

지능형 관제시스템 개발

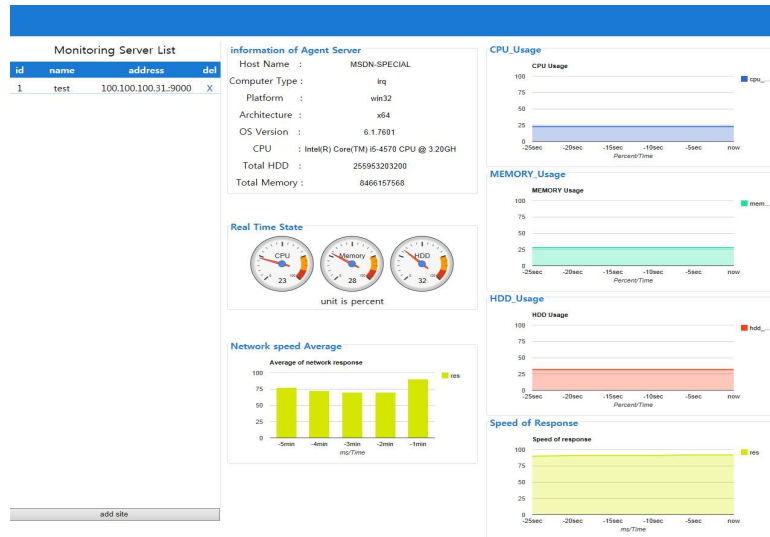
2014년 06월

우석대학교 산학협력단

개발 결과의견서

개발과제명	지능형 관제시스템 개발						
개발기간	2013년 07월 01일 ~ 2014년 06월 30일						
사업비	직접비(예산)	직접비 중 집행 금액			주관기관 집행금액	참여기업 집행금액	집행잔액
		계	현금	현물			
	62,879,000	62,682,110	47,282,110	15,400,000	39,382,110	23,300,000	196,890
주관기관 과제책임자	소속 : 우석대학교 산학협력단 직위(급) : 교 수 성명 : 천 은 홍						
개발결과 (과제책임자 작성)	가) 과제개발 실적(요약 작성) <ul style="list-style-type: none"> ○ 관제 대상의 컴퓨터 상세 정보 및 서버의 운영 현황 모니터링 시 장애등급별로 구별 가능 ○ 서버 CPU 부하 및 SWAP 분석을 위한 성능 수치 그래프 표현 ○ 다수의 연결을 효율적으로 관리하고 비용을 최소화 할 수 있는 Node.js를 이용하여 관제서버 구축 ○ 1초마다 로그 메시지를 지속적으로 출력하여 실시간으로 확인가능 ○ 성능 저하의 원인이 되는 프로세스를 식별하여 프로세스별로 디스크 I/O 사용 수치 조회 기능 구축 ○ 사용자편리성을 고려한 인터페이스 화면 						
	나) 특허(실용신안, 의장 등)출원 및 등록 : 4건 (프로그램 등록 2건) <ul style="list-style-type: none"> ① 출원명 및 번호(출원일자) : <ul style="list-style-type: none"> - 지능형 관제 시스템 및 방법/10-2013-0119404 (2013.10.07.) - 화상디자인이 표시된 컴퓨터 모니터/30-2014-0031495 (2014.06.26.) ② 등록명 및 번호(등록일자) : <ul style="list-style-type: none"> - 노드관제모니터링프로그램/C-2014-000831 (2014.01.13.) - 노드관제모니터링프로그램v2.0/C-2014-015066 (2014.06.26.) ③ 논문 발표(발표일자) : <ul style="list-style-type: none"> - 서버 관제 시스템 과부하 방지 알고리즘 제안(2014.05.07) <p>* 지식재산권명 및 출원 등록 구분은 명확히 하고 건별로 모두 작성</p>						
	다) 시제품 제작 및 상품화 실적 <ul style="list-style-type: none"> ○ 지능형 관제시스템 개발 ○ 기업 전문 엔지니어를 활용한 통합화된 양산 체제 구축 <ul style="list-style-type: none"> - 분 석 : 내부 양산 활동 (프로세스, 산출물 검토) - 개 발 : 납품 및 설치, 규격/사양 검사, 단위 및 통합시험 						

- 배 포 : 사용자 및 운영자 교육, 유지보수 계획 수립
- 유지보수 : 안정적 운영지원, 무상유지보수
- 인터넷 및 전자신문, 정보통신 장비로 광고 게재 및 방송 매체 홍보
- 제품 홍보 광고, 마케팅 시장성 분석
- 회사 인프라를 통한 네트워크 장비 개발 업체와의 제휴를 통해 영업망 및 판로 구축



라) 기타 실적(신기술로 인정될 수 있는 각종 품질인증 마크 획득건수 및 내용)

◎ 개발목표 달성도에 대한 종합의견(참여기관 대표가 작성)

개발목표

- 실시간 장애분석 기능을 이용한 관제 프로그램 개발
- 사용자 중심의 인터페이스 화면을 이용한 도식화 화면 개발
- 관제프로그램 이원화 구축 및 통합 관제 서버 개발

개발목표달성에 대한 종합의견 (□에 V로 표기하고 의견 작성)

☒ 매우 만족 ☐ 만족 ☐ 보통 ☐ 미흡 ☐ 불만족

○ 개발된 지능형 관제시스템은 서버 시스템 및 네트워크 규모가 커지고 복잡해짐에 따라, 정보통신 인프라 자원에 대해서 효율적인 운영 및 위험관리를 위하여 새로운 개념의 기술적 관리대책이 절실히 요구되어지고 있는 시점에 실시간 장애 분석 기능을 이용한 서버 관제 프로그램을 도입하여 효율적인 업무운영을 도모할 수 있으며 다양하게 분산된 각종 IT자원을 효과적으로 관리가 가능하며 서버의 장애발생 시 서비스가 정상적으로 진행되는지의 여부를 실시간으로 관제하여 서버 다운이 발생하면 운영자에게 미리 등록된 연락처를 통해 SMS를 발송으로 인해 서버 장애에 대해 신속히 대처가 가능한 최적의 솔루션이며 추후 다양한 분야에서 활용이 가능하며 이로 인해 일자리 창출 등의 많은 파생효과가 나타날 것으로 생각됨

최종보고 요약서

과 제 명	지능형관제시스템 개발		주관기관명	우석대학교 산학협력단		
참여기업명	(주)솔루션뱅크		대표자	문 영 희		
과제책임자	소 속		직 위	성 명		
	우석대학교 산학협력단		교 수	천 은 홍		
협약기간	2013년 07월 01일 ~ 2014년 06월 30일					
사 업 비 (단위 : 원)	직접비 (예산)	집행 내역			집행잔액	집행비율 (%)
		계	현금	현물		
	62,879,000	62,682,110	47,282,110	15,400,000	196,890	99

1. 최종목표

- 실시간 장애분석 기능을 이용한 관제 프로그램 개발
- 사용자 중심의 인터페이스 화면을 이용한 도식화 화면 개발
- 관제프로그램 이원화 구축
- 통합 관제 서버 개발

2. 개발내용 및 결과

- 다양한 서비스 장애 인식 기능
- 실시간 성능 감시 기능
- 다양한 모니터링 관리화면 제공
- 시스템 비교 분석 기능
- 실시간 Web Log 파일 실시간 추적 분석 기능

3. 사업성과

- 기술적 성과

개발 프로그램은 서버 모니터링 시스템이다. 서버 모니터링 시스템은 현재 운영 중인 서버의 CPU, HDD, RAM, NETWORK의 부하율, 상태를 모니터링 함으로써 서버의 이상 현상 발생을 미리 방지하고, 이상 현상이 발생하였을 시 원인을 분석, 문제에 따른 해결 방안을 마련하기 위해 사용한다. 서버 모니터링을 하기 위해 모니터링 대상이 되는 에이전트 프로그램과 모니터링을 실시할 서버 프로그램을 구성함

- 경제적 성과

구 분	사업화 년도		
	(2014)년 (개발종료 해당년)	(2015)년 (개발종료 후 1년)	(2016)년 (개발종료 후 2년)
사업화 제품	지능형 관제 시스템	지능형 관제 시스템	지능형 통합 관제 시스템

투자계획(백만원)		100	200	400
판매 계획 (백만원)	내 수	500	800	1500
	수 출	0	200	500
	계	500	1,000	2,000
수입대체효과(백만원)		500	1,000	2,000
고용 창출(명)		2	3	5

○ 사회적 성과(일자리 창출 등)

시스템 및 네트워크 규모가 커지고 복잡해짐에 따라 정보 통신 인프라 자원에 효율적인 운영과 위험관리를 위하여 새로운 기술적 관리 대책이 절실히 요구되어지는 추세이다. 본 사업을 통해 개발된 지능형 관제 시스템으로 인하여 효율적인 관리, 실시간 장애분석, SMS 서비스 등을 통하여 서버를 효율적인 업무 운영이 가능하며 사용자를 고려한 인터페이스를 통하여 편리성 및 다양한 분야에서의 활용이 가능하며 이로 인하여 관련 IT산업의 발전과 동시에 관련 서버 분야의 일자리 창출에도 큰 기여효과를 나타낼 수 있음

4. 기술개발결과 활용계획

- 서버 과부하 원인 프로세스를 식별하기 위한 프로세스 별 프로세스 명, CPU 점유율, 메모리시스템 자원 점유율 정보 제공
- 신뢰성(Reliability) 인증 확보 계획
- 제품 양산 계획
- 판로 확보 및 마케팅 계획
 - 지능형 관제 프로그램의 가격, 제품, 커뮤니케이션 등 크게 세 가지 측면에서 각각 마케팅 톨을 도입
 - 전산센터 및 관련분야 실사용자 중심으로 제품 개선방향에 대한 수요조사를 통한 제품 업데이트
 - 온라인/오프라인 등 다양한 매체를 통한 제품 홍보

5. 기술개발자료의 보안관리

- 보안등급 분류기준에 따른 등급부여 및 관련 의견 기술
- 연구개발의 기술임치 및 특허 출원·등록 실적
 - 지능형 관제 시스템 및 방법(특허출원 : 10-2013-0119404)
 - 화상디자인이 표시된 컴퓨터모니터(특허출원 : 30-2014-0031495)
 - 노드 관제 모니터링 프로그램 등록 v1.0(등록번호 : C-2014-000831)
 - 노드 관제 모니터링 프로그램 등록 v2.0(등록번호 : C-2014-015066)

최종보고서 작성순서 및 목차

① 과제별 최종보고서 표지(제출서)

② 최종보고 요약서

③ 본문

제 1 장 서론

제 1 절 기술 개발의 필요성 및 현황

제 2 절 기술 개발 목표

제 2 장 과제개발 내용 및 방법

제 1 절 과제개발 내용

제 2 절 연구개발 추진 일정

제 3 절 기관 별 역할 분담

제 3 장 사업성과

제 1 절 기술적 성과

제 2 절 경제적 성과

제 3 절 기타 성과

제 4 장 결론

제 1 절 기술개발결과 활용계획

제 2 절 기대 효과

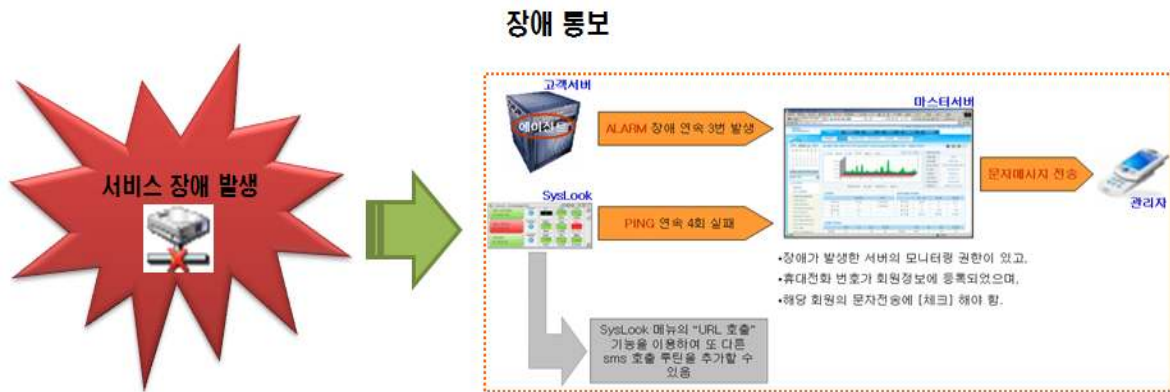
*별첨

제1장 서론

제1절 기술개발의 필요성 및 현황

- 본 개발은 시스템 및 네트워크 규모가 커지고 복잡해짐에 따라, 정보통신 인프라 자원에 대해서 효율적인 운영 및 위험관리를 위하여 새로운 개념의 기술적 관리 대책이 절실히 요구되어지고 있는 시점에 실시간 장애 분석 기능을 이용한 서버 관제 프로그램을 도입하여 효율적인 업무운영을 도모하고자 함
- 고객 서비스 향상을 위하여 기업의 IT기술과 전산자원이 급격히 증가함에 따라, 이를 지원하는 서버와 네트워크 장비, 어플리케이션들을 효과적으로 관리하기 위해서는 많은 노력이 필요 해졌지만 일정 부분에 관리 툴을 적용했으나 총체적인 업무현황 파악이나 체계적 관리가 및 장애감지가 쉽지 않으며, 신속하지 못한 대응으로 고객의 항의를 받는 경우가 빈번하게 발생함. 다양하게 분산된 각종 IT자원을 효과적으로 관리할 수 있는 최적화된 솔루션의 개발이 요구됨
- 서버관련 시스템, 모니터링 및 보안시스템 등을 제공하는 것은 고가의 비용이 필요한 별도의 서비스 영역이었으며, 기존 서버호스팅 서비스에서의 서비스 영역은 인터넷 회선과 서버 공간을 제공하는 것이 기본적이었고 관리서버에서만 운영되고 있어 관제 서버가 다운시 관제를 할 수 없으므로 관제서버와 운영서버에서 동시에 관제를 할 수 있도록 업그레이드가 필요함
- 사용자가 사용하고 있는 서버 시스템에 대해 트래픽 현황 및 운영 현황을 직접 관리할 수 있도록 제공하는 서비스이며, 한 개의 브라우저에 5개까지 모니터링이 가능하고 관제 프로그램의 코딩 수정 없이 관리자 화면에서 서비스 모니터를 추가하거나 삭제시 관리프로그램 내에서 설정이 가능. 주 서버의 장애발생시 서비스가 정상적으로 진행되는지의 여부를 실시간으로 관제하여 서버 다운이 발생하면 운영자에게 미리 등록된 연락처를 통해 SMS를 발송하여 알려줌

