

SEED를 이용한 Web 기반 클러스터시스템 어카운팅 설계

The Design for the Web Based Cluster System Accounting applying SEED

오 충 식

한국과학기술정보연구원

Oh Chung-Shick

Korea Institute of Science and Technology Information

요약

컴퓨팅 환경의 고도화와 인터넷 사용자의 급증으로 인해 현재의 Web기반 클러스터 시스템 어카운팅 서비스는 과거에 비해 다수의 사용자로 하여금 보다 많은 양의 데이터를 고속으로 접근할 수 있게 하고 있다. 하지만 시스템 환경의 편리함 만큼이나 사용자 정보와 데이터의 보안 역시 중요한 요소 중 하나이며 특히 데이터 유출, 해킹 그리고 악의적인 코드 등으로 인한 개인과 기관의 피해는 인터넷 서비스 사업 분야에 있어 매우 중요한 문제로 대두되고 있다. 본 연구에서는 국내 TTA(한국정보통신기술협회) 표준 암호 알고리즘인 SEED를 이용한 보다 안정적인 웹기반 클러스터시스템 어카운팅 서비스 방안을 제안한다.

Abstract

Both the highly developed computing environment and the rapid increase of the internet users enable the present web based cluster system accounting service to help many users access to numerous data at high speed. However, the information security of users and data is also as important as the convenience of the systematic environment. Especially, the significance of damage to the individuals and organizations resulted from the data outflow, hacking and malicious coding has risen up to one of the most essential problems in the internet service business. In this study, I suggest a more safe web based cluster system accounting service solution applying SEED, the Korean Telecommunications Technology Association (TTA) standard encryption algorithm.

1. 서론

최근 들어, 컴퓨터 자원들의 초고속화, 대용량화 추세로 인하여 클러스터 시스템 기반의 분산 컴퓨팅 체제가 구축되어지고 있으며, 생명공학, 유체역학, 기상기후 예측 등의 고부가 첨단정보의 계산 및 저장자원을 필요로 하게 되었다. 그리고 정보 보호의 필요성이 높아지면서, 암호화 및 복호화에 대한 관심이 커지고 있다. 특히, 대용량 정보 처리를 위해 필요한 시스템 자원의 효율을 높이고 형평성 있는 서비스 제공을 위하여 클러스터 어카운팅 시스템 개발이 필수적이다[5]. 그러나 기존에 구축된 텍스트 기반의 어카운트 시스템은 텍스트 기반으로 관리하기 때문에 많은 보안적인 문제점을 야기하고 있었다. 따라서 본 논문에서는 사용자접근의 편의성을

고려하며, 데이터 유출로 인한 개인과 기관의 피해를 최소화하기 위해 1999년 한국정보통신기술협회(TTA)에서 제정한 국내 표준 암호 알고리즘인 SEED를 적용한 웹 기반 클러스터 어카운팅 시스템을 설계하였다.

2. 관련 연구

2.1 기존 시스템과 비교

기존에 개발되고 있는 어카운팅 시스템과 본 논문에서 제안하는 시스템을 비교하면 아래와 같으며, 본 논문에서 제안하고자 하는 시스템은 사용자 접근성, 데이터 보안성, 사용의 편리성, 시스템 안정성 등을 모두 보장할 수 있다는 특징을 가진다. 그림 1은 기존 시스템과

클러스터 어카운팅 시스템에 대하여 비교하였다.

구분	기존 시스템	사용자 계정 관리 시스템
데이터 형태	Plain Text	Cipher Text
접근 방식	Telnet, FTP	Web 기반 HTTP
명령 처리 형태	개별적인 Command Line 방식	통합 및 자동 처리
로그 관리	서버별로 기록	각각의 Node 및 메인 클러스터에 기록하며 사용자 계정관리시스템에서 통합함
일회화	없음	SEED 128Bit 블록 암호화 기법 적용
통제 수단	시스템 관리자의 수작업	자동 로그 분석 및 통계 리포트 기능
확장성	없음(독립된 서버 단위)	Web기반 통합 계정관리 시스템을 통해 또 다른 클러스터의 추가, 변경, 공동사용 등의 관리성을 제공 가능

