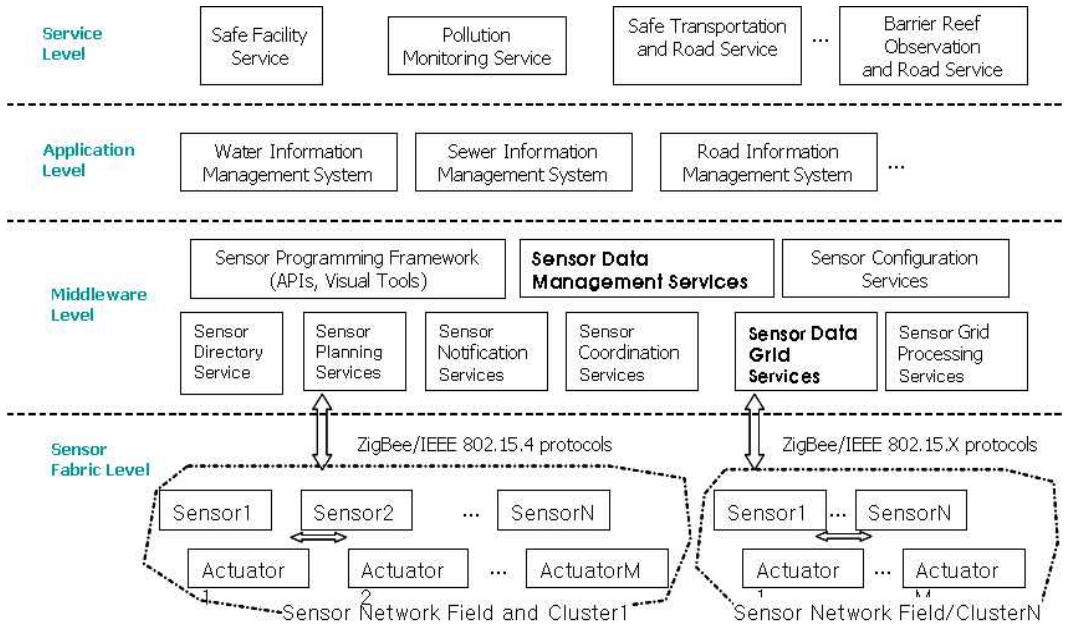

FIGURE 2. 도시시설물관리 GRID 적용 범위

이 가능하며, 미들웨어 단계에서는 그리드 컴퓨팅을 활용한 그리드 미들웨어 기술에 적용할 수 있다. 애플리케이션 단계에서는 글로버스 툴킷(Globus Tool Kit)을 활용한 CPU와 메모리 그리드 기술에 적용할 수 있으며, 서비스 레벨 단계에서는 서비스지향아키텍처(SOA ; Service Oriented Architecture)를 적용한 그리드 서비스 등에 적용할 수 있다.

본 연구에서는 센서노드 상단의 게이트웨이 단계인 미들웨어 단계에서의 센서 데이터 처리에 관한 그리드 아키텍처를 개발하여 u-GIS 기반 도시시설물 관리 분야에 적용할 수 있는 모델을 제시하고자 한다.

1. 시설물 관리분야 그리드 적용

u-GIS 기반 시설물 관리 시스템에서는 센서노드에서 수집되는 정보가 게이트웨이를 통해 데이터 수집 서버로 전달된다. 각 게이트웨이마다 관리하는 노드들은 일반적으로 그림 3과 같이 분포한다.

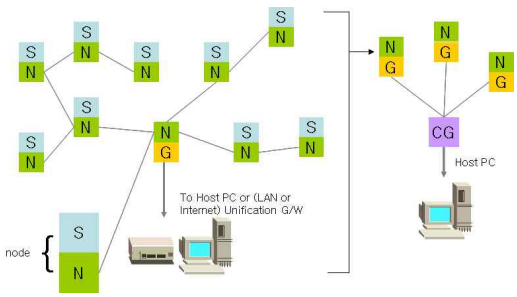

FIGURE 3. 센서 데이터 수집 구조도

S : Sensor, N : Sensor Node, G : Gateway

u-City 등 도시에 센서노드들이 설치된다면 다수의 센서노드와 이들을 관리하는 다수의 게이트웨이 그룹으로 배치가 되고 이들 정보가 서버로 실시간 전송되는 구조가 만들어진다. 센서노드 설치 시 게이트웨이의 용량을 고려하여 센싱 주기나 데이터 전송 주기를 설정할 수 있지만, 데이터를 수집하고 처리하는 서버의 경우에는 추가로 게이트웨이가 설치될 경우 시스템의 증설이나 장비의 증설이 필요하기 때문에 적절한 규모를 결정하기 어렵다.