제2절 기술개발 목표



[그림 2] 관제 시스템 개발 예시

- 본 개발은 시스템 및 네트워크 규모가 커지고 복잡해짐에 따라, 정보통신 인프라 자원에 대한 보다 효율적인 운영관리 및 위험관리를 위하여 새로운 개념의 기술적 관리대책이 절실히 요구되어지고 있는 시점에 서버 관제 프로그램을 도입하여 효율적인 업무운영을 도모
- 실시간 장애분석 기능을 이용한 관제 프로그램 개발
 - 서버 시스템에 대해 트래픽 현황 및 운영 현황을 직접 관리할 수 있도록 개발
 - 한 개의 브라우저에 5개까지 모니터링 시스템 개발
 - 관제 프로그램 코딩수정 없이 관리자 화면에서 설정하는 관리프로그램 개발
 - 모니터링 시스템 수 자동증가 구축
 - 연동 SMS환경 구축
- 사용자 중심의 인터페이스 화면을 이용한 도식화 화면 개발
 - 통계, 성능 분석, 오류분석 다양한 용도로 사용
 - 운영자에게 효율적인 시각화로 표현
- 관제프로그램 이원화 구축
 - 관제서버 오류시 서비스 운영 서버에서 운영자에서 연동된 SMS 알림 기능



[그림 3] SMS서비스 개념도

○ 통합 관제 서버 개발

- 데이터베이스 및 HTML5 기반의 애플리케이션 서비스, 저장 및 스트리밍을 하나의 서버로 지원하는 통합 서버 개발
- 임베디드 리눅스 기반의 저전력 서버 시스템 개발
- 주요 H/W 모듈

제2장 과제개발 내용 및 방법

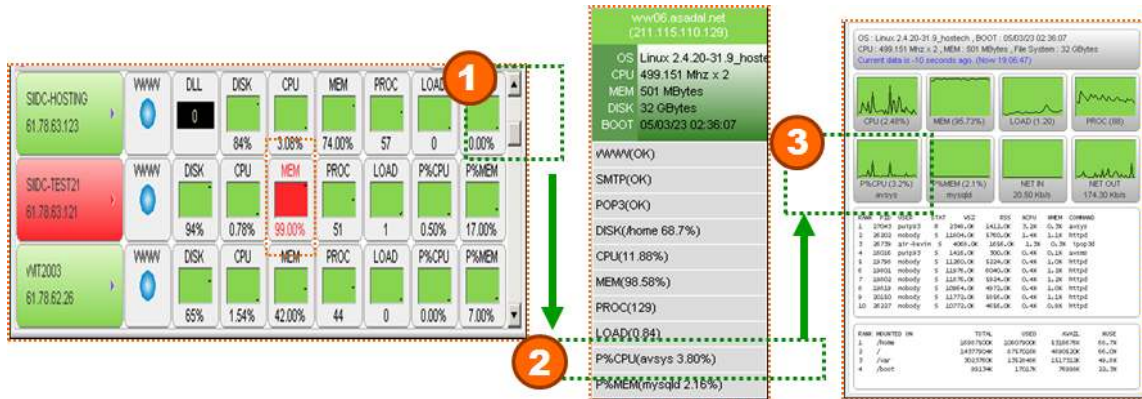
제1절 기술개발 목표 및 내용

- 다양한 서비스 장애 인식 기능
 - 지능형 관제 시스템의 24시간 365일 무장애 시스템의 구현을 목적으로 서비스 프로그램의 정상작동 여부를 판단해 주는 최종 장애 여부 인식 기능
 - ▶ Port Monitoring : 실제 통신포트로 접근하여 정상적인 응답을 확인
 - ▶ Process Monitoring : 실행중인 프로세스의 정상작동 여부를 확인
 - ▶ Log Monitoring : 서비스 프로그램이 남기는 다양한 로그파일을 분석하여 장애 여부를 확인



[그림 4] 다양한 서비스 장애 인식 기능

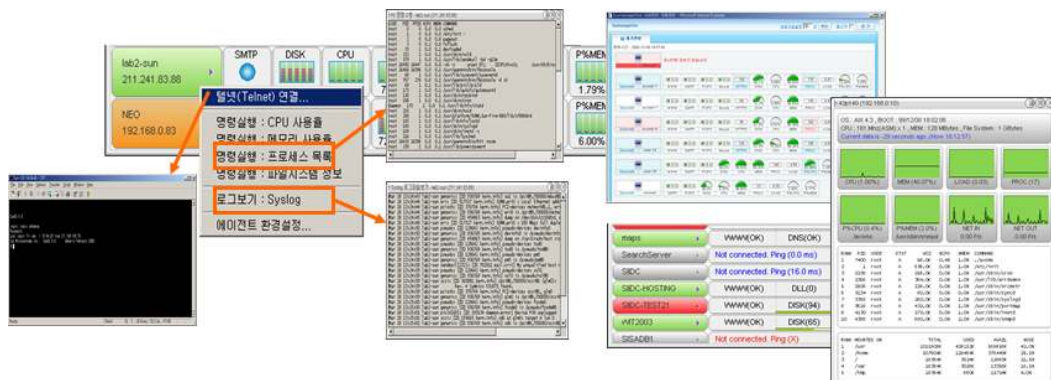
- 실시간 성능 감시 기능
 - 시스템 관리자가 시스템의 향후 위험상황을 쉽게 예측 할 수 있도록, 현재 시스템의 주요 지표(CPU, RAM 사용률, 로드, 파일시스템, 프로세스 및 특이사항을 보이는 프로세스 등)를 실시간 알림 기능
 - ▶ DISK : 여러 파일시스템 중 가장 사용률이 높은 파일시스템의 정보를 통보
 - ▶ CPU : 실행중인 여러 프로세스 중 CPU를 가장 높게 사용하는 프로세스 알림
 - ▶ MEM : 실행중인 여러 프로세스 중 메모리를 가장 많이 사용하는 프로세스를 알림



[그림 5] 실시간 성능 감시 기능

○ 다양한 모니터링 관리화면 제공

- 시스템 관리자가 시스템의 향후 위험상황을 쉽게 예측 할 수 있도록, 현재 시스템의 주요 지표(CPU, RAM 사용률, 로드, 파일시스템, 프로세스 및 특이사항을 보이는 프로세스 등)를 실시간 모니터링 기능

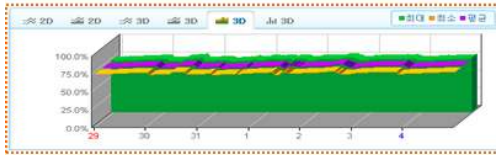


[그림 6] 다양한 모니터링 관리 화면

○ 시스템 비교 분석 기능

- 시스템 대부분의 성능지표를 보여주는 성능 데이터 화면 구성
 - ▶ CPU/메모리/프로세스/IO/DISK 뿐만 아니라 Inbound/Outbound 트래픽 데이터 제공
 - ▶ 다른 서버의 성능정보와 비교할 수 있는 비교 그래프
 - ▶ 다양한 그래프 (2D/3D)
 - ▶ 업무시간대별, 시간대별, 날짜별 비교 등 다양한 분석 보고서 제공
 - ▶ 특정 프로세스 및 사용자 추적분석 기능

<시스템 비교>



<비교 데이터>

요약정보			업무요청별 요약정보			
성능정보		TIME	월 - 금		토요일	일요일
최대 값	76	10/30 17:00	최대 값	76	77	75
최소 값	71	10/30 01:00	최소 값	71	73	72
평균 값	73.75		평균 값	73.77	74.23	73.17
단 위	%		점 유 율	33.35%	33.56%	33.08%

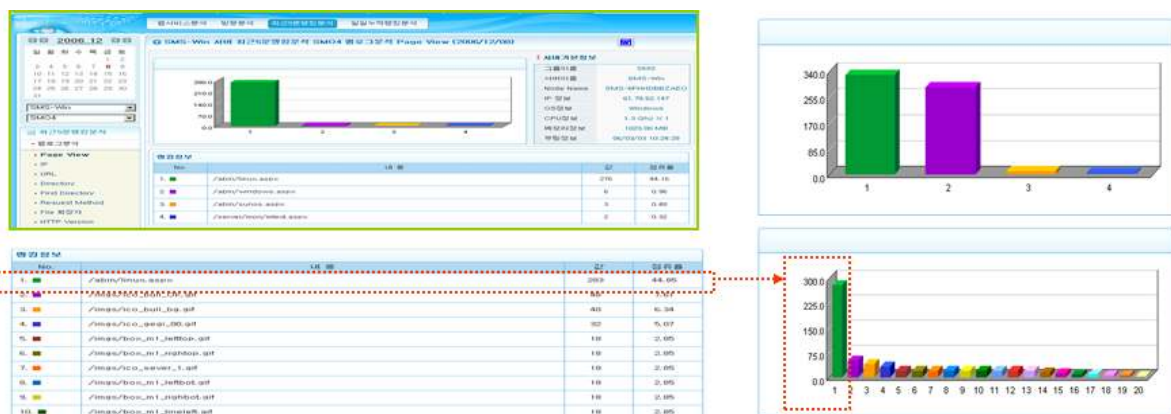
<시스템 성능 분석>



[그림 7] 시스템 성능 분석 기능

○ 실시간 Web Log 파일 실시간 추적 분석 기능

- 웹서비스를 제공하는 서버의 시스템 관리자의 웹로그 파일은 매우 중요 파일로 이용되고 있으며, 지속적으로 변경되는 웹로그 파일을 실시간으로 추적 및 분석하여 실시간 모니터링 상태를 제공
 - ▶ 다양한 항목의 실시간 분석결과 제공(CommonLog부터 RefererLog까지 모두 분석 가능)
 - ▶ 웹 서비스 Load Balancing 상태를 그래프로 비교하여 실시간으로 볼 수 있음
 - ▶ 웹 프로세스 추적 분석 기능이 기본으로 채택되어 실시간 결과 제공
 - ▶ 관리자가 현재의 서비스 상태를 확인이 용이 하도록 세부항목 분석 제공



[그림 8] 실시간 Web Log 파일 실시간 추적 분석 기능

제2절 연구개발 추진 일정

일련 번호	세부 개발내용	수행 주체	세부 추진 일정 (개월)												비 고
			7	8	9	10	11	12	1	2	3	4	5	6	
1	선행연구 조사 및 개발환경 구축	주관/ 참여													
2	관계 관리 프로그램 구성도	주관													
3	자동설치 프로그램 작성	참여													
4	제어를 위한 유무선 통신 인터페이스 개발	주관													
5	도식화 화면 개발	참여													
6	데이터베이스 연동 및 어플리케이션 개발	주관													
7	지능형 장애 대응시스템 개발	참여													
8	시작품 개발 및 성능평가	주관/ 참여													
9	시험인증	주관/ 참여													
10	논문, 특허	주관/ 참여													

제3절 기관 별 역할 분담

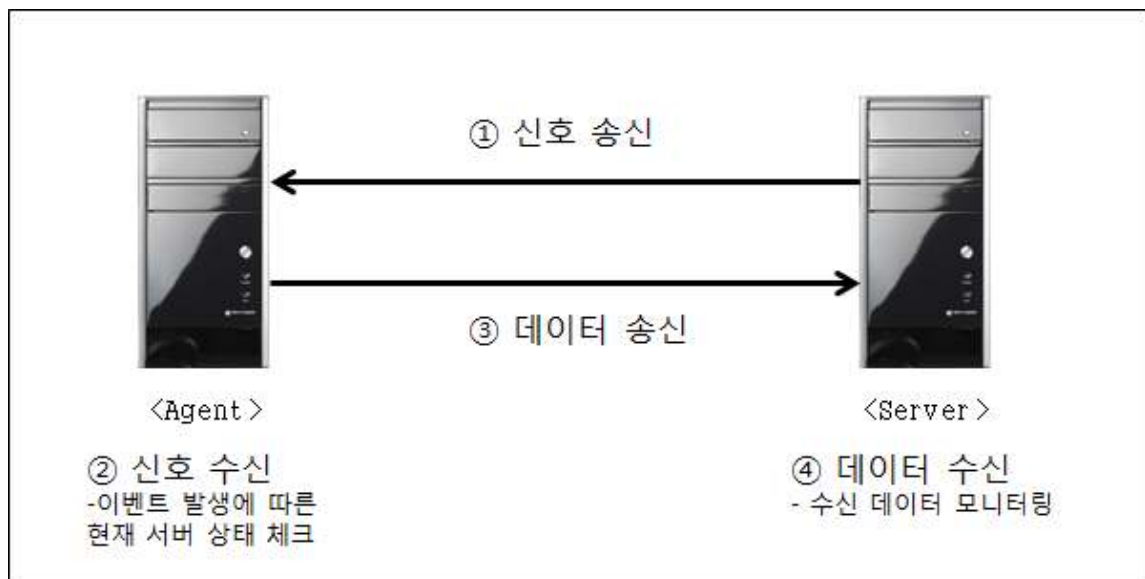
수행기관	주요 담당 업무	기술개발 비중(%)
주관기관	<ul style="list-style-type: none"> 선행연구 조사 및 개발환경 구축 관계 관리 프로그램 구성도 제어를 위한 유무선 통신 인터페이스 개발 데이터베이스 연동 및 어플리케이션 개발 시작품 개발 및 성능평가 논문, 특허 	50%
참여기업	<ul style="list-style-type: none"> 선행연구 조사 및 개발환경 구축 자동설치 프로그램 개발 도식화 화면 개발 지능형 장애 대응시스템 개발 시험인증 	50%
총 계	<ul style="list-style-type: none"> 지능형 관계 시스템 개발 	100%

제3장 사업 성과

제1절 기술적 사업성과

1. 지능형 관제프로그램 개발 개요

- 개발 프로그램은 서버 모니터링 시스템이다. 서버 모니터링 시스템은 현재 운영 중인 서버의 CPU, HDD, RAM, NETWORK의 부하율, 상태를 모니터링 함으로써 서버의 이상 현상 발생을 미리 방지하고, 이상 현상이 발생하였을 시 원인을 분석, 문제에 따른 해결 방안을 마련하기 위해 사용한다. 서버 모니터링을 하기 위해 모니터링 대상이 되는 에이전트 프로그램과 모니터링을 실시할 서버 프로그램을 구성하였다. 프로그램의 구동 원리는 다음과 같다.