▶ 그림 1. 기존시스템과 클러스터 어카운팅 시스템 비교

2.1 관련연구

2.1.1 클러스터 시스템

최근 인터넷의 급속한 보급으로 인터넷을 활용한 사회 활동이 급증하면서 웹서버, 전자상거래서버, VOD서버, 게임서버 등을 포함한 여러 분야에서 고성능 서버에 대한 수요가 증가하고 있다. 그러나, 고성능 서버는 가격이 수억에서 수십억 원에 이르고 있어 구입 및 활용에 상당한 어려움을 가지고 있다. 이런 문제를 해결하기 위한 방안으로서 클러스터 시스템이 등장하였다. 클러스터 시스템은 저가의 고성능 PC 및 workstation을 고속 네트워크를 사용하여 연결함으로써 고성능 서버의 성능을 발휘하는 시스템을 제작하는 것으로 NOW(Network Of Workstations) 혹은 COW(Cluster Of Workstations)로도 불린다.[5] 클러스터 시스템의 장점은 저가의 상용제품을 사용함으로써 기존 고성능 서버보다 수배에서 수십배 작은 비용으로 동일한 성능의 시스템 구현이 가능하여 가격 대 성능비가 높다는 것이다. 그리고 사용자가 직접 상용부품을 사용하여 즉각적인 업그레이드가 가능함으로 시스템 유지비용이 감소하고, 사용이 편리한 PC 및 Workstation의 개발 환경을 그대로 사용할 수 있기 때문에 프로그램 개발이 용이하다. 클러스터 시스템이 고성능 서버에 필적하는 단일 컴퓨팅 시스템으로서의 성공 여부는 시스템 노드를 연결하는 네트워크의 성능과 지원 소프트웨어에 달려있다.

클러스터 시스템의 네트워크는 고성능 서버의 시스템 버스 역할을 담당하지만, 현재 클러스터 시스템에 사용중인 네트워크는 고성능 서버의 시스템 버스 성능과는 많은 차이가 있다. 이러한 문제를 해결하기 위해서 gigabit급의 네트워크들이 개발되고 있으며 통신에 필요한 소프트웨어 부하(overhead)를 최소화시킨 통신 라이브러리들이 개발되고 있다. 그리고 클러스터 시스템의 성능향상을 위해서는 각 노드에 분산된 자원을 효율적으로 관리하는 소프트웨어가 필요하다.

2.1.2 어카운팅 시스템

어카운팅 시스템은 네트워크 서비스 시스템 자원에 대한 상태 및 사용여부, 사용자에 대한 인증 및 접근 통제 그리고 서비스 자원에 대한 비용산출을 계산하는 시스템이다.

인터넷 어카운팅 시스템을 이해하는데 있어서 가장 먼저 이해해야 할 점은 "얼마나 지불되어야 하는가"(what is being paid for)와 "누구에게 지불되어야 하는가"(who is being paid)이다.[6]

"얼마나 지불되어야 하는가"(what is being paid for)에 있어서 문제는 transport accounting과 content accounting으로 구분되어진다. transport accounting의 목적은 인터넷을 통한 데이터를 전송하는 행위에 대해 요금을 책정하는 것이다. content accounting은 인터넷을 통해 전송하는 콘텐츠나 서비스에 대해 요금을 책정하는 것이다.

"누구에게 지불되어야 하는가"(who is being paid)에 대한 대답은 두 가지 답이 있을 수 있다. 바로 네트워크 제공자이거나 콘텐츠 소유자이다. 네트워크 제공자가 계산서를 발행할 경우에는 provider-based accounting이라고 불린다.[6]

본 연구에서 제안하는 시스템은 데이터 서비스를 제공하는 content accounting에 기반한 시스템으로서 사용자 이용정보를 상위관리 모듈로 적재하여 이용정보 통계를 내어 관리자가 사용자 관리 및 모니터링 할 수 있을 뿐만 아니라 사용자 자신의 이용정보를 확인할 수 있는 구조를 갖는다.

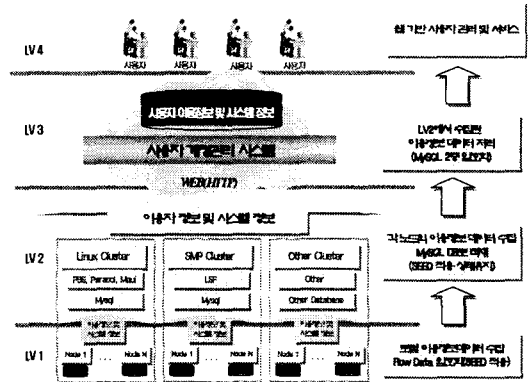
2.3 국내 표준 SEED 암호화 알고리즘

암호 시스템은 암호화 키와 복호화 키가 서로 같은가 다른가에 따라 비밀키 시스템과 공개키 시스템으로 나눌 수 있다. 송수신자간에 암호화 복호와 키가 서로 동일하면 이것을 비밀키 암호 시스템 또는 대칭키 암호 시스템이라고 하고 암호복호화 키가 서로 다르면 공개키 암호 시스템 또는 비대칭키 암호 시스템이라고 한다.