본 연구에서 그리드 기술을 도입하려는 분야

는 대량 데이터 발생 시 게이트웨이와 u-GIS 서버간 '부하 분산(Load Balancing)'분야에 적용하는 것으로 한정하고, 추후 게이트웨이 고장 등 이상상황 발생 시 인접 게이트웨이로 통신 경로 재설정 분야와 센서네트워크 커버리지 재설정으로 유휴자원이 많은 게이트웨이로 '부하 분산' 등의 분야로 확대할 예정이다.

2. 그리드 아키텍처의 적용

본 논문에서 의미하는 u-GIS 서버란 GIS 기능을 포함하면서, 다수의 게이트웨이에서 수집하는 정보를 취합하고 처리하는 기능을 가진 서버를 뜻한다. u-GIS 서버의 역할은 센싱 데이터의 저장, 센싱 데이터의 분석(이벤트 감지), 이벤트 상황 처리, 센서의 아날로그 데이터 변환 등의 기능을 수행한다. 서버 자원의 사용량은 게이트웨이 개수에 비례하여 증가하는데, 처리 용량의 한계에 이르면 서버 운영이 중단이 된다. 일반적인 경우, 중단된 서버는 다시 구동시키거나 성능을 증가시키지 않으면 더 이상 작동하지 않고, 데이터의 수집 및 처리 기능도 더 이상 작동하지 않는다. 이러한 문제를 해결하기 위해 본 연구에서는 다수의 게이트웨이에서 수집서버로 데이터나 처리할 연산이 증가할 경우 그리드 서버에서 유휴 게이트웨이를 검색하여 서버 연산의 일부를 게이트웨이로 분산시키는 기능에 대해 연구하였다.

센서에서 수집되는 정보는 기본 원시데이터(Raw Data)는 아무 의미를 갖지 않는다. 원시데이터는 변환 테이블을 참조하여 변환 연산을 거쳐야만 의미있는 데이터가 생성된다. 이러한 변환 기능은 컴퓨팅 자원을 많이 요구하지 않지만, 수집된 데이터 양이 많아지면 서버에서 처리가 불가능할 수도 있다. 따라서 본 연구에서는 그리드 기능을 적용하여 유휴 자원이 풍부한 게이트웨이를 검색한 다음 서버의 부하를 분산시키는 기능을 개발하였다.

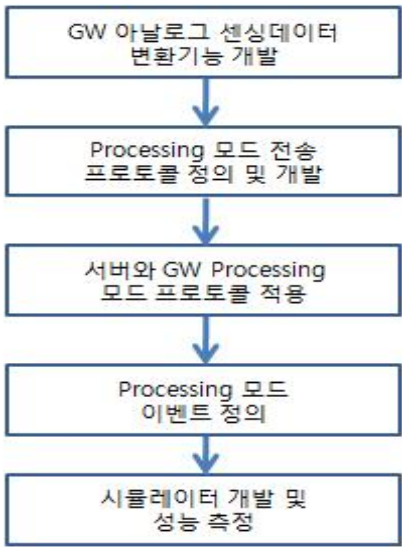

FIGURE 4. 그리드 아키텍처 연구순서

본 연구를 위해 그림 4와 같은 단위 모듈을 개발하였다. 게이트웨이에서 아날로그 형태의 센싱 데이터를 연산하여 처리할 수 있는 기능을 구현하고, 'Processing 모드'로 그리드 서버에 의해 변환될 수 있는 프로토콜을 정의하고 개발하였으며, 서버와 게이트웨이에 개발된 프로토콜을 적용하고, 그리드 서버에 의해 프로세싱 모드가 변환될 수 있는 이벤트 상황을 정의함으로서 최종적으로 이러한 상황에 대한 성능을 측정하기 위한 시뮬레이터를 개발하였다. 이렇게 개발한 시스템을 통해 기존 방법과 비교 연구를 수행하였다. 편의상 센서에서 수집되는 원시데이터를 아날로그 데이터로, 처리된 변환 데이터를 디지털 데이터로 정의하였다.