[그림 9] 서버 모니터링 시스템 구동 원리

① 신호 송신

모니터링 서버는 등록된 에이전트 서버로 주기적 신호를 송신한다.

본 개발에서는 신호 주기를 5초(5000ms)로 지정하였다.

② 신호 수신

모니터링서버에서 송신한 신호를 에이전트서버에서 받았을 때, 에이전트 서버에는 이벤트가 발생하게 된다. 이벤트 발생 시 수행하는 일은 서버의 상태를 체크하게 된다.

③ 데이터 송신

에이전트 서버에서 체크한 서버상태를 모니터링 서버로 송신한다.

④ 데이터 수신

에이전트 서버로부터 수신한 데이터를 가공하여 디스플레이 한다. 모든 데이터는 그래프화 하여 표현한다.

2. 개발 언어 설치 및 동작 방법

1) 프로그램 개발 툴

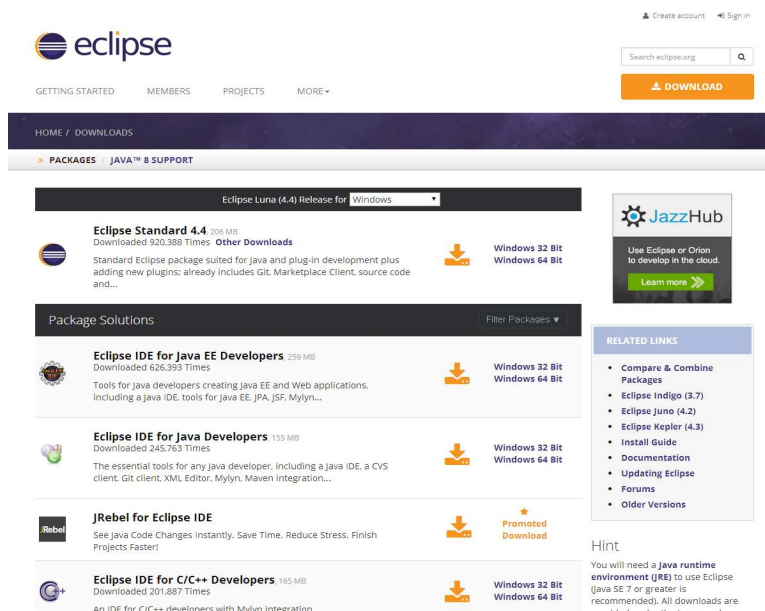
본 프로그램은 웹 어플리케이션으로 개발하였다. 지금까지 개발되어 왔던 프로그램과 같이 PHP, ASP, JSP 등의 서블릿 언어가 아닌 서버사이드 언어 Node.js를 이용하여 개발하였다.

- 개발 언어 : JavaScript, Node.js, HTML5, mysql, CSS
- 개발 에디터 : eclipse, editplus

2) 개발 툴 설치

- eclipse 설치

- ① <http://www.eclipse.org/downloads/> 접속
- ② eclipse 다운로드



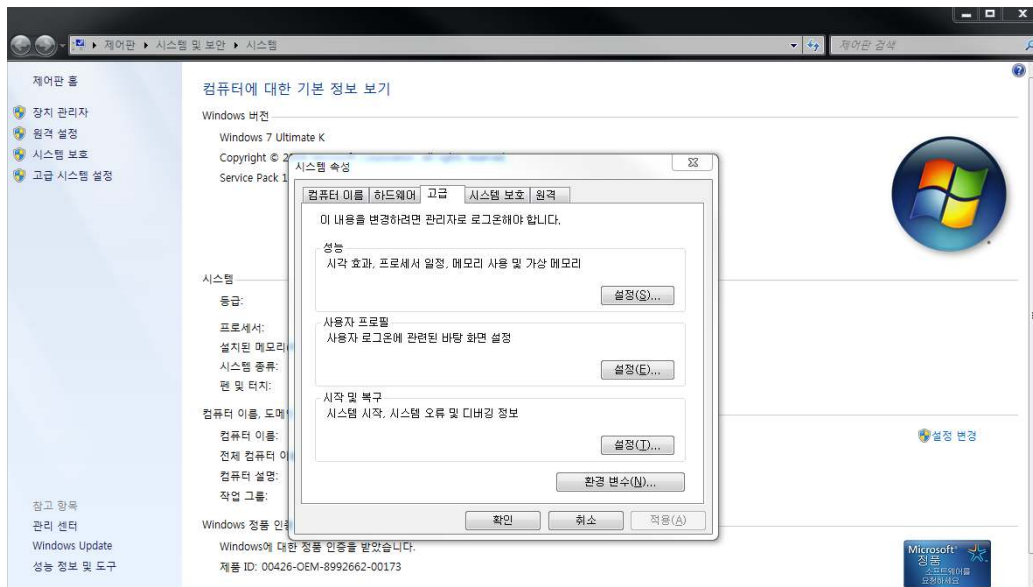
[그림 10] 개발 툴 설치

* 주의 : standatd 버전은 web 개발이 지원되지 않기 때문에 꼭 eclipse IDE for java EE Developers로 다운로드 해야함

- ③ 다운로드 파일 압축 해제

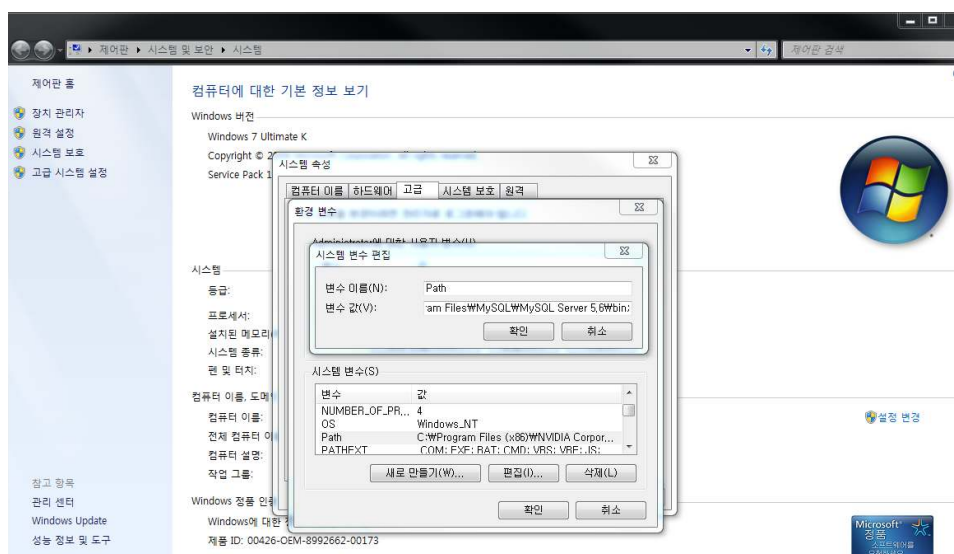
- Java 설치

- ① <http://www.cracle.com/technetwork/java/javase/downloads/index/html> 접속
- ② Java SE download
- ③ Java SE 설치
- ④ 컴퓨터 속성 - 고급 시스템 설정 - 환경변수



[그림 11] Java 설치 중 컴퓨터 속성 설정

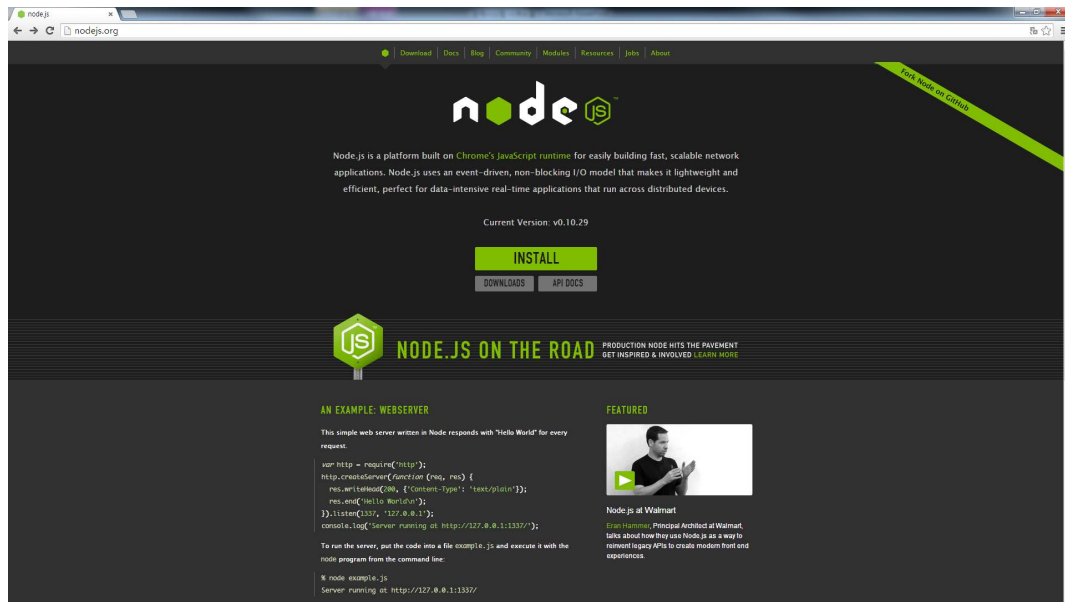
- ⑤ 컴퓨터 속성 - 고급 시스템 설정 - 환경변수
- ⑥ 시스템변수(S) -Path 편집 - 변수 값(V)에 Java 설치 경로 설정



[그림 12] Java 설치 중 변수 값에 Java 설치 경로 설정




- Node.js 설치

- ① <http://nodejs.org> 접속
- ② DOWNLOADS 클릭



[그림 13] Node.js 설치

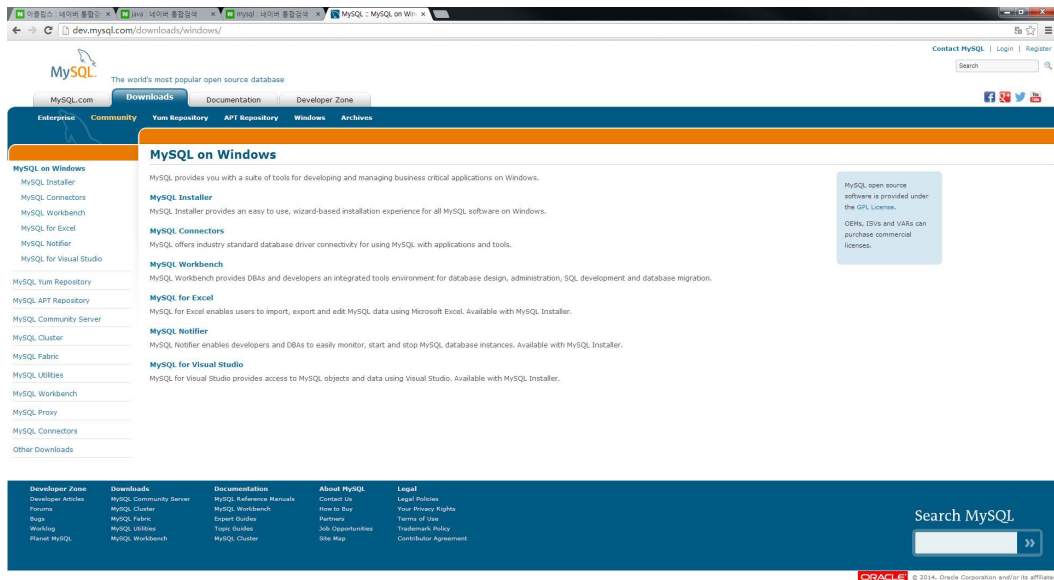
③ 자신의 OS/VERSION에 맞는 Node.js 다운로드 및 설치

<div>  Windows Installer node-v0.10.29-x86.msi </div> <div>  Macintosh Installer node-v0.10.29.pkg </div> <div>  Source Code node-v0.10.29.tar.gz </div>		
Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.exe)	32-bit	64-bit
Mac OS X Installer (.pkg)	Universal	
Mac OS X Binaries (.tar.gz)	32-bit	64-bit
Linux Binaries (.tar.gz)	32-bit	64-bit
SunOS Binaries (.tar.gz)	32-bit	64-bit
Source Code	node-v0.10.29.tar.gz	

[그림 14] 컴퓨터 속성에 맞는 Node.js 설치

- mysql 설치

- ① <http://www.mysql.com/> 접속
- ② Downloads - Windows -MySQL Installer



[그림 15] MySQL 설치

- ③ 버전에 맞는 MSI Installer 다운로드
- ④ mysql 설치
 - ID : root, PW : rkdwms23
- ⑤ java와 마찬가지로 환경변수 설정