비밀키 암호 시스템[2]은 공개키 암호시스템에 비하여 그 키가 상대적으로 작아서 효율성이 좋으며 공개키 암호 시스템[3]에 비해 암호복호화 속도가 상당히 빠르므로 대량 정보를 암호화 하는 데에도 적합하다. 그리고 알고리즘 구조가 치환과 순열의 조합으로 되어 있어서 시스템 환경에 맞는 적절한 암호 알고리즘 개발을 쉽게 구현할 수 있다는 장점이 있다. 하지만 키관리의 어려움 때문에 상호 지리적으로 멀리 떨어진 통신 당사자간의 안전한 키 교환이 상당히 어려우며 통신당사자가 달라질 때마다 계속해서 다른 키를 요구하므로 불특정 다수를 대상으로 하기에는 부적당하다는 단점도 있다.[4] 본 연구에서는 비밀키 암호 시스템 중에서 국내표준인 SEED를 적용하여 시스템을 설계 하였다.

3. 클러스터 어카운팅 시스템 설계

본 논문에서는 크게 다음과 같은 것들을 고려하였다. 첫째, 타인의 시스템 자원을 임의로 사용 방지를 통해 형평성 있는 요금부과와 실시간 시스템 자원 사용 현황 파악 및 통계 자료 작성을 위해 클러스터 정보를 수집, 둘째, 시스템에 대한 신뢰성 저하로 인한 사용자 감소 및 정보 유출의 우려를 최소화하기 위해 국내 표준 암호화 알고리즘 SEED를 적용, 셋째, 사용자 접근의 편의성과 자원 관리의 편의성을 고려한 Web 기반 계정관리 시스템으로 크게 3부분으로 나누어서 설계하였다. 그림 2는 전체 시스템 구성도 이다.



▶▶ 그림 2 전체 시스템 구성도

3.1 클러스터 정보 수집

3.1.1 로그 정보 스키마

각각의 클러스터에 분산되어있는 단위 노드들에 대한 이용 정보 로그 파일을 대상으로 필요한 정보를 추출하여 통계에 활용하게 되는데 이때, 생성되는 노드들의 모든 로그정보들의 효율적인 관리를 위하여 각 단위작업마다 표준 스키마를 기반으로 로그파일을 생성하게 된다.

3.1.1.1 단위노드

클러스터에서 사용자 업무를 각각의 노드에서 나누어서 병렬로 수행하게 될 때, 어카운팅 시스템은 단위 노드의 이용에 관한 다음과 같은 정보를 기록하게 된다. 참고로 Interactive 작업 시 로그정보는 /var/account/pacct 기록되며, Batch 작업 시에는 /usr/spool/PBS/server_priv/accounting에 생성되도록 지정하였으며, 52개의 노드로 구성된 인텔기반 리눅스 클러스터 시스템을 이용했다 .

[표 1] CpuUsage.log(CPU 사용 정보)

구분	타입	설명
PNode_Name	varchar(32)	노드명
PNode_Username	varchar(32)	사용자명
PNode_Cmd	int(15)	수행시킨 명령어수
PNode_Re	float(18)	실사용 cpu time
PNode_Cpu	float(18)	총사용 cpu time
PNode_Aio	float(18)	평균 I/O 연산
PNode_Mem	varchar(10)	평균 메모리 사용량 (core-KB/sec)
PNode_Date	varchar(10)	날자(2003/10/01)
PNode_Time	varchar(11)	시간(10:11:12)

[표 2] ConnectUsage.logConnection사용정보)

구분	타입	설명
CNode_Name	varchar(32)	노드명
CNode_Username	varchar(32)	사용자명
CNode_Conn	float(16)	연결 시간
CNode_Date	varchar(10)	날짜(2003/10/01)
CNode_Time	varchar(11)	시간(10:11:12)

[표 3] DiskUsage.log(DISK 사용 정보)