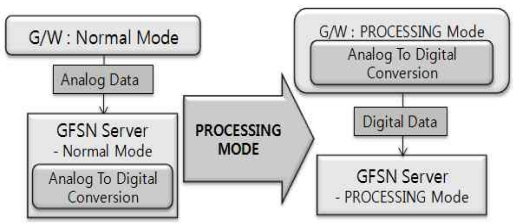

FIGURE 5. 서버에 그리드 적용

본 연구에서는 편의상 게이트웨이가 아날로그 데이터를 처리하는 연산모드를 'Processing 모드', 서버가 아날로그 데이터를 처리하는 연산모드를 'Normal 모드'라고 정의하였다.

그림 5는 'Normal 모드'에서 그리드 서버에 의해 'Processing 모드'로 변환되는 개념을 표현한 것이다. 좌측 그림이 'Normal 모드'인데, 게이트웨이에서 센서의 아날로그 데이터를 서버로 보내면 서버에서 연산을 통해 데이터를 변환하여 저장하는 작업을 수행한다. 우측편은 'Processing 모드'의 개념도이다. 서버의 부하가 증가할 경우 그리드 서버에서 유휴 게이트웨이를 찾아 게이트웨이에서 변환연산을 수행하고 처리된 결과를 서버로 바로 전송하여, 추가적인 작업 없이 데이터베이스에 저장하는 기능을 수행하도록 설계하였다

프로토콜 정의 및 개발

1. 아날로그 데이터 변환

본 연구에서 데이터 변환 시뮬레이션을 위하여 'Process 모드'에서 아날로그 데이터가 디지털 데이터로 변환 되는데, 사용한 변환식은 다음과 같다.

$$Sensor\ Value =$$

$$\left\{\frac{Smax - Smin}{ADmax - ADmin} \times (Adc\,Value - ADmin)\right\}$$

$$+ Smin$$

위의 수식에서 각 변수들의 의미는 다음과 같다.

$Smax$: 센서의 측정 최대 값
$Smin$: 센서의 측정 최소 값
$Sensor\ Value$: 실제 측정 값
$ADmax$: 최대 변환 값
$ADmin$: 최소 변환 값
$Adc\ Value$: 측정된 변환 값
$Smax$, $Smin$, $ADmax$, $ADmin$은 센서별로 제품 출하시 설정된 고유값이다. 이 값은 센서별로 데이터 시트 형태로 정의되어 있는데, 이 값을 활용하여 변환 연산을 수행하였다.

2. Normal 모드의 프로토콜 정의

'Normal 모드'는 게이트웨이에서 수집된 정보가 그대로 서버로 전송되고, 서버에서 연산을 통해 데이터 변환을 수행하고, 그 결과를 데이터베이스에 저장하는 모드를 의미한다. 'Normal 모드'의 진행 프로세스는 그림 6과 같다.

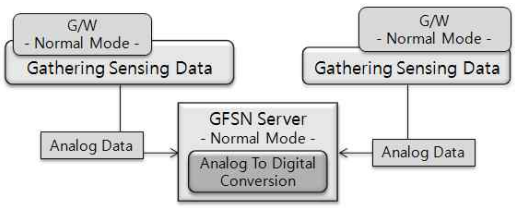

FIGURE 6. 수집서버 : Normal 모드

'Normal 모드'에서의 전송 프로토콜 규약은 표 1과 같다. 이 경우 센서 정보와 함께 측정된 데이터를 그대로 서버로 전송하게 된다.

TABLE 1. 서버의 Normal Mode 전송 프로토콜

```
<Head>
  <Command  Target="Sensor" Action="Current"
  NodeID=" {NodeID }" BoardID=" {BoradID}" >
  REQUEST
  </Command>
</Head>
<Data>
</Data>
```

본 연구에서 제시한 프로토콜은 표 1과 같은 형태로 구성된다. 'Normal 모드'와 'Processing 모드'의 시작은 <Head></Head> 사이의 <Command> </Command> 태그를 사용하여 구분한다. Command 태그의 값은 모드의 상태를 변경하는 Config와 데이터를 요청하는 Request로 구성된다.

게이트웨이에서의 'Normal 모드'에서의 전송 프로토콜은 표 2와 같다.