3. 프로그램 구현

1. Agent Server Programming

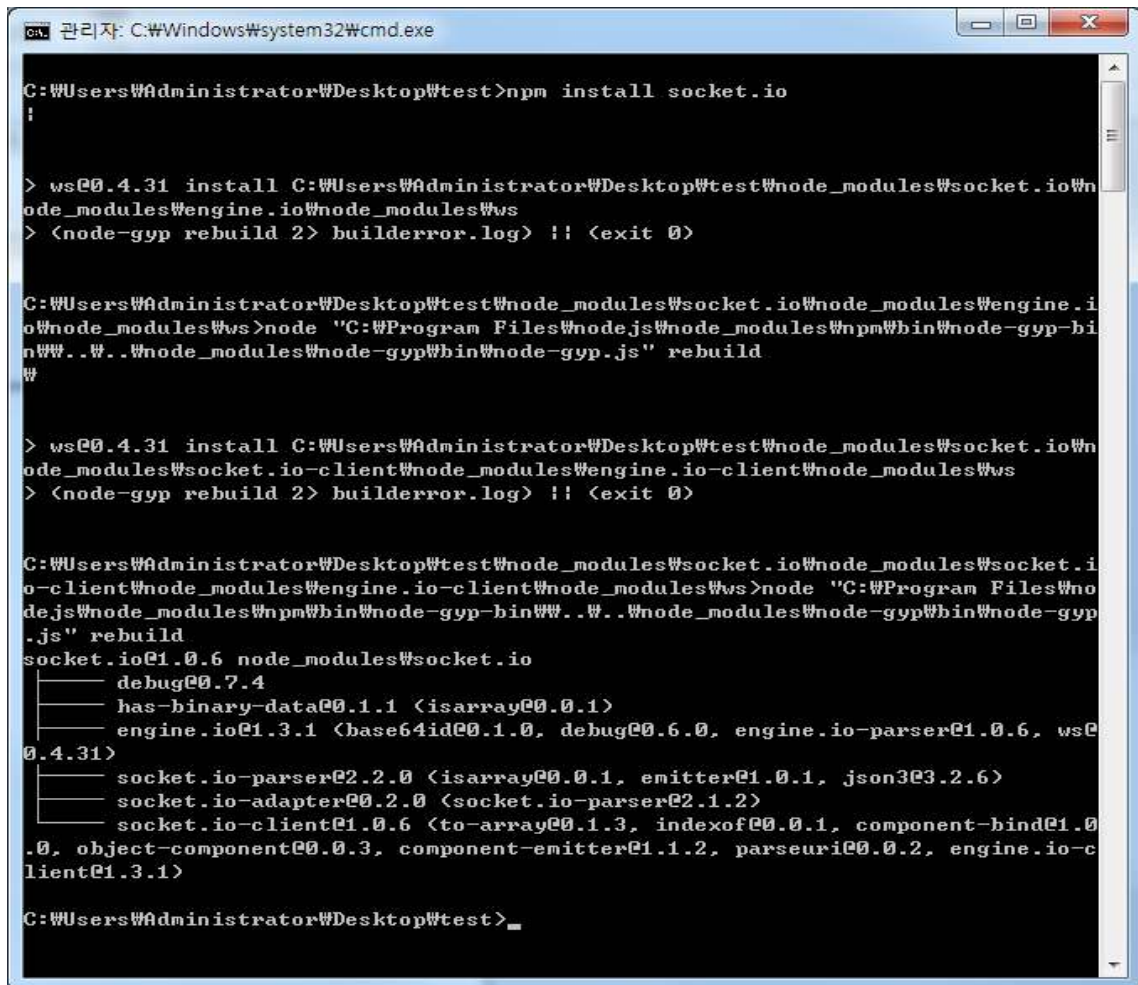
1) Agent Program 기능

- CPU 부하량 체크
- Memory 부하량 체크
- HDD 부하량 체크
- 접속자 ip && port 체크

2) Agent Program 주요 구현

- 외부 확장 Module 설치
 - * 해당 프로젝트 내에서 설치

① socket.io



```

C:\Users\Administrator\Desktop\test>npm install socket.io
!

> ws@0.4.31 install C:\Users\Administrator\Desktop\test\node_modules\socket.io\node_modules\engine.io\node_modules\ws
> (node-gyp rebuild 2) builderror.log> !! (exit 0)

C:\Users\Administrator\Desktop\test\node_modules\socket.io\node_modules\engine.io\node_modules\ws>node "C:\Program Files\nodejs\node_modules\npm\bin\node-gyp-bin\..\..\node_modules\node-gyp\bin\node-gyp.js" rebuild

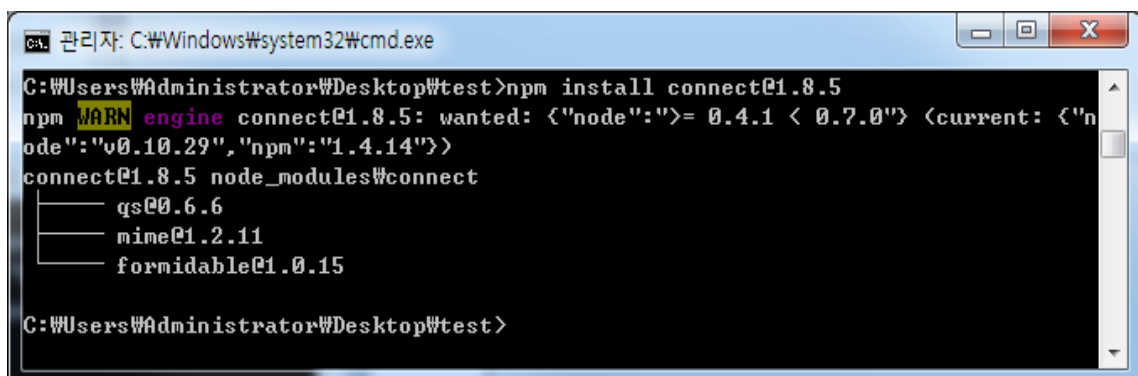
> ws@0.4.31 install C:\Users\Administrator\Desktop\test\node_modules\socket.io\node_modules\socket.io-client\node_modules\engine.io-client\node_modules\ws
> (node-gyp rebuild 2) builderror.log> !! (exit 0)

C:\Users\Administrator\Desktop\test\node_modules\socket.io\node_modules\socket.io-client\node_modules\engine.io-client\node_modules\ws>node "C:\Program Files\nodejs\node_modules\npm\bin\node-gyp-bin\..\..\node_modules\node-gyp\bin\node-gyp.js" rebuild
socket.io@1.0.6 node_modules\socket.io
├── debug@0.7.4
├── has-binary-data@0.1.1 <isArray@0.0.1>
├── engine.io@1.3.1 <base64id@0.1.0, debug@0.6.0, engine.io-parser@1.0.6, ws@0.4.31>
├── socket.io-parser@2.2.0 <isArray@0.0.1, emitter@1.0.1, json3@3.2.6>
├── socket.io-adapter@0.2.0 <socket.io-parser@2.1.2>
├── socket.io-client@1.0.6 <to-array@0.1.3, indexof@0.0.1, component-bind@1.0.0, object-component@0.0.3, component-emitter@1.1.2, parseuri@0.0.2, engine.io-client@1.3.1>
C:\Users\Administrator\Desktop\test>

```

[그림 16] 프로그램 구현-socket.io

② connect 모듈



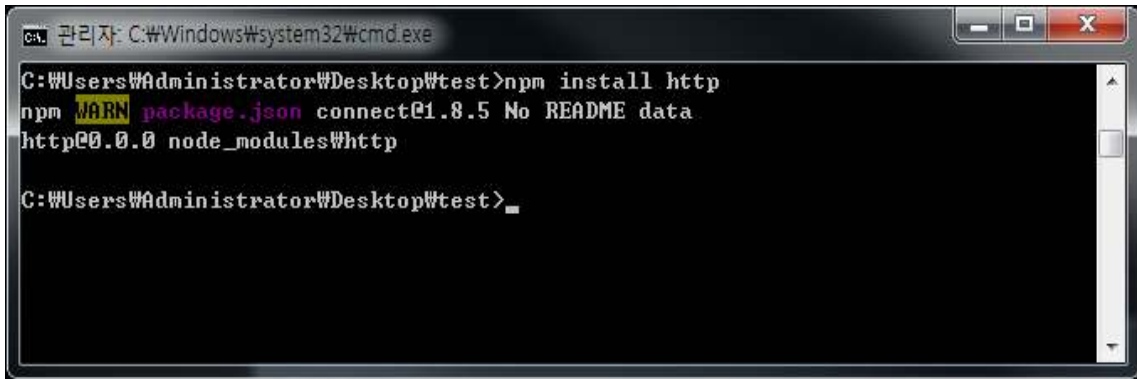
```

C:\Users\Administrator\Desktop\test>npm install connect@1.8.5
npm WARN engine connect@1.8.5: wanted: {"node": ">= 0.4.1 < 0.7.0"} <current: {"node": "v0.10.29", "npm": "1.4.14"}>
connect@1.8.5 node_modules\connect
├── qs@0.6.6
├── mime@1.2.11
├── formidable@1.0.15
C:\Users\Administrator\Desktop\test>

```

[그림 17] 프로그램 구현-connect 모듈

③ http 모듈 설치



```

C:\> 관리자: C:\Windows\system32\cmd.exe

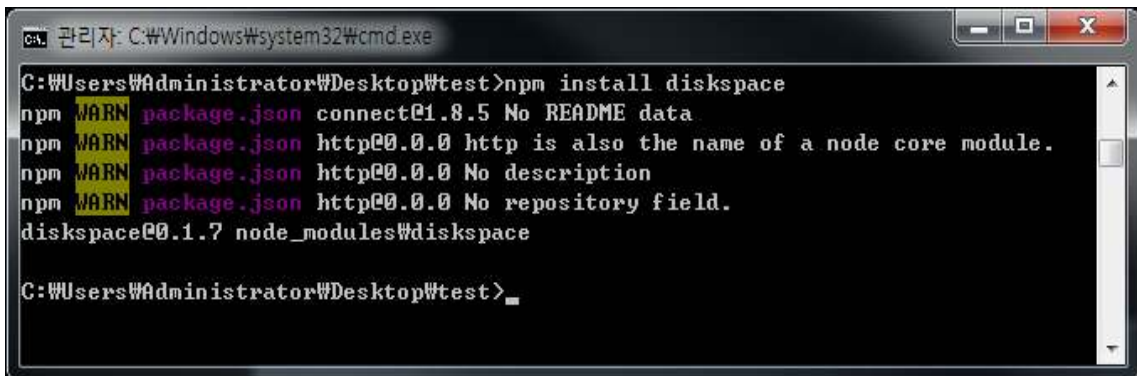
C:\Users\Administrator\Desktop>test>npm install http
npm WARN package.json connect@1.8.5 No README data
http@0.0.0 node_modules\http

C:\Users\Administrator\Desktop>test>_

```

[그림 18] 프로그램 구현-http 모듈 설치

④ diskpace 모듈 설치



```

C:\> 관리자: C:\Windows\system32\cmd.exe

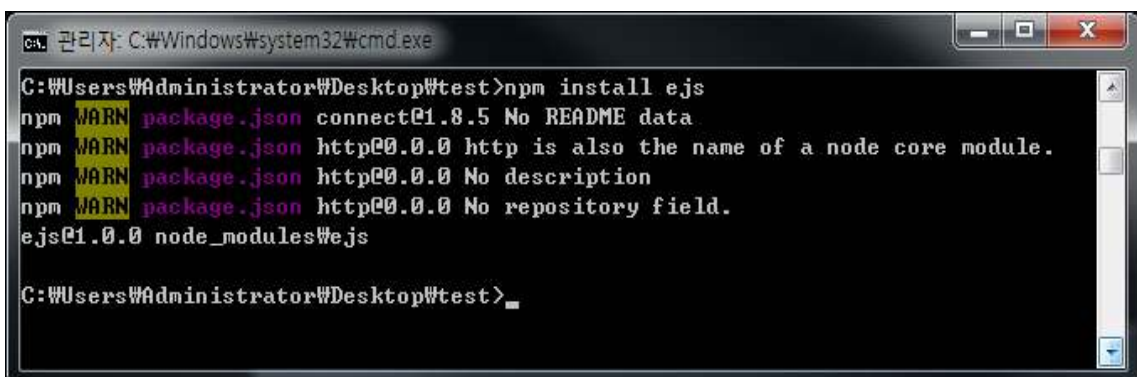
C:\Users\Administrator\Desktop>test>npm install diskpace
npm WARN package.json connect@1.8.5 No README data
npm WARN package.json http@0.0.0 http is also the name of a node core module.
npm WARN package.json http@0.0.0 No description
npm WARN package.json http@0.0.0 No repository field.
diskpace@0.1.7 node_modules\diskpace

C:\Users\Administrator\Desktop>test>_

```

[그림 19] 프로그램 구현-diskspace 모듈 설치

⑤ ejs 모듈 설치



```

C:\> 관리자: C:\Windows\system32\cmd.exe

C:\Users\Administrator\Desktop>test>npm install ejs
npm WARN package.json connect@1.8.5 No README data
npm WARN package.json http@0.0.0 http is also the name of a node core module.
npm WARN package.json http@0.0.0 No description
npm WARN package.json http@0.0.0 No repository field.
ejs@1.0.0 node_modules\ejs

C:\Users\Administrator\Desktop>test>_

```

[그림 20] 프로그램 구현-ejs 모듈 설치

⑥ mysql 모듈 설치

```

C:\Users\Administrator\Desktop\test>npm install mysql
npm WARN package.json http@0.0.0 http is also the name of a node core module.
npm WARN package.json http@0.0.0 No description
npm WARN package.json http@0.0.0 No repository field.
npm WARN package.json connect@1.8.5 No README data
mysql@2.4.1 node_modules\mysql
├── require-all@0.0.8
├── bignumber.js@1.4.0
├── readable-stream@1.1.13-1 (string_decoder@0.10.25-1, isarray@0.0.1, inherits@2.0.1, core-util-is@1.0.1)
C:\Users\Administrator\Desktop\test>

```

[그림 21] 프로그램 구현-mysql 모듈 설치

- 주요 기능 구현

① 서버 생성

```

m_agent.js
1: var http = require('http');
2: var connect = require('connect');
3:
4: var server = http.createServer(function (req, res){
5:     fs.readFile('test.html', 'UTF-8', function(err,data){
6:         res.writeHead(200, { 'Content-Type':'text/html'});
7:         res.end(data);
8:
9:     });
10: });
11: server.listen(9000);

```

- node.js에 있어 가장 중요한 부분이다. node.js도 apache와 같은 서버 역할을 하기 때문에 꼭 서버를 만들어서 열어야 서비스가 가능 하게 된다. 이번 개발 프로그램은 웹서버를 모니터링 하기 때문에 웹서버를 열어주고 서비스 할 페이지를 등록 해 준다.

