데이터 분배 서비스를 이용한 전투 체계 관리 시스템 개발에 관한 연구

봉승용*, 최재영**, 김점수***
*숭실대학교 컴퓨터학과
**숭실대학교 컴퓨터학과
***국방과학연구소 함정전투체계개발팀
e-mail:sybong@ss.ssu.ac.kr, choi@ssu.ac.kr, chskim@add.re.kr

A Study for Combat Management System Development using Data Distribution Service

Seung-Yong Bong*, Jae-Young Choi**, Chum-Su Kim***
*Dept of Computer Science, Soongsil University
**Dept of Computer Science, Soongsil University
***Dept of Navy Research, Agency for Defense Development

요 약

과학 기술의 발전과 함께 함정에 탑재되는 무기와 각종 장비들은 컴퓨터 시스템에 의해 통제, 조작되는 형태로 변하고 있다. 다수의 장비들에 대한 모니터링 정보를 비롯한 다수의 메시지와 이를 조작하는 다양한 프로세스들을 관리해야 할 필요성이 증가하고 있다. 대규모의 실시간 메시지와 분산 환경의 감시의 일관성을 위해 해외에서는 기술표준화 작업이 진행되고 있으며, 이는 DDS(Data Distribution Service)를 기반으로 구성되고 있다. 본 논문에서는 컴퓨터 시스템에 대한 통제를 위해 OMG의 DDS(Data Distribution Service) 통신 미들웨어와 공개 프로젝트인 Fuse를 이용하여 분산 환경의 모니터링 및 파일 복제 서비스를 제공하는 전투 체계 관리 소프트웨어인 CSIMS를 설계, 구현한다.

1. 서론

과학 기술이 발전하면서 현대의 군사 장비들은 전자화, 기계화로 점차 고도화 되었고, 네트워크화 되면서 원격으로 장비를 통제하게 되었다. 이러한 발전은 복잡도가 높아진 장비들을 일관성 있게 유지하고 대량의 메시지를 손실 없이 신속하게 처리할 수 있는 컴퓨터 시스템과 통신 체계를 요구하게 되었다. 이를 위해 미 해군은 해군 함정 전투체계의 표준화 작업을 수행하였고, 그 결과 OMG의 DDS(Data Distribution Service)를 채택하여 통신 규격 및 메시지 통합을 구현하였다.

본 논문에서는 이를 배경으로 하여 국내 해군 함정 환경에 적합한 환경 감시 및 전투 체계 관리를 위한 소프트웨어 CSIMS(Combat System Integrated Management System)를 구현한다.

CSIMS는 분산 통신 미들웨어인 DDS의 통신 규격과 메시지 통합 방법을 활용한 시스템으로 분산된 노드의 모니터링을 처리하고, 부하 분산을 위한 로드밸런싱을 통해 시스템 전체의 고가용성을 유지한다. 또한 분산된 노드 데이터의 일관적인 관리를 위해 Fuse를 활용하여 모든 노드에 동일한 정보를 보유하는 어플리케이션을 설계하고 구현한다. 제 2장에서는 DDS와 Fuse를 이용한 관련연구에 대해서 기술하고 3장에서는 CSIMS의 설계 및 구현에 대해서 기술한다. 제 4장에서는 결론 및 향후 연구에 대해서 기술한다.