TABLE 2. GW의 Normal Mode 전송 프로토콜

```
<Data>
  <Info>
    <AnalogInput>
      < Channel  idx=1> {value} </ Channel >
                    ….
      < Channel   idx=4> {value} </ Channel >
    </ AnalogInput >
    <DigitalInput>
      < Channel   idx=1> {value} </ Channel >
                    …
      < Channel   idx=8> {value} </ Channel >
    </ DigitalInput >
    <DigitalOutput>
      < Channel   idx=1> {value} </ Channel >
                    …
      < Channel   idx=8> {value} </ Channel >
    </ DigitalOutput >
  </Info>
</ Data >
```

3. Processing 모드의 프로토콜 정의

'Processing 모드'에서의 기능은 서버의 부하가 집중되어 처리가 어려울 경우, 그리드 서버에서 이를 판단하여 유휴 게이트웨이를 찾은 다음 데이터 변환 연산을 게이트웨이에서 수행하는 것으로, 기능은 그림 7과 같다.

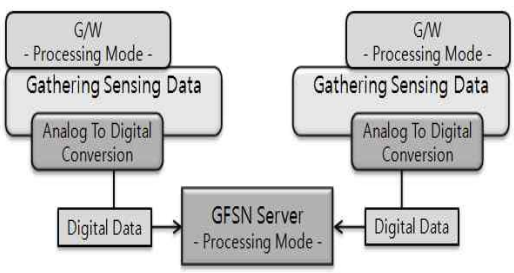

FIGURE 7. 수집서버 : Processing 모드

표 3은 'Processing 모드'에서의 서버의 전송 프로토콜 규약을 정의한 것이다. 미리 설정된 cpu의 임계치(본 연구에서는 40%로 설정하였다.)를 초과하는 경우 서버는 게이트웨이로 'Processing 모드'로의 변환을 명령하는 설정값을 Command 명령을 통해 Config 명령을 보내고, Data 태그를 통해서 게이트웨이 쪽으로 아날로그 데이터의 변환을 위한 센서의 각종 설정값을 보낸다. 이 설정값을 통해 게이트웨이에서 데이터 처리리를 시작한다.

TABLE 3. 서버의 Processing 모드 전송프로토콜

```
<Command  Target= "Process"  Action= "Start"
>
          CONFIG
</Command>
                        …
<Data>
  <Info>
    <AnalogInput  idx=" {value }" >
      <SensorName> {value } </SensorName>
      <MinValue> {value } </MinValue>
      <MaxValue> {value } </MaxValue>
      <ADMinValue> {value } </ADMinValue>
      <ADMaxValue> {value } </ADMaxValue>
      <Unit> {value } </Unit>
    </AnalogInput>
                        …
    <DigitalInput  idx=" {value }" >
      <SensorName> {value } </SensorName>
      <EventType> {value } </ EventType >
      <MessageHigh> {value } </ MessageHigh >
      <MessageLow> {value } </ MessageLow >
    </ DigitalInput >
                        …
    <DigitalOutput  idx=" {value }" >
      <SensorName> {value } </SensorName>
      <EventType> {value } </ EventType >
      <MessageHigh> {value } </ MessageHigh >
      <MessageLow> {value } </ MessageLow >
    </ DigitalOutput >
                        …
  </Info>
</Data>
```

표 3의 내용을 보면 Command 태그를 통해 명령을 전달하는데, 이 내용은 프로세스 모드로 변환하고 이를 시작하라는 내용의 명령을 서버에서 게이트웨이로 전송하게 된다. Info 태그는 센서 설정정보, AnalogInput 태그는 아날로그 센서 정보, DigitalInput은 디지털 센서

정보를 나타낸다. 표 3의 프로토콜을 통하여 'Processing'모드의 시작을 명령한다.

표 4는 'Processing 모드'일 경우 게이트웨이에서 데이터를 전송하는 프로토콜의 규약을 정의한 것이다.

TABLE 4. GW의 Processing 모드 전송 프로토콜

```
<Head>
                        …
  <Command  Target="Snesor"
          Action="ProcessingInfo"
          NodeID=" {NodeID }" BoardID="
          {BoradID}" >
          REQUEST
  </Command>
                        …
</Head>

<Data>
  <Info>
    <AnalogInput>
      < Channel  idx=1>
        <SensorName> {value }  </SensorName>
        <Value> {value } </Value>
        <Unit> {value }  </Unit>
      </Channel>
                        ….
    </AnalogInput>
    <DigitalInput>
      < Channel  idx=1>
        <SensorName> {value }  </SensorName>
        <Value> {value } </Value>
      </Channel>
                        …
    </DigitalInput>
    <DigitalOutput>
      < Channel  idx=1>
        <SensorName> {value }  </SensorName>
        <Value> {value } </Value>
      </Channel>
                        …
    </DigitalOutput>
  </Info>
</Data>
```

표 4는 게이트웨이에서의 전송 프로토콜을 표현한 것이다. Command 태그를 통해 센서 데이터를 처리하는 명령을 나타낸다.