포트번호를 할당하고 <http://xxx.xxx.xxx.xxx.:portnum> 으로 접속하면 서비스가 되는 것을 확인할 수 있다.

② CPU 부하량

m_agent.js

1: var os = require('os');

- 중 략 -

2: function cpuused(){

3: var cpu = os.cpus();

4: //input data of CPU in array

5: var cpu_total = 0;

6: var cpuidle = 0;

7: var cpu_idle = 0;

8: var usagecpu = 0;

9: cpumodel = cpu.model;

10:

11: for(var i = 0, len = cpu.length; i < len; i++) {

12: //console.log("CPU %s:", i);

13: var cpus = cpu[i];

14:

15: for(type in cpus.times){

16: cpu_total += cpus.times[type];

17:

18: }

19: cpuidle += cpus.times.idle;

20: cpumodel = cpus.model;

21:

22: }

23:

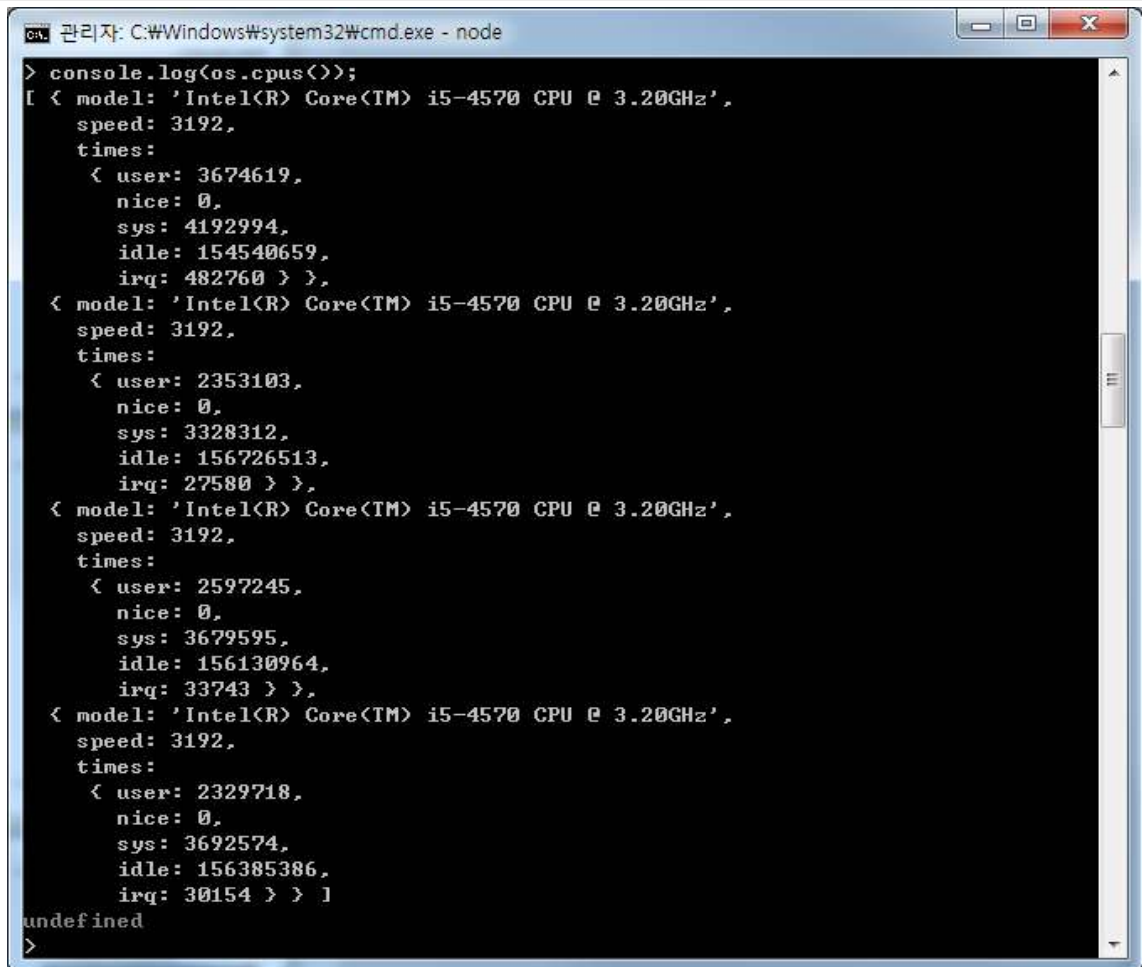
24: cpu_idle = cpuidle/len;

25: usagecpu = Math.round((cpu_idle / cpu_total) * 100);

26: return usagecpu;

27: }

-
- OS 모듈은 현재 시스템에 대한 정보를 제공한다. OS 모듈이 제공하는 함수 중 os.cpus() 함수는 cpu의 각 thread별 정보를 출력하는데 출력하는 부분은 다음과 같다.



```
> console.log(os.cpus());
[ { model: 'Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz',
  speed: 3192,
  times:
    { user: 3674619,
      nice: 0,
      sys: 4192994,
      idle: 154540659,
      irq: 482760 } },
  { model: 'Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz',
    speed: 3192,
    times:
      { user: 2353103,
        nice: 0,
        sys: 3328312,
        idle: 156726513,
        irq: 27580 } },
    { model: 'Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz',
      speed: 3192,
      times:
        { user: 2597245,
          nice: 0,
          sys: 3679595,
          idle: 156130964,
          irq: 33743 } },
      { model: 'Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz',
        speed: 3192,
        times:
          { user: 2329718,
            nice: 0,
            sys: 3692574,
            idle: 156385386,
            irq: 30154 } } ]
undefined
>
```

[그림 22] os.cpus() 출력 내용

- 각 스레드 별 사용자영역, nice영역, 시스템영역, idle영역 irq 영역을 출력한다. 부하량의 공식은 전체 스레드 (times의 합 - ilde time)/thead 개수가 되므로 위의 소스와 같이 코딩을 하여 결과 값을 출력하였다.

③ Memory 부하량

m_agent.js

```
1: var os = require('os');
```

- 중 략 -

```
2: var memory_total = os.totalmem();
```

```
3:
```

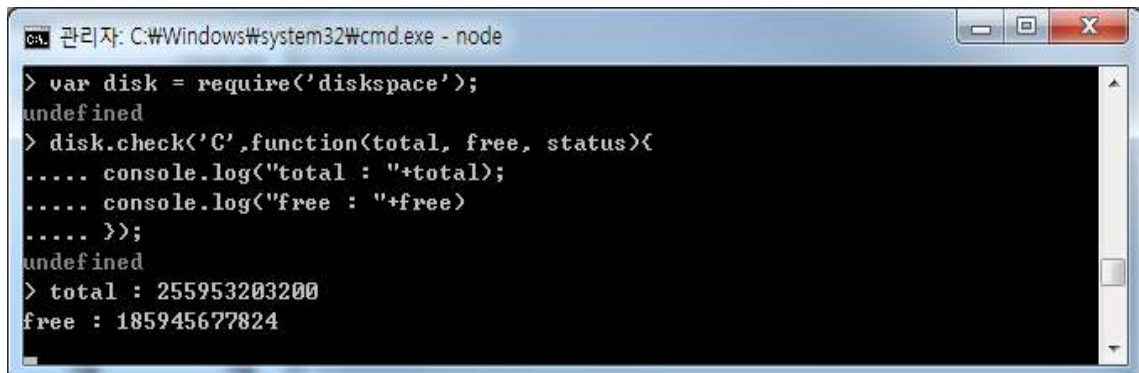
```
4: var memory_usage = Math.round(100- (os.freemem() / os.totalmem())*100);
```

- OS모듈은 os.totalmem()과 os.freemem()을 지원한다. 함수명 그대로 메모리 전체 규모와 사용되지 않는 메모리의 양을 출력한다. 메모리의 부하량을 결과값으로 얻어야 하므로 total/free * 100(free의 양)을 100에서 빼주었다.

④ HDD 부하량

```
m_agent.js
1: var os = require('os');
2: var disk = require('diskspace');
3:
4: var hdd_total=0;
5: var hdd_free=0;
6: var hdd_usage=0;
7:
8: disk.check('C', function (total, free, status){
9:
10:     hdd_total = total;
11:     hdd_usage = 100 - Math.round((free/total)*100);
12: });
```

- diskspace 모듈은 현재 내 디스크의 상태를 반환한다. 첫 번째로 지정 한 부분은 드라이브로 C드라이버를 지정하였고, 두 번째는 반환받을 함수를 지정하였다. 결과는 다음과 같다.



```
> var disk = require('diskspace');
undefined
> disk.check('C',function(total, free, status){
..... console.log("total : "+total);
..... console.log("free : "+free)
..... });
undefined
> total : 255953203200
free : 185945677824
```

[그림 23] diskspace

- 부하율 공식은 memory와 같이 코딩하였다.

⑤ mysql 연동

```
m_agent.js
1: var mysql = require('mysql');
2:
3: var db = mysql.createClient({
4:     user: 'root',
5:     password: 'rkdwns23',
6:     database: 'control'
7: });
```


- mysql 모듈은 PC 내에 생성된 DB에 연결하게 해 주는 역할을 한다.

```
변수명 = mysql.createClient({
  user: '유저명',
  password: '패스워드',
  database: 'control'
});
```

- 코딩을 해주면 DB에 접근이 가능하다. 또한 어떻게 query를 활용하느냐에 따라 동적 DB생성과 TABLE 생성이 가능하다. 본 프로그램의 query문은 전체 소스 코드페이지에서 확인하도록 하자.

⑥ Client IP 추적

m_agent.js

- 중 략 -

```
1: client_ip = req.connection.remoteAddress;
2: client_port = req.connection.remotePort;
```

- 중 략 -

⑦ 소켓 생성

m_agent.js

```
1: var socketio = require('socket.io');
```

- 중 략 -

```
2: var io = socketio.listen(server);
3: io.sockets.on('connection', function(socket){
4:   socket.on('msg', function(data){
```

- 중 략 -

```
5:   });
6: });
```

- 모니터링 서버와 데이터를 주고 받기 위해서는 socket통신부가 필요하다.

node.js의 가장 큰 강점은 socket 통신이 될 수 있는데 코드와 같이 소켓을 열어두고 통신 준비를 하면 된다.

2. Monitoring Server Programming

1) Monitoring Server Program 기능

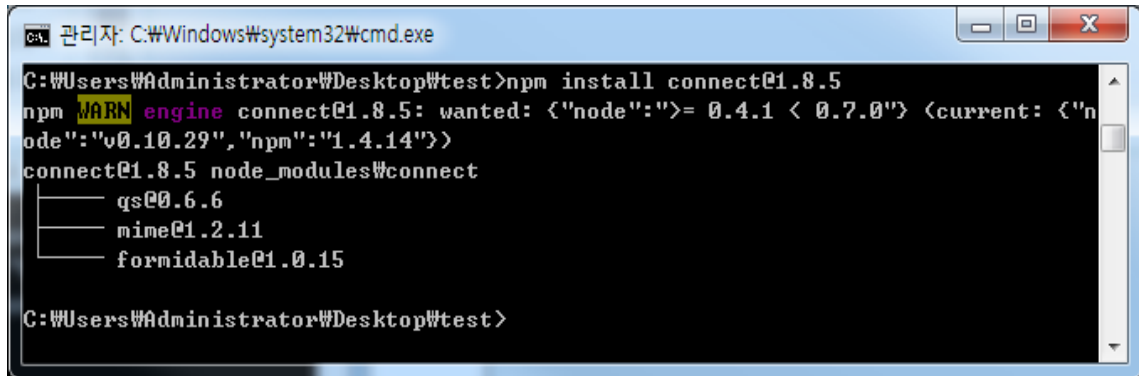
- Agent 상태 디스플레이(CPU, MEMORY, HDD, NETWORK 등)
- 에이전트 서버 로그 DB 저장
- 상태 그래프 제공

2) Monitoring Server Program 주요 구현

- 외부 확장 Module 설치

* 해당 프로젝트 내에서 설치

① connect 모듈



```

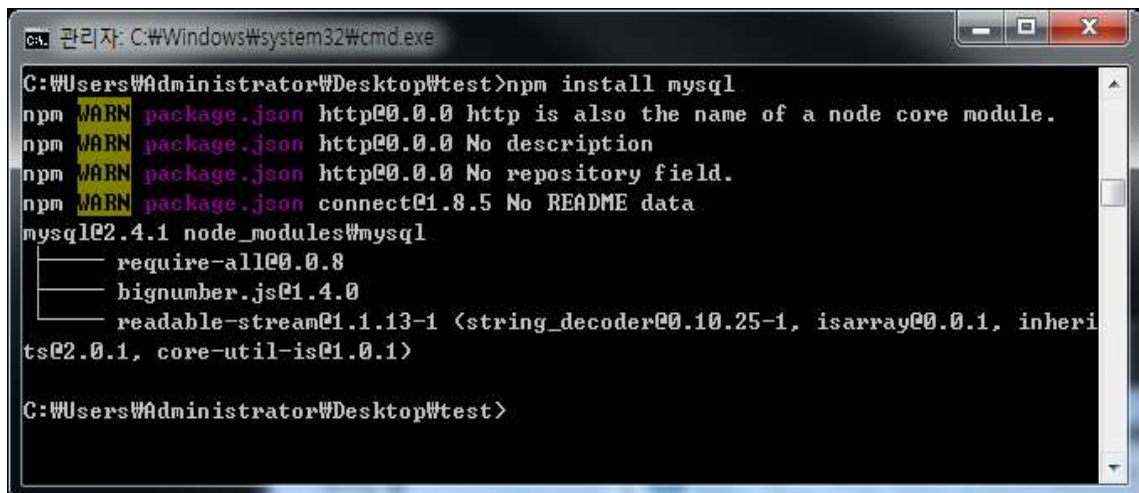
CA: 관리자: C:\Windows\system32\cmd.exe
C:\Users\Administrator\Desktop\test>npm install connect@1.8.5
npm WARN engine connect@1.8.5: wanted: {"node": ">= 0.4.1 < 0.7.0"} <current: {"n
ode": "v0.10.29", "npm": "1.4.14"}>
connect@1.8.5 node_modules\connect
├── qs@0.6.6
├── mime@1.2.11
└── formidable@1.0.15

C:\Users\Administrator\Desktop\test>