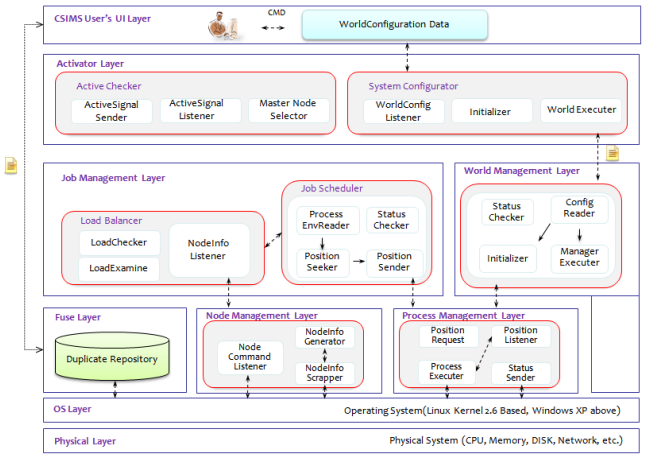
2. 관련 연구

적은 오버헤드를 갖는 발간/구독(publish/subcribe) 구조를 가진 DDS는 국내외에서 국방 사업을 중심으로 연구 프로젝트와 개발이 병행되고 있다. 국내에서는 OMG 표준 규격에 맞는 DDS의 실제 구현을 통해 JMS와의 성능 비교를 중심으로 한 연구를 진행하고 있다.[1]DDS 통신 미들웨어를 활용한 소프트웨어는 해외에서 주로 연구가 진행되고 있으며 Themis 등의 군수산업체들을 통해 활발한 연구 개발이 진행되고 있다.

Fuse는 리눅스 기반의 오픈소스 프로젝트로 사용자가 원하는 파일 시스템을 직접 구축할 수 있는 특징을 갖고 있으며 국내에서는 스트리밍 파일 서비스를 구현한 aPod[2]에서 활용된 바 있다.

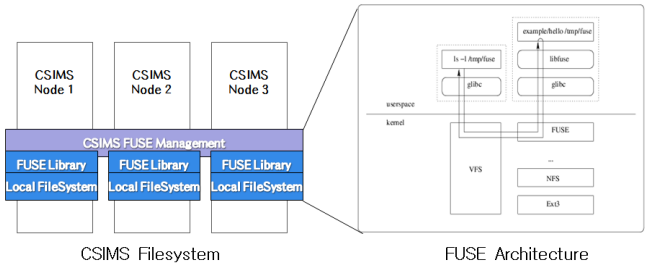
3. 설계 및 구현

CSIMS는 다양한 노드들의 감시는 물론 논리적으로 구분된 그룹들의 프로세스 통제와 데이터 복제에 대해 다루는 시스템으로 크게 사용자가 제어하는 UI Client부와 실제 각 노드별로 구동하는 CSIMS 에이전트, 그리고 최하부의 파일 시스템인 Fuse로 구성된다. CSIMS의 전체 시스템 설계는 그림 1과 같다. CSIMS는 사용자 UI를 포함한 6개의 블록으로 구성되며, 각 블록간의 내부 통신 메시지는 DDS의 Message Topic을 구성하여 전달된다. 발행자에 의해 정의되며, Topic되며,당되는 메시지를 수신하기로 정의된 구독자만이 메시지를 받 전달도록 설정되



(그림 1) CSIMS 시스템 구조

는 특징을 갖기 때문에 Topic의 정의에 따라 다양한 구성이 가능하다. 노드 지를 수관리 통제하는 NodeManager는 각 노드에 대한 통제(재시작 및 중단) 메시지를 처리하고, 쉘 스크립트를 통해 수집된 시스템 감시 정보를 발행하는 지를 수신하한다. CSIMS는 노드별 가상 논리 그룹의 생성을 통해 그룹별로 프로세스를 통제 관리를 신하한다. 각 노드는 Master와 Local로 구분되어 각 논리그룹 전체를 통제하는 역할과 자신만 수관리여 전된다. WorldManager는 사용자가 요청한 논리 그룹의 생성 및 감시 지를 수신하기며, 각 논리 그룹의 프로세스들을 개별 관리여는 ProopssManager를 구동 시킨다. 당 ProopssManager는 논리 그룹 내에 새로운 프로세스 배치시 지를 정보요청을 JobManager에 전전달하고, JobManager는 수신된 프로세스 정보를 확인하고 LoadBalancer를 통해 CPU와 메모리 사용량을 기준으로 부하가 가장 적은 최적의 노드 위치를 ProcessManager에 전달하도록 설계되었다. Activator는 CSIMS Agent의 전체를 관장하는 역할을 맡아 각 Activator 프로그램들의 heartbeat검사를 통한 오류 감지 기능을 수행할 뿐만 아니라, 클라이언트에서 발생한 사용자 명령 메시지를 수신하여 각 하위 블록 Manegement 프로세스에게 전달하는 역할을 수행한다.