그리드 서버 구현 및 시뮬레이터 개발

1. 그리드 서버 구현

서버의 'Processing 모드' 테스트를 위한 시스템 구성은 표 5와 같다.

TABLE 5. GRID 테스트 환경

구 분		내 용
Server (GRID)	CPU	Intel Quad CPU Q6600 2.4GHz
	MEM	3.25(4)GB DDR2 667 SDRAM
	OS	Windows XP Pro. sp3
	Dev.	JDK 1.6.0.04
	Tool	Eclipse 3.3
	DBMS	MySQL Community Server 5.0
Simulator (GW)	CPU	Intel Mobile Core Duo T7500 2.2Ghz
	MEM	2GB DDR2 667 SDRAM
	OS	Windows XP Pro. sp3
	Dev.	JDK 1.6.0.07
	Tool	Eclipse 3.4
	UI	SWT 1.4.2
공 통	Base Platform	NodeOne Gateway Platform v1.0
	Monitoring Tool	JConsole

서버(Server)는 u-GIS서버이며, 이 서버내에 그리드 컨트롤을 수행할 수 있는 그리드 서버 기능을 개발하여 내장하였다. 시뮬레이터(Simulator)는 게이트웨이를 가상화해서 소프트웨어적으로 구현한 것이다.

실제 성능 측정을 위해서는 게이트웨이를 물리적으로 연결하여, 게이트웨이의 개수를 100개 이상 늘려가면서 측정해야 하지만, 이는 현실 테스트 환경에서는 불가능한 일이다. 따라서 본 연구에서는 이런 게이트웨이의 역할을 수행할 수 있는 시뮬레이터 게이트웨이를 소프트웨어적으로 구현하였다.

2. 시뮬레이터 개발 및 실험

그림 8은 게이트웨이를 생성하기 위하여 개발한 시뮬레이터 화면이다. 하나의 시뮬레이터 메인 프로그램에서 100개씩 게이트웨이를 생성하였고, 그림 9에서는 101번에서 200번까지와, 301번에서 400번까지 게이트웨이를 생성한 화면이다.

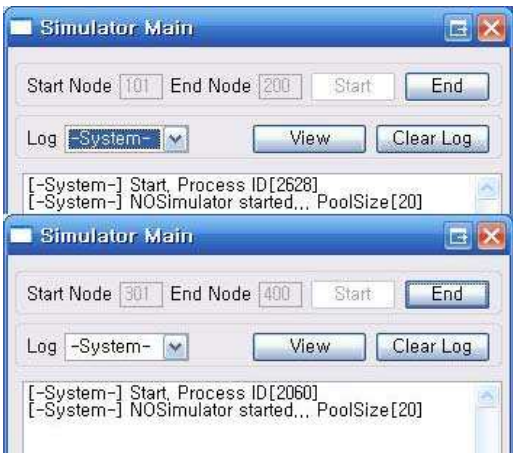

FIGURE 8. Simulator(400개 게이트웨이 연결)

이러한 창을 총 4개를 생성시켜 400개의 게이트웨이를 접속한 환경을 1차 실험하였다. 그림 10과 11과 같이 400개의 게이트웨이가 접속하였을 때 서버의 메모리 사용은 36.3Mb 이며, CPU의 사용량은 37.3%이다.