```

[그림 24] 프로그램 구현-connect 모듈

② mysql



```

CA: 관리자: C:\Windows\system32\cmd.exe
C:\Users\Administrator\Desktop\test>npm install mysql
npm WARN package.json http@0.0.0 http is also the name of a node core module.
npm WARN package.json http@0.0.0 No description
npm WARN package.json http@0.0.0 No repository field.
npm WARN package.json connect@1.8.5 No README data
mysql@2.4.1 node_modules\mysql
├── require-all@0.0.8
├── bignumber.js@1.4.0
└── readable-stream@1.1.13-1 <string_decoder@0.10.25-1, isarray@0.0.1, inheri
ts@2.0.1, core-util-is@1.0.1>

C:\Users\Administrator\Desktop\test>

```

[그림 25] 프로그램 구현-mysql

③ socket.io

```

C:\Windows\system32\cmd.exe
C:\Users\Administrator\Desktop>test>npm install socket.io
!

> ws@0.4.31 install C:\Users\Administrator\Desktop>test\node_modules\socket.io\node_modules\engine.io\node_modules\ws
> <node-gyp rebuild 2> buildererror.log> !! <exit 0>

C:\Users\Administrator\Desktop>test\node_modules\socket.io\node_modules\engine.io\node_modules\ws>node "C:\Program Files\nodejs\node_modules\npm\bin\node-gyp-bin\..\.node_modules\node-gyp\bin\node-gyp.js" rebuild

> ws@0.4.31 install C:\Users\Administrator\Desktop>test\node_modules\socket.io\node_modules\socket.io-client\node_modules\engine.io-client\node_modules\ws
> <node-gyp rebuild 2> buildererror.log> !! <exit 0>

C:\Users\Administrator\Desktop>test\node_modules\socket.io\node_modules\socket.io-client\node_modules\engine.io-client\node_modules\ws>node "C:\Program Files\nodejs\node_modules\npm\bin\node-gyp-bin\..\.node_modules\node-gyp\bin\node-gyp.js" rebuild
socket.io@1.0.6 node_modules\socket.io
├── debug@0.7.4
├── has-binary-data@0.1.1 <isArray@0.0.1>
├── engine.io@1.3.1 <base64id@0.1.0, debug@0.6.0, engine.io-parser@1.0.6, ws@0.4.31>
├── socket.io-parser@2.2.0 <isArray@0.0.1, emitter@1.0.1, json3@3.2.6>
├── socket.io-adapter@0.2.0 <socket.io-parser@2.1.2>
├── socket.io-client@1.0.6 <to-array@0.1.3, indexOf@0.0.1, component-bind@1.0.0, object-component@0.0.3, component-emitter@1.1.2, parseuri@0.0.2, engine.io-client@1.3.1>
C:\Users\Administrator\Desktop>test>_

```

[그림 26] 프로그램 구현-socket.io

④ http 모듈

```

C:\Windows\system32\cmd.exe
C:\Users\Administrator\Desktop>test>npm install http
npm WARN package.json connect@1.8.5 No README data
http@0.0.0 node_modules\http

C:\Users\Administrator\Desktop>test>_

```

[그림 27] 프로그램 구현-http 모듈

⑤ ejs

```

C:\Users\Administrator\Desktop\test>npm install ejs
npm WARN package.json connect@1.8.5 No README data
npm WARN package.json http@0.0.0 http is also the name of a node core module.
npm WARN package.json http@0.0.0 No description
npm WARN package.json http@0.0.0 No repository field.
ejs@1.0.0 node_modules\wejs

C:\Users\Administrator\Desktop\test>_

```

[그림 28] 프로그램 구현-ejs

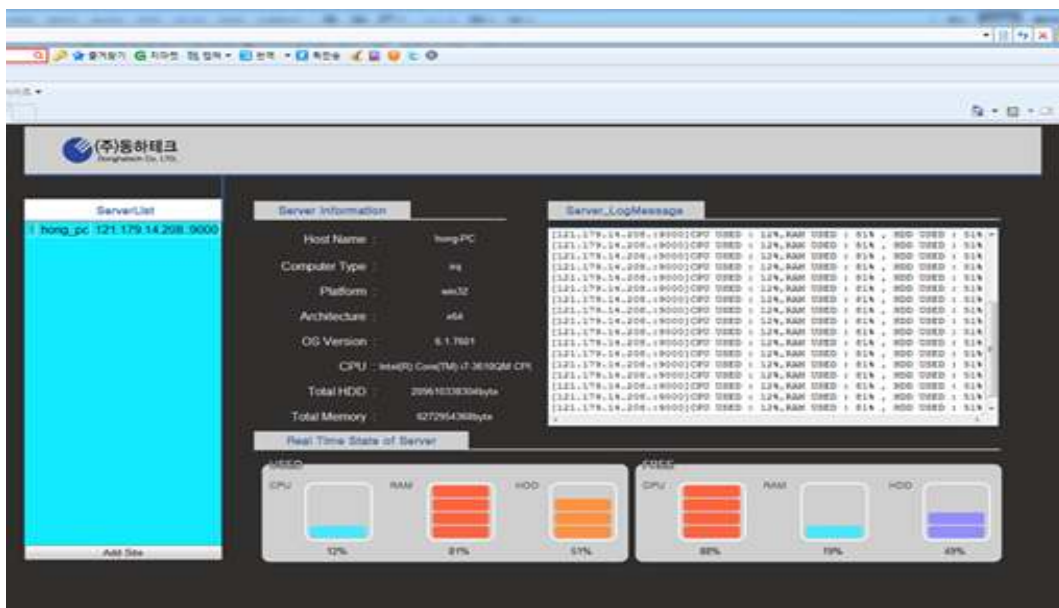
- 주요 기능 구현

- 본 프로그램은 에이전트 프로그램과 소켓통신을 통하여 모니터링 서버로 데이터를 수신한다. 사실 모니터링서버는 수신 한 데이터를 가공하여 사용자에게 디스플레이 하는 역할이다.
- 본 프로그램을 제작할 때 그래프로 가공된 데이터를 표시하고 데이터 저장의 경우 동적 테이블 생성을 통해 에이전트 등록 시 테이블이 생성되는 구조로 설계하였다.

3. 시제품 및 시스템 시연 화면

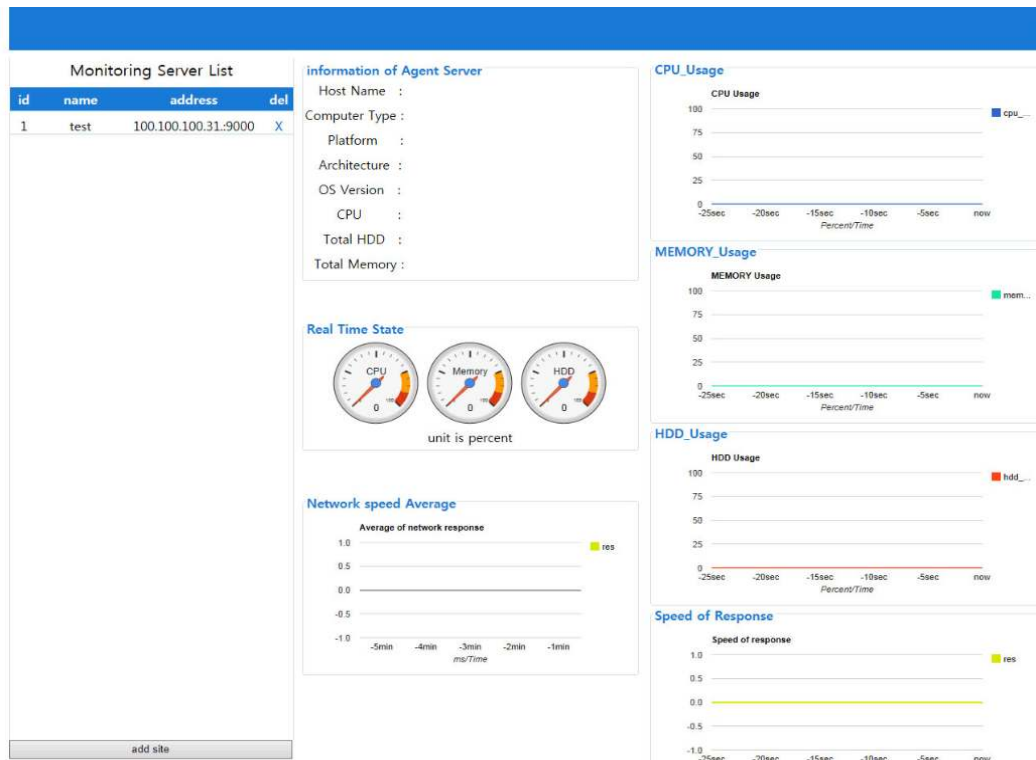
개발된 관제 시스템의 화면은 다음과 같다.

- 관제 시스템 메인 화면 (ver 1.0)



[그림 29] 관제 프로그램 v1.0

- 관제 시스템 메인 화면 (ver 2.0)



[그림 30] 관제 프로그램 v2.0

- Agent 서버 IP 추가 화면

추가 사이트 정보를 입력하십시오

Site Name :

IP Address :

Port Number :

확인 취소

[그림 31] Agent server IP 추가

- Agent 서버 방문자 DB 화면

id	ipaddr	portnum	date	time
1	100.100.100.47	51570	2014-06-18	14:10:20
2	100.100.100.33	60232	2014-06-18	14:12:20
3	100.100.100.33	60232	2014-06-18	14:12:20
4	100.100.100.35	61200	2014-06-18	14:12:23
5	100.100.100.35	61202	2014-06-18	14:12:23
6	100.100.100.47	51572	2014-06-18	14:14:06
7	100.100.100.33	60238	2014-06-18	14:14:32
8	100.100.100.33	60238	2014-06-18	14:14:32
9	100.100.100.47	40121	2014-06-18	14:15:22
10	100.100.100.47	40121	2014-06-18	14:15:22
11	100.100.100.32	49343	2014-06-18	14:18:37
12	100.100.100.32	49343	2014-06-18	14:18:38
13	100.100.100.47	51575	2014-06-18	14:19:03
14	100.100.100.56	49210	2014-06-18	14:22:12
15	100.100.100.56	49211	2014-06-18	14:22:12
16	127.0.0.1	61831	2014-06-20	19:01:54
17	127.0.0.1	61846	2014-06-20	19:04:40
18	100.100.100.33	50016	2014-06-20	19:05:34
19	127.0.0.1	62385	2014-06-20	19:49:43
20	127.0.0.1	62526	2014-06-20	19:54:55
21	127.0.0.1	62531	2014-06-20	19:56:04
22	127.0.0.1	62531	2014-06-20	19:56:52
23	100.100.100.31	62581	2014-07-20	19:57:16
24	100.100.100.35	12345	2014-07-18	14:12:20
25	127.0.0.1	56366	2014-07-21	18:39:09
26	127.0.0.1	56368	2014-07-21	18:39:18
27	127.0.0.1	56364	2014-07-21	18:39:22
28	127.0.0.1	56373	2014-07-21	18:39:27
29	127.0.0.1	56386	2014-07-21	18:40:00
30	127.0.0.1	56404	2014-07-21	18:40:33
31	127.0.0.1	56420	2014-07-21	18:42:33
32	127.0.0.1	56439	2014-07-21	18:44:23
33	127.0.0.1	56454	2014-07-21	18:44:29
34	100.100.100.31	65010	2014-07-25	17:13:55
35	100.100.100.31	65029	2014-07-25	17:14:29
36	100.100.100.31	65041	2014-07-25	17:14:42

[그림 32] Agent server 방문자 DB

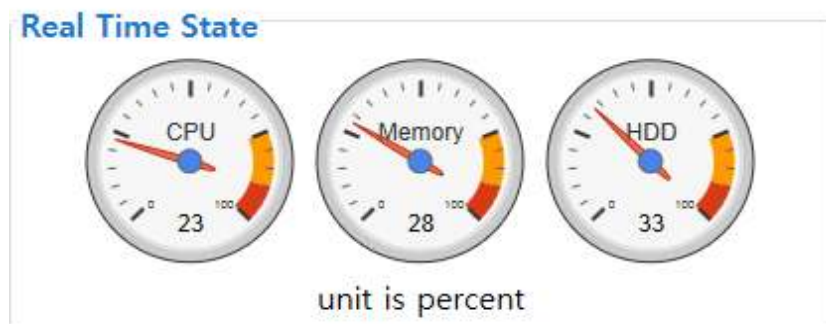
- Agent 서버 정보 표시 화면

information of Agent Server

Host Name :	MSDN-SPECIAL
Computer Type :	irq
Platform :	win32
Architecture :	x64
OS Version :	6.1.7601
CPU :	Intel(R) Core(TM) i5-4570 CPU @ 3.20GH
Total HDD :	255953203200
Total Memory :	8466157568