(그림 2) FUSE 메커니즘

그림 2는 fuse의 메커니즘을 보여주는 그림으로 파일 입출력 관련 처리를 커널 레벨이 아닌 유저 레벨에서 사용 가능하다는 점과 특정 마운트 지점을 갖기 때문에 가상적으로 마운트 된 별도의 디스크처럼 보일 수 있다는 장점을 갖고 있다. 마운트 된 지점에서 파일 제어 관련 명령이 수행되면 Fuse API[3]는 기존의 파일 처리 기능을 가로채 사용자가 원하는 동작들을 유저레벨에서 구현 할 수 있도록 설계되었다. 표 1은 fuse API를 통해 파일 생성 작업을 수행하는 코드를 보여주고 있다.

<표 1> fuse의 API 가로채기 구현

```
static int xmp_release(const char *path, struct fuse_file_info *fi)
{
    struct stat buf;
    char absPath[1024];
    stat(getAbsPath(path, absPath),&buf);
    /* 해당 path에 create_file 작업을 수행한다 */
    if(S_ISREG(buf.st_mode)){
                create_file(path);
     }
    close(fi->fh);
            return 0;
}
```

현재 CSIMS에서는 파일의 생성 시, 각 노드별로 마운트된 Repository에 동일한 파일 데이터를 전송하여 생성하도록 하는 기능을 수행하고 있으며, 이는 TCP/IP소켓을 이용한 서버-클라이언트의 통신과 쓰레드를 이용한 브로드캐스트를 통해 이뤄진다. 표 2는 브로드캐스팅 구현을 보여 준다.

<표 2> fuse의 broadcast 구현

```
void broadcast(int clntSock, void *message, int size)
{
        int index;
        for(index = 0; index < execuSockCnt ; index++){
                if(arrExeSock[index] != clntSock ){
                        write( arrExeSock[index], message, size);
                                fflush(stdout);
                }
        }
}
```

또한 상위 서비스 블록과 관계없이 독립적으로 운영, 동작하며 CSIMS운용에 필수적인 정보들을 저장하는 복제 저장소를 구축하기 위해 개발되었다.

4. 결론 및 향후 연구

본 논문에서 제안하는 CSIMS는 통신 미들웨어인 DDS를 통해 범용성을 갖는 체계관리 소프트웨어를 구성하였다. DDS의 일관된 메시지포맷은 일반 TCP/IP 통신에서 겪는 단점들을 극복 할 수 있다. 또한 오픈소스 fuse 파일 시스템을 구축하여 분산 노드상의 파일 일관성 유지를 꾀하였다. 향후 연구로는 DDS의 QoS를 활용한 메시지 제어 연구를 통한 기능 분화 작업이 연구되어야 할 것이다.

참고문헌

[1] 박충범, 권기정, 차다함, 최 훈, 김점수, "데이터 분배 서비스 시스템 설계 및 분석", 한국정보과학회, 정보과학회 논문지: 컴퓨팅의 실제 및 레터 제14권 제2호, pp 211-215, 2008. 04.
[2] 김형석, 이진복, 최재영, "그리드 환경에서 가상파일시스템을 이용한 응용프로그램 스트리밍 시스템, 한국정보과학회, 2008종합학술대회 논문집, 제35권 제1호(A) pp 317-318, 2008.06
[3] FuseAPI[Online], http://fuse.sourceforge.net/doxygen/index.html