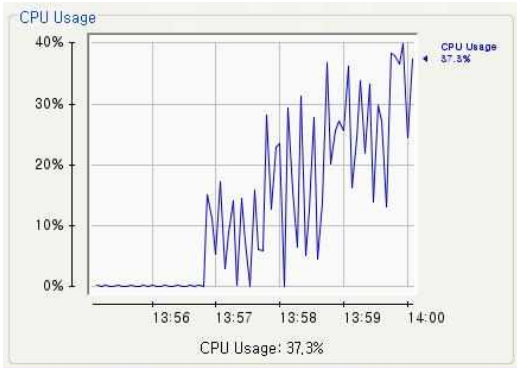

FIGURE 9. 400개의 게이트웨이를 연결하였을 경우의 CPU 사용량

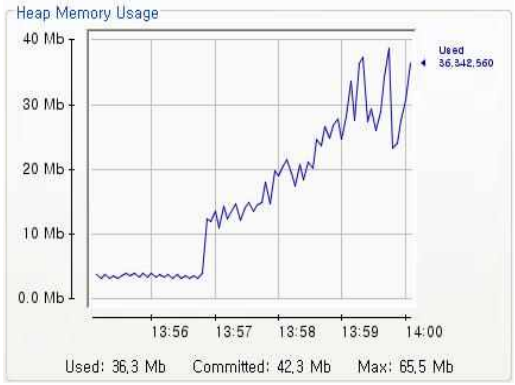

FIGURE 10. 400개의 게이트웨이를 연결하였을 경우의 메모리 사용량

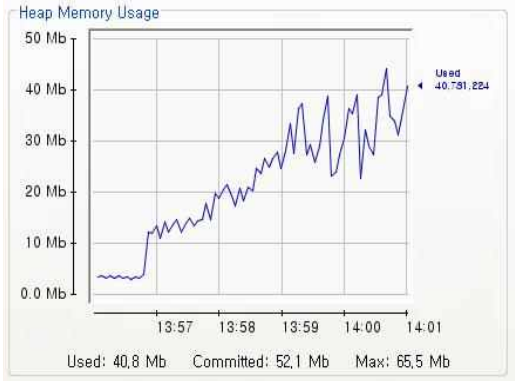

FIGURE 11. 500개의 게이트웨이를 연결하였을 경우의 메모리 사용량

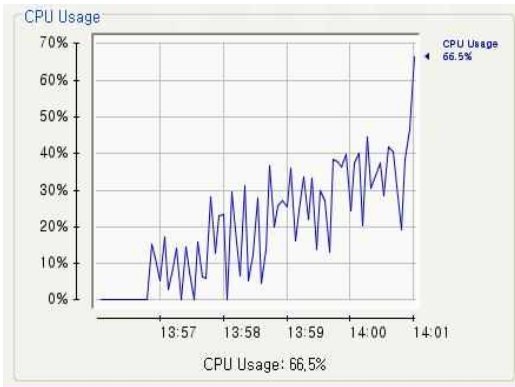

FIGURE 12. 500개의 게이트웨이를 연결하였을 경우의 CPU 사용량

본 실험에서 서버와 시뮬레이터를 가동하면서 'Normal 모드'에서 'Processing 모드'로 전환되는, 즉 그리드 서버가 가동을 시작하는 CPU의 부하 임계치는 40%로 설정하였다. 시뮬레이터를 통하여 게이트웨이의 개수를 100개씩 증가하여 CPU 점유율이 40%를 넘기는 시점을 모니터링 하였으며, 500개의 게이트웨이가 연결되었을 때 그림 12와 같이 서버의 CPU 점유율은 66.5%로 증가하였다.

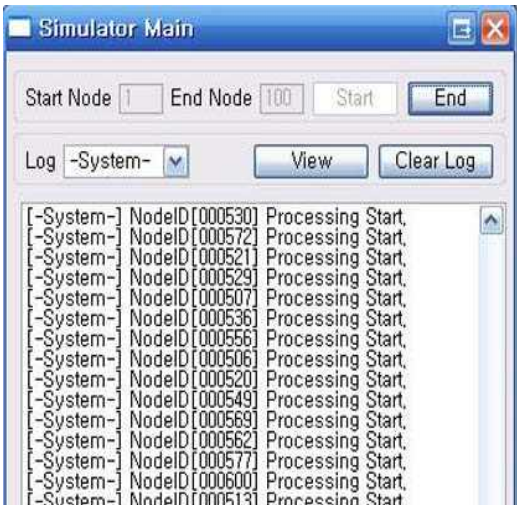

FIGURE 13. Processing 모드일 때의 콘솔화면

그림 12에서와 같이 CPU 사용량이 40%를 넘어서게 되면, 그림 13과 같이 콘솔화면에 'Processing Start' 문구가 표시되면서 어떤 노드가 'Processing 모드'로 변환되었는지 표시된다. 그림 13은 1~100개까지 게이트웨이를 그리드 서버에 연결시킨 시뮬레이터에서 각각의 게이트웨이가 'Processing 모드'로 연산처리를 하면서 서버로 데이터를 전송하게 된다.