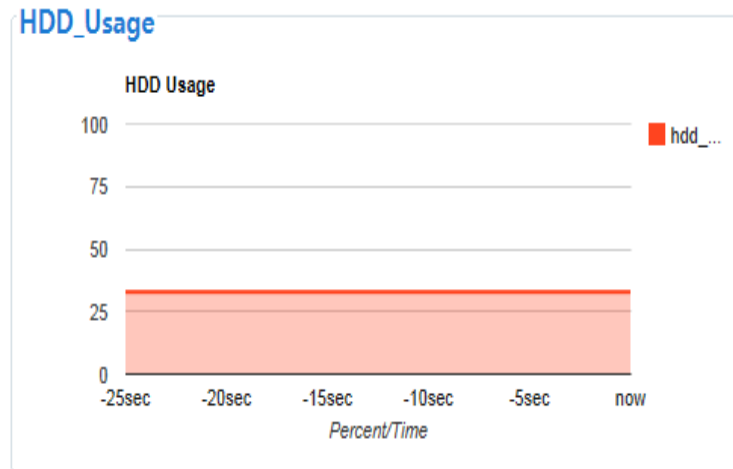
[그림 33] Agent server 정보 표시 화면

- 실시간 서버 상태 화면



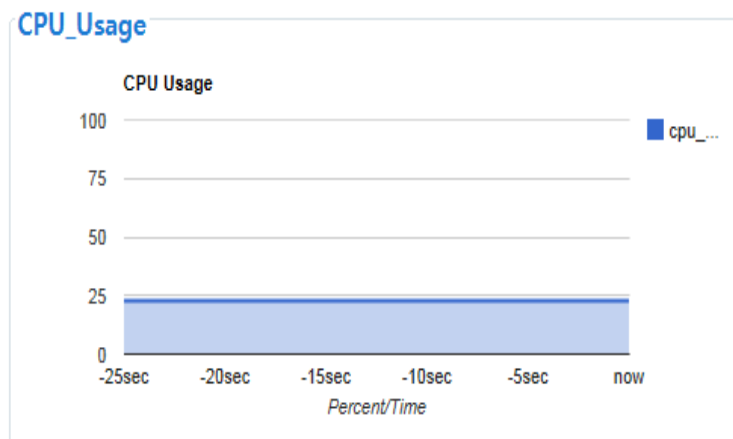
[그림 34] 실시간 서버 상태 화면

- HDD 사용률 화면



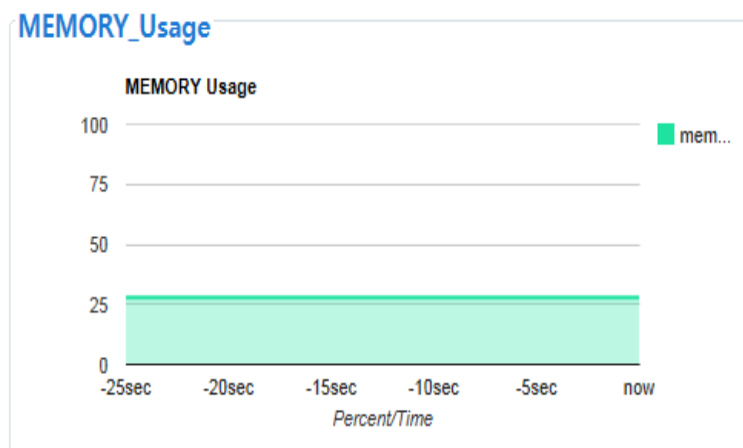
[그림 35] HDD 사용률

- CPU 사용률 화면



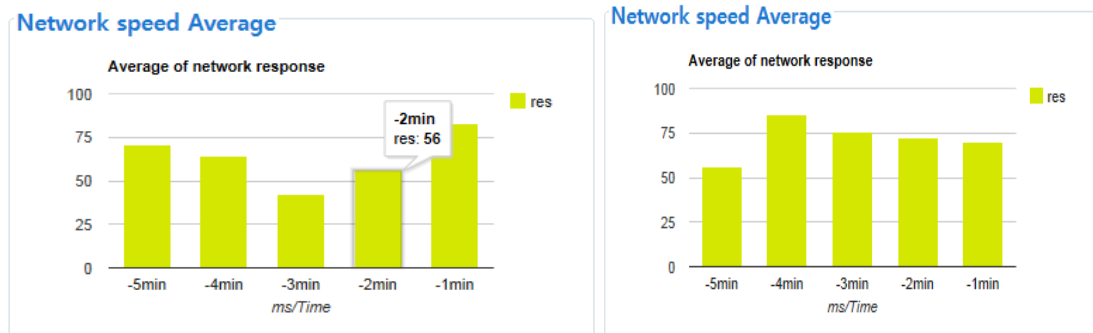
[그림 36] cpu 사용률

- Memory 사용률 화면



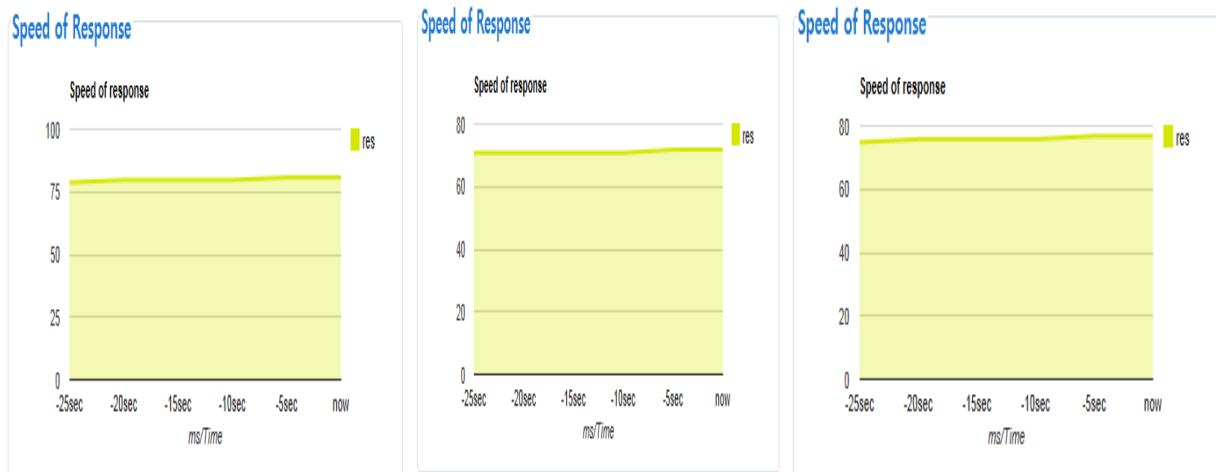
[그림 37] 메모리 사용률

- 네트워크 평균 속도 표시 화면



[그림 38] 네트워크 평균 속도

- 서버 반환률 화면



[그림 39] 서버 반환률

제2절 경제적 성과

1. 국내 주요 예상 판매처

구분	판매처	판매 단가 (천원)	예상 연간 판매량(개)	예상 판매기간(년)	예상 총판매금 (천원)	관련제품
국내	국민건강 보험공단	2,000	100	4	800,000	지능형 관제 시스템
	포스코	2,000	300	4	2,400,000	지능형 관제 시스템
	한국무역 협회	2,000	75	4	600,000	지능형 관제 시스템
	국민연금 공단	2,000	50	5	500,000	지능형 관제 시스템
	동양증권	2,000	200	3	1,200,000	지능형 관제 시스템

2. 사업화 계획

구 분		사업화 년도		
		(2014)년 (개발종료 해당년)	(2015)년 (개발종료 후 1년)	(2016)년 (개발종료 후 2년)
사업화 제품		지능형 관제 시스템	지능형 관제 시스템	지능형 통합 관제 시스템
투자계획(백만원)		100	200	400
판매 계획 (백만원)	내 수	500	800	1500
	수 출	0	200	500
	계	500	1,000	2,000
수입대체효과(백만원)		500	1,000	2,000
고용 창출(명)		2	3	5

제3절 기타 사업 성과

1. 특허출원

지능형 관제 시스템 및 방법	
출원번호통지서	페이지 1 / 3
<p>관인생략</p> <h2 style="margin: 0;">출원번호통지서</h2>	
출원일자	2013.10.07
특기사항	심사청구(유) 공개신청(무)
출원번호	10-2013-0119404 (접수번호 1-1-2013-0907388-14)
출원인명칭	우석대학교 산학협력단(2-2004-006121-0) 외 1명
발명자성명	천은홍 문병대 김종석 하태진 민병국 천상인
발명의명칭	지능형 관제시스템 및 방법
<h2 style="margin: 0;">특허청장</h2> <p style="margin: 0;"><< 안내 >></p>	
<div style="border: 1px solid black; padding: 10px;"> <p>1. 귀하의 출원은 위와 같이 정상적으로 접수되었으며, 이후의 심사 진행상황은 출원번호를 통해 확인하실 수 있습니다.</p> <p>2. 출원에 따른 수수료는 접수일로부터 다음날까지 동봉된 납입영수증에 성명, 납부자번호 등을 기재하여 가까운 우체국 또는 은행에 납부하여야 합니다. <small>* 납부자번호 : 0131(기관코드) + 접수번호</small></p> <p>3. 귀하의 주소, 연락처 등의 변경사항이 있을 경우, 즉시 [출원인코드 정보변경(경정), 정정신고서]를 제출하여야 출원 이후의 각종 통지서를 정상적으로 받을 수 있습니다. <small>* 특허로(patent.go.kr) 접속 > 민원서식다운로드 > 특허법 시행규칙 별지 제5호 서식</small></p> <p>4. 특허(실용신안등록)출원은 명세서 또는 도면의 보정이 필요한 경우, 등록결정 이전 또는 의견서 제출기간 이내에 출원서에 최초로 첨부된 명세서 또는 도면에 기재된 사항의 범위 안에서 보정할 수 있습니다.</p> <p>5. 외국으로 출원하고자 하는 경우 PCT 제도(특허·실용신안)나 마드리드 제도(상표)를 이용할 수 있습니다. 국내출원일을 외국에서 인정받고자 하는 경우에는 국내출원일로부터 일정한 기간 내에 외국에 출원하여야 우선권을 인정받을 수 있습니다. <small>* 제도 안내 : http://www.kipo.go.kr-특허마당-PCT/마드리드</small> <small>* 우선권 인정기간 : 특허·실용신안은 12개월, 상표 디자인은 6개월 이내</small> <small>* 미국특허상표청의 선출원을 기초로 우리나라에 우선권주장출원 시, 선출원이 미공개상태이면, 우선일로부터 16개월 이내에 미국특허상표청에 [전자적교환허가서(PTO/SB/39)]를 제출하거나 우리나라에 우선권 증명서류를 제출하여야 합니다.</small></p> <p>6. 본 출원사실을 외부에 표시하고자 하는 경우에는 아래와 같이 하여야 하며, 이를 위반할 경우 관련법령에 따라 처벌을 받을 수 있습니다. <small>* 특허출원 10-2010-0000000, 상표등록출원 40-2010-0000000</small></p> <p>7. 기타 심사 절차에 관한 사항은 동봉된 안내서를 참조하시기 바랍니다.</p> </div>	

2. 디자인출원

화상디자인이 표시된 컴퓨터 모니터	
출원번호통지서	페이지 1 / 3
<p>관 인 생 략</p> <h2 style="margin: 0;">출 원 번 호 통 지 서</h2>	
출 원 일 자	2014.06.26
특 기 사 항	공개신청(무) 참조번호(14022)
출 원 번 호	30-2014-0031495 (접수번호 1-1-2014-0600122-55)
출 원 인 명 칭	우석대학교 산학협력단(2-2004-006121-0) 외 1명
대 리 인 성 명	특허법인 아주양현(9-2001-100005-9)
<h2 style="margin: 0;">특 허 청 장</h2> <p style="margin: 10px 0;"><< 안내 >></p>	
<div style="border: 1px solid black; padding: 10px;"> <p>1. 귀하의 출원은 위와 같이 정상적으로 접수되었으며, 이후의 심사 진행상황은 출원번호를 통해 확인하실 수 있습니다.</p> <p>2. 출원에 따른 수수료는 접수일로부터 다음날까지 동봉된 납입영수증에 성명, 납부자번호 등을 기재하여 가까운 우체국 또는 은행에 납부하여야 합니다. <small>* 납부자번호 : 0131(기관코드) + 접수번호</small></p> <p>3. 귀하의 주소, 연락처 등의 변경사항이 있을 경우, 즉시 [출원인코드 정보변경(경정), 정정신고서]를 제출하여야 출원 이후의 각종 통지서를 정상적으로 받을 수 있습니다. <small>* 특허로(patent.go.kr) 접속 > 민원서식다운로드 > 특허법 시행규칙 별지 제5호 서식</small></p> <p>4. 특허(실용신안등록)출원은 명세서 또는 도면의 보정이 필요한 경우, 등록결정 이전 또는 의견서 제출기간 이내에 출원서에 최초로 첨부된 명세서 또는 도면에 기재된 사항의 범위 안에서 보정할 수 있습니다.</p> <p>5. 외국으로 출원하고자 하는 경우 PCT 제도(특허·실용신안)나 마드리드 제도(상표)를 이용할 수 있습니다. 국내출원일을 외국에서 인정받고자 하는 경우에는 국내출원일로부터 일정한 기간 내에 외국에 출원하여야 우선권을 인정받을 수 있습니다. <small>* 제도 안내 : http://www.kipo.go.kr-특허마당-PCT/마드리드 <small>* 우선권 인정기간 : 특허·실용신안은 12개월, 상표·디자인은 6개월 이내 <small>* 미국특허상표청의 선출원을 기초로 우리나라에 우선권주장출원 시, 선출원이 미공개상태이면, 우선일로부터 16개월 이내에 미국특허상표청에 [전자적교환허가서(PTO/SB/39)]를 제출하거나 우리나라에 우선권 증명서류를 제출하여야 합니다.</small></small></small></p> <p>6. 본 출원사실을 외부에 표시하고자 하는 경우에는 아래와 같이 하여야 하며, 이를 위반할 경우 관련법령에 따라 처벌을 받을 수 있습니다. <small>* 특허출원 10-2010-0000000, 상표등록출원 40-2010-0000000</small></p> <p>7. 기타 심사 절차에 관한 사항은 동봉된 안내서를 참조하시기 바랍니다.</p> </div>	

3. 프로그램 등록

- 노드관제 모니터링 프로그램 (ver 1.0)

4. 논문

서버 관제 시스템 과부하 방지 알고리즘 제안

2014년도 춘계학술대회

서버 관제 시스템 과부하 방지 알고리즘 제안

천은홍*, 김종석**, 이태훈***, 박혜진****, 차준섭*****
 *우석대학교 산학협력단, ***(주)솔루션뱅크, ***㈜디유, ****㈜성광,
 *****호남대학교 인터넷콘텐츠학과

e-mail : !