구분	타입	설명
DNode_Name	varchar(32)	노드명
DNode_Username	varchar(32)	사용자명
DNode_Files	int(15)	파일수
DNode_Size	int(15)	파일크기
DNode_Aves	float(16)	파일평균크기
DNode_Date	varchar(10)	날짜(2003/10/01)
DNode_Time	varchar(11)	시간(10:11:12)

[표 4] BatchUsage.log(Batch 사용 정보)

구분	타입	설명
BNode_Name	varchar(32)	노드명
BNode_Username	varchar(32)	사용자명
BNode_Job	varchar(32)	작업명
BNode_Stat	varchar(2)	상태
BNode_Queue	varchar(32)	큐이름
BNode_Ctime	varchar(22)	제출시간
BNode_Qtime	varchar(22)	큐 제출시간
BNode_Etime	varchar(22)	큐 완료시간
BNode_Start	varchar(22)	실행시작시간
BNode_End	varchar(22)	실행완료시간
BNode_Hosts	int(3)	실행 호스트 수
BNode_Cpu	varchar(10)	CPU 사용시간
BNode_Mem	varchar(10)	메모리 사용량
BNode_Vmem	varchar(10)	가상 메모리 사용량
BNode_Wall	varchar(10)	실행시간
BNode_Exitst	int(5)	상태코드
DNode_Date	varchar(10)	날짜(2003/10/01)
DNode_Time	varchar(11)	시간(10:11:12)

3.1.1.2 클러스터 통합 스키마

각각의 노드에 쌓이는 정보들은 클러스터 내의 MySQL 데이터베이스에 저장된다. 이는 데이터베이스가 가지는 편의성을 활용하기 위함이며, 또한, MySQL에서 제공하는 필드 암호화를 적용하여 보다 정보보호 정책을 확실히 하고자 함이다.

[표 5] Cluster.tbl(클러스터 정보 : 일별 레코드)

구분	타입	설명
P_Key	int(12)	주 키
CLServer_Type	varchar(32)	클러스터 명
CLNode_Name	varchar(32)	노드명
CL_Username	varchar(32)	사용자명
CL_Cmd_Sum	int(15)	수행시킨 명령어수 합
CL_Re_Sum	float(18)	실사용 cpu time 합
CL_Cpu_Sum	float(18)	총사용 cpu time 합
CL_Aio_Sum	float(18)	평균 I/O 연산 합
CL_Mem_Sum	varchar(10)	평균 메모리 사용량 합 (core-KB/sec)
CL_Conn_Sum	float(16)	연결 시간 합
CL_Files_Sum	int(15)	파일수 합
CL_Size_Sum	int(15)	파일크기 합(디스크사용량)
CL_Aves_Sum	float(16)	파일평균크기 합
CL_Date	varchar(10)	날짜(2003/10/01)
CL_Time	varchar(11)	시간(10:11:12)
CL_Salt1	varchar(2)	전체레코드 암호화 키
CL_Salt2	varchar(2)	salt1 암호화 키



- ^ : SEED를 이용해서 암호화 한 후 MySQL로 통합하여 저장시 MySQL 암호화 함수 이용
 v : MySQL 암호화 함수만 이용
 † : SEED로 암호화된데이터

▶▶ 그림 3. MySQL 암호화 형식

3.1.1.3 사용자 관리 스키마

사용자마다 클러스터 시스템의 사용에 대한 현황이나, 사용자 요구로 인한 클러스터의 트래픽 관리를 위해

필요한 스키마는 다음과 같다.

[표 6] User.tbl(사용자 정보데이터블)

구분	타입	설명
P_Key	int(12)	주 키
MServer_Type	varchar(32)	클러스터 명
MNode_Name	varchar(32)	노드명
M_Username	varchar(32)	사용자명
M_Uid	int(10)	사용자 ID
M_Gname	varchar(32)	그룹명
M_Gid	int(10)	그룹 ID
M_Name	varchar(32)	사용자성명
M_Jumin	varchar(14)	주민등록번호
M_Rep	varchar(20)	직위
M_Email	varchar(40)	전자메일
M_Addr	varchar(52)	주소
M_Mobile	varchar(20)	핸드폰
M_Tel	varchar(20)	전화
M_Fax	varchar(20)	팩스
M_Company	varchar(20)	회사명
M_Div	varchar(20)	소속
M_Grade	varchar(4)	구분
M_Regdate	varchar(20)	시스템사용시작일
M_Expdate	varchar(20)	시스템사용종료일
M_Stopdate	varchar(21)	정지일시
M_Field	varchar(5)	분야
M_Group	varchar(5)	기관구분
UD_Date	varchar(10)	등록날자(2003/10/01)
Ud_Time	varchar(11)	등록시간(10:11:12)
UD_Mgr	varchar(32)	관리자이름