그림 14는 'Processing 모드'로 전환 되는 순간의 서버의 CPU 사용량과 메모리 사용량을 나타낸 그래프이다. 이 그래프에서 보는 바와 같이 CPU 사용량이 66.5%에서 1.1%로 감소되며, 메모리 사용량은 40.8Mb에서 31.3Mb로 감소되었다.

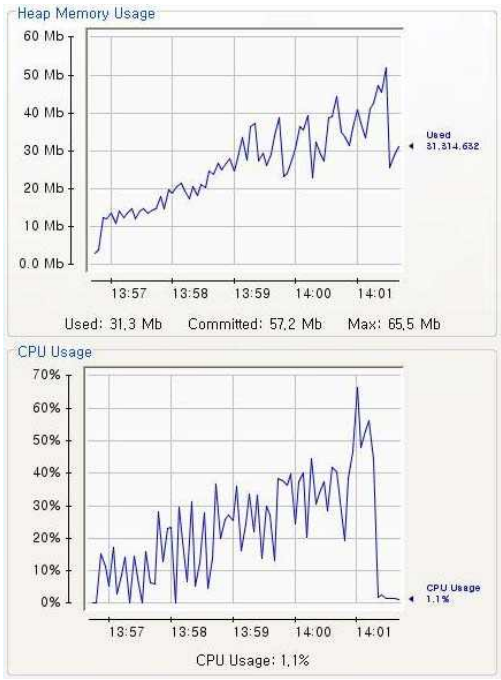

FIGURE 14. Processing 모드일 때의 메모리와 CPU 사용량

이 때 콘솔 화면에서는 다시 'Processing 모드'가 중지되고, 게이트웨이에서의 작업이 u-GIS 서버 쪽으로 넘어가게 된다. 즉 'Processing 모드'에서 일정 시간이 지나면 CPU 사용량이 임계치인 40%가 될 때까지 게이트웨이를 10개씩 'Normal 모드'로 전환하게 된다. 그림 15는 'Processing 모드'에서 CPU 사용량이 임계치가 될 때까지 사용량을 증가시키는 콘솔 화면이다.

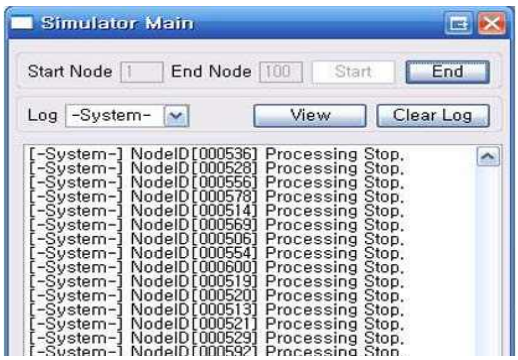

FIGURE 15. Normal 모드일 때의 콘솔화면

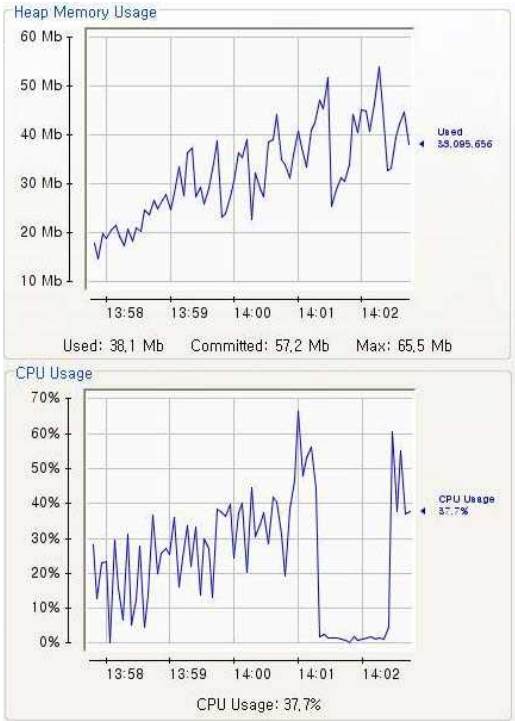

Figure 16. Normal 모드일 때의 메모리와 CPU 사용량

그림 16은 그리드 서버가 작동한 다음의 결과 화면인데, 'Processing 모드'에서 'Normal 모드'로 전환될 때의 콘솔창 화면과 서버의 자원 사용량을 보면, CPU의 사용량이 다시 60%까지 올랐지만 게이트웨이의 개수를 조정하여 미리 설정된 임계치 40% 미만으로 고정되어 서버가 작동하고 있음을 알 수 있다. 즉 게이트웨이의 수가 증가하더라도 CPU 사용량 임계치 이상으로 높아지지 않도록 유지하면서 서버를 운영할 수 있는 것이다.