Algorithms for overload protection of server control system

Eun-Hong Cheon*, Jong-Seok Kim**, Tae-Hun Lee***, Hye-Jin Park****,
 Joon-Seub Cha*****

*Foundation.Of Industrial-educational cooperation, Woosuk University, Korea
 SolutionBank Co., LTD., Korea, *Dius Korea, ****Sung Gwang Korea,
 *****Dept. Of Internet Contents, Honam University, Korea

요 약

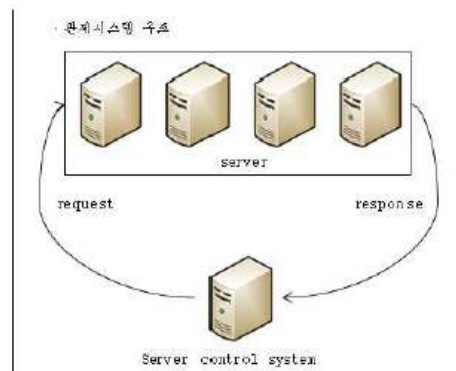
본 논문에서는 관제시스템이 관제대상 서버가 많아질 경우 서버 딜레이와 서버 다운현상을 방지하고, 일정한 속도를 유지함으로써 관제시스템은 정확한 데이터를 출력하여 기존 관제시스템의 문제점을 개선하고자 하였다. 또한, 불안정했던 관제시스템을 안정화 시키기 위해 최소한의 데이터 통신과 최적의 알고리즘을 제안함으로써 스레드로 인한 시스템 과부하를 완화하기 위한 알고리즘 구현 방법을 제안하고자 하였다.

1. 서 문

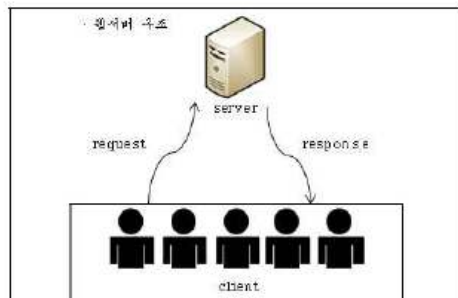
서버 관제 시스템은 서비스를 제공 하는 서버의 상태를 체크하기 위해 이용된다. 서버를 관제하는데 가장 중요한 사항은 서버의 정상운영 여부이다. 서버의 다운으로 인하여 서비스를 제공하지 못하는 경우를 감지하여 빠른 복구 작업에 임하기 위하여 서버 관제시스템을 이용한다.

그리고, 서버 다운현상을 방지하기 위함이다. 서버의 다운현상을 미연에 방지하기 위하여 CPU, MEMORY, HDD의 부하량과 인터넷 연결 여부를 체크하며, 제작된 알고리즘에 의하여 각각의 부하량 가이드라인이 정해지고 가이드라인을 넘어서게 되면 위험신호를 알리게 된다.

그럼 1과 같이 기존의 관제시스템의 개발언어와 개발방식은 웹서버 제작과 유사한 구조로 이루어져 있다.



(그림 1) 웹 서버와 관제시스템의 구동 원리 구조



그럼1과 같이 웹 서버와 관제시스템은 request와 response 구조로 이루어져 있으며 PHP, JSP, ASP와 같은 Server Language로 제작되어 다중 스레드 방식 기반 동기방식으로 제작되어 있다.

다중 스레드 방식 기반 동기방식은 하나의 Server 당 하나의 Thread가 생성되어 데이터를 수신하게 되는 구조이다. 스레드기반 동기방식은 스레드를 여러개 생성하여 일을 동시에 처리하기 때문에 굉장히 좋은 방법이며 웹서버와 관제시스템 개발에 많이 사용되는 방식이다. 하지만

제4장 결 론

제1절 기술개발결과 활용계획

1. 제품개발 추진 계획

○ 추진체계

- 관리자가 전체/개별 서버의 성능 항목(CPU/메모리/SWAP의 사용률, 부하 평균)을 모니터링 기능 및 화면구현 제공
- 서버의 운영현황 모니터링 시 장애등급별 구분기능 제공
- 그룹별 서버 상태는 장애등급이 가장 높은 서버 상태를 기준으로 모니터링
- 전체/개별 서버의 성능 모니터링 한 결과의 부하율 평균값으로 제시되도록 구축
- 서버 목록은 정상 상태, 장애 발생 및 장애 수준별로 선택되도록 구현기능 제공
- 수집 저장된 자료를 바탕으로 임의 선택한 기간에 대해 통계 처리정보 제공
- CPU, Memory에 대한 모니터링은 서버의 하드웨어적 특성과 프로세싱 방식에 따라 다르게 제공
- 저장 공간별로 구분하여 전체크기, 사용량, 가용량, 사용율 등의 구현기능 제공
- 서버의 운영체제 계정 별로 각 파일시스템에 계획 및 사용 중인 할당량(권장치 및 경고치)에 대해 조회기능, 계획된 할당량에 대해서는 권장치 및 경고치를 구분하여 구현기능 제공
- 서버 CPU 부하 및 SWAP 분석을 위한 성능수치 그래프 제공
- 서버 과부하 원인 프로세스를 식별하기 위한 프로세스 별 프로세스 명, CPU점유율, 메모리시스템 자원 점유율 정보 제공
- 성능저하의 원인이 되는 프로세스를 식별하여 프로세스 별 디스크 I/O사용 수치 조회 기능 제공
- 감시 대상으로 구분된 프로세스의 자원 점유율 통계 값 정보 제공
- 네트워크 인터페이스의 IP 및 MAC Address, 운용상태(up/down), 사용량(bps, pps)과 송수신 에러, 충돌발생량, TCP상태 값 등의 기능과 서버의 라우팅 테이블 조회 기능 제공
- 선택한 서버에 대하여 파일 배포 가능한 파일 관리 기능 제공
- 서버 파일 관리 기능의 접근 권한은 서버 접근 시에 입력한 계정 권한에 따라 제공해야 하고, 서버파일 관리 기능은 디렉토리의 생성 및 삭제, 이동 등의 기능 제공
- 관리대상 장비에 대해 일/주/월/년 통계그래프 기능 제공
- 사용자 작성 shell script 수행 결과 값에 대한 감시 기능 제공

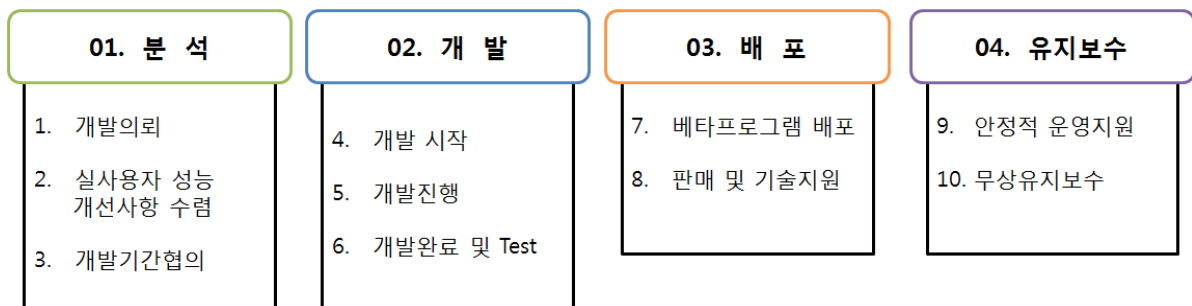
- 에이전트 별 모니터링 항목 데이터 감시주기 설정 및 On/Off 기능 제공

2. 신뢰성 (Reliability) 인증 확보 계획

- 개발 제품과 관련하여 공인된 인증기관을 통해 신뢰성 확보 예정
- 지원 운영체제의 정도
 - 지원 가능 운영체제의 종류 및 운영체제와의 충돌여부
- 관리의 용이성
 - 이용자의 하드웨어 설치가 용이한지 여부
 - 이용자의 하드웨어 관리가 쉬운지 여부
 - 다른 기종의 하드웨어와 충돌 여부

3. 제품 양산계획

- 기업 전문 엔지니어를 활용한 통합화된 양산 체제 구축
 - 분 석 : 내부 양산활동 (프로세스, 산출물 검토)
 - 개 발 : 납품 및 설치, 규격/사양 검사, 단위 및 통합시험
 - 배 포 : 사용자 및 운영자 교육, 유지보수 계획 수립
 - 유지보수 : 안정적 운영지원, 무상유지보수



(그림 7) 제품 개발 및 양산 프로세스

4. 판로 확보 및 마케팅 계획

- 판로 확보
 - 참여기관 인프라를 활용하여 전문 영업 인력을 투입 영업지원
 - 참여기업 비즈니스 파트너 활용
(PIOLINK, SUN, FUJISU, LG, IBM, EMC, FALCONSTOR, VIRBAK)

- 지능형 관제 프로그램의 가격, 제품, 커뮤니케이션 등 크게 세 가지 측면에서 각각 마케팅 톨을 도입
 - 소비자의 구매 속성 중에서 가장 대표적인 가격과 브랜드를 가지고 소비자 조사를 실시
 - 우리 브랜드를 기준으로 했을 때, 경쟁 브랜드의 가격에 대해 조사
 - 실제적인 소비자 구매 시나리오를 분석
 - 우리 제품의 가격변화에 따른 구매 의사, 경쟁 제품의 가격 변화에 따른 구매 의사등을 종합적으로 분석
- 지능형 관제 프로그램의 최적화를 위한 상품기획 모형
 - 제품의 각 기능 및 기능별 수준에 대한 선호도를 조사하여 이를 상품기획에 반영
 - 조사 결과를 차기 상품기획에 반영, 기능 추가, 불필요한 기능 삭제
 - 원가 절감, 경쟁력 제고
- 마케팅 전략
 - 전산센터 및 관련분야 실사용자 중심으로 제품 개선방향에 대한 수요조사 및 제품 Upgrade
 - 제품 관련 지식재산권 확보 및 프로그램등록, 품질인증을 통한 제품 신뢰성 확보
 - 프로그램 성능 개선을 통한 목표시장 적용 및 추가 목표시장 모색
 - 제품 브로셔 작성 및 인터넷 소셜 마케팅을 통한 제품 홍보
- 홍보 및 판매 계획
 - 2015년 개발완료와 개발제품 관련 특허 출원 및 프로그램 등록
 - 인터넷 및 전자신문, 정보통신 장비로 광고 게재 및 방송 매체 홍보
 - 제품 홍보 광고, 마케팅 시장성 분석
 - 호남 지역은 본사 직영 판매 및 공장 설립에 따른 양산 및 출시
 - 기타 지역은 수도권 소재의 네트워크 장비 개발 업체와의 제휴를 통해 영업망 및 판로 구축

제2절 기대효과

구 분	내 용
기술적 측면	<ul style="list-style-type: none"> ▪ 시설물 운영상태 실시간 감시 및 장애통보, 장애분석을 통한 재해예방 ▪ 시설 이력관리 및 부속품 적기 교체로 장비수명 연장 및 사고 피해를 미연방지 ▪ 인력중심의 관리에서 전산화에 의한 업무처리로 신속성, 정확성 증대 ▪ 중앙 집중 관리 운영으로 재난, 재해시 신속한 조치 용이 ▪ Data의 DB화 및 History 관리를 통한 사고원인 분석 용이 ▪ 관제 관리 프로그램 기술 확보 <ul style="list-style-type: none"> - 국내에 관제 프로그램이 일정한 정도 축적된 상태이나, 고립 분산적으로 존재하고 있어, 본 사업을 통해 국내 요소 기술을 결집하고 세계적인 기업과 국·내외 시장에서 경쟁 할 수 있는 기술력을 확보할 수 있음. - 향상된 관제 시스템 S/W로 상품성과 관제율을 높일 수 있고 그로인해 안정적으로 운영서버를 운영할 수 있음 ▪ 플래시를 이용한 도식화 화면 개발 <ul style="list-style-type: none"> - 프로그램 상태와 장애 발생 요소들을 도식화 하여 분석함으로써 나중에 발생하는 피해를 사전에 미리 예방할 수 있음 ▪ 운영서버와 관제서버 이원화 기술 개발 <ul style="list-style-type: none"> - 본 기술은 기존의 관제 프로그램에 관제 기능이 있는 운영서버를 더 함으로써 과부하시에 서버가 다운되는 것을 방지하여 생산성 향상 및 융통성 있고 효율적인 인력 운영이 가능해 질 것임

	<ul style="list-style-type: none"> 본 기술에 보안기술을 더하여 전산장애와 보안문제를 겪고 있는 공공기관을 비롯하여 금융권에 보급하여 보안관제 시스템으로서의 관제 시장 발전에 크게 기여 할 것으로 전망
경제 · 산업적 측면	<ul style="list-style-type: none"> 장애 발생시 자동으로 발송되는 SMS로 실시간 빠르고 정확한 대응으로 경제적 손실을 줄이고 장애 복구 시간 단축으로 시간낭비를 감소시킬 것으로 전망 관제관리프로그램을 운영서버를 운영하는 전산 운영센터 및 중소기업의 전산 운영센터에 제공함으로써 각 센터의 업무 효율성을 증대시킬 것으로 예상 이용자 안전성 및 만족도 제고, 국가재정지원 최소화 및 효율적 역할