결 론

본 연구에서는 u-City 도시 시설물 관리를 위한 USN 기반 시설물 관리 시스템에서, 서버와 게이트웨이에 그리드 기술을 도입하여 센서와 게이트웨이의 수가 증가하더라도, 서버가 다운되지 않도록 '부하 분산'시켜 운영될 수 있는 기술을 개발하고 이를 시뮬레이터를 활용하여 분산을 통해 서버의 부하량 조절을 자동으로 하는 실험을 하였다. 본 연구의 실험 결과를 보면 게이트웨이의 수가 일정 수량을 넘어서 늘어날 경우 서버의 CPU 사용량이 계속 증가하는데, 이 때 그리드 서버에서 접속되는 부하를 다시 유휴 게이트웨이쪽으로 전송시켜 서버를 유지시킬 경우 서버의 부하는 감소시키면서 기능은 그대로 유지할 수 있다는 것이 확인되었다. 센서나 게이트웨이의 수가 늘어남에 따라 물리적인 서버의 성능을 늘리는 것도 필요한 일이지만, 본 연구의 결과를 활용하면 컴퓨터의 남는 유휴 자원을 효율적으로 활용하여, 친환경적이고 자원을 절약할 수 있는 시스템을 구축할 수 있을 것이다. 향후 개발되는 시스템은 센서노드 자체에서 데이터를 필터링하고 1차 가공하여 데이터를 전송하는 기술이 도입될 경우 이를 반영하여 그리드 시스템 개발하는 추가 연구가 진행될 필요가 있다. KAGIS

참고 문헌

권혁종, 김장욱, 남상관, 김태훈, 임성모. 2009. GRID를 적용한 USN 관리시스템에 관한 연구. 2009 측량학회 춘계학술발표회. 419-427쪽.

김완석, 김정국. 2004. 그리드 기술 연구 동향. 주간기술동향 1134:1-14.

김의명, 이윤, 김성수, 김인현, 최영희. 2006. 무선인식과 지형공간정보체계를 이용한 효율적인 가로수관리. 한국지리정보학회지 9(1):137-148.

박찬국, 용태석. 2009. 스마트 그리드 도전과제와 추진방향. 주간기술동향 통권14(11):1-13.

오세진, 이채우. 2008. 센서 네트워크와 그리드 네트워크와의 연동을 위한 u-Healthcare 센서 그리드 게이트웨이 설계 및 구현. 대한전자공학회지 49(6): 64-72.

이성현, 이재승, 문기영. 2007. 웹서비스 기반 그리드 보안 기술 동향. 전자통신동향분석 22(1):24-36.

이필우, 황일선. 2003. 그리드(GRID). TTA 저널 (제89호) 표준 기술동향 130-136.

전문장, 심규원, 김민. 2009. 과수의 생육환경 모니터링을 위한 유비쿼터스 센서 네트워크의 적용 -사과나무를 대상으로-. 한국지리정보학회지 12(3):56-65.

한국정보산업연합회. 2003. 그리드컴퓨팅 민간백서.

Aqeel-ur-Rehman, Zubair A. Shaikh. 2008. Towards design of context-aware sensor grid framework for agriculture. World Academy of Science, Engineering and Technology 38:244-247.

Chen-Khong Tham. 2006. Sensor-Grid computing and SensorGrid architecture for event detection, classification and decision-making. In: N.P Mahalik(ed). Sensor Network and configuration: Fundamentals, Techniques, Platforms and Experiments. Spirnger-Verlag, Germany. 1-16pp.

Chen-Khong Tham and Rajkumar Buyya. 2005. SensorGrid: integrating sensor networks and grid computing. CSI Communications 29(1): 24-29.

Rajkumar Buyya. 2005. SensorGrid: A New Cyberinfrastructure Integrating Sensor Network and Grid Computing for e-Science Applications. National Workshop on Wireless Sensor Networks.

http://www.wikipedia.org. 2009. 10