3.1.2 로그정보 생성 및 수집

어카운팅 시스템에서 로그생성 방법은 유닉스 나 리눅스 운영체제에 따라 차이가 있으나 본 논문에서는 리눅스 어카운팅 정보 생성 및 수집 방법을 선택하여 적용하였다.

리눅스에서 어카운팅은 프로세스에 대한 정보를 기록하는 pacct(표 2 참조)와 connection 정보를 보관하는

wtmp(표 3 참조), 그리고 배치작업(PBS : Portable batch system)의 로그를 보관하는 accounting(표 5 참조) 등이 있으며, 또한 각 사용자별 디스크 사용량을 체크하기위해 쿼터를 적용할 수 있으나 같은 사용자 정보가 여러 파일시스템에 분산되어 기록되었을 경우 통합 정보 생성의 어려움이 있기 때문에 각각의 디스크에 흩어져있는 사용자의 디스크 이용정보를 생성하기 위해 전체 파일시스템의 모든 사용정보를 파악할 수 있는 풀-스캔(Full-Scan) 방법 적용 하였다.

[표 7] 어카운팅 로그 파일 위치

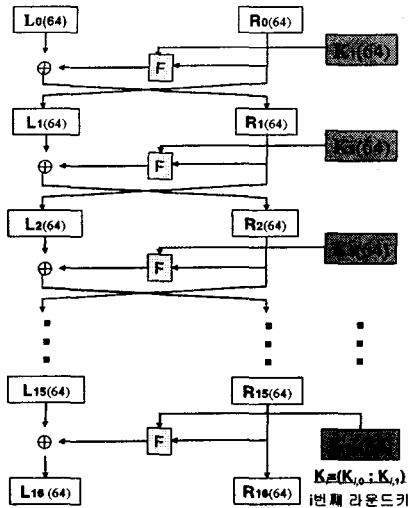
로그 위치 정보	설명
/var/log/wtmp	connection 어카운팅 정보
/var/account/pacct	프로세스 어카운팅 정보
/var/account/hdd_used	디스크 어카운팅 정보 (스크립트 이용)
/usr/spool/PBS/server_priv/accounting	PBS 어카운팅 정보

본 논문에서는 4가지 로그정보를 자동으로 통합하고 관리하는 모듈을 개발하여 정보를 수집하여 MySQL에 저장하도록 하였다.

3.2 암호 알고리즘 적용

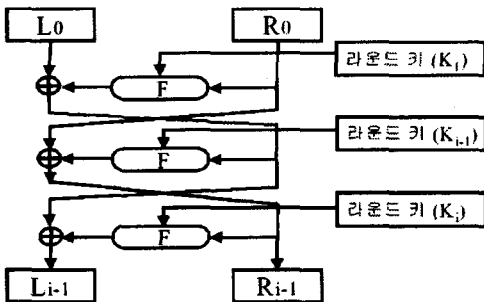
3.2.1 SEED 적용

SEED 알고리즘의 전체 구조는 Feistel 구조[그림 5 참조]로 이루어져 있으며, 128비트의 평문 블록단위당 128비트 키로부터 생성된 16개의 64비트 라운드 키를 입력으로 사용하여 총 16라운드를 거쳐 128비트 암호문 블록을 출력한다. 다음 그림 4와 같은 SEED 알고리즘을 적용하여 로그정보를 암호화 하였다



▶▶ 그림 4. SEED 전체 구조도

그림 4에서 보는바와 같이 SEED 알고리즘은 128비트 입력 평문블록을 2개의 64비트 블록 ($L_0(64), R_0(64)$)으로 나누어, 16개의 라운드 키(64비트)를 이용하여 16라운드를 순차적으로 수행한 후, 최종 128비트 암호문 블록을 출력 ($L_{16}(64), R_{16}(64)$)한다.



▶▶ 그림 5. Feistel 구조

이러한 Feistel 구조는 간단하게 디자인된 동일한 구조를 여러 번 반복함으로써 구현이 용이하고 복잡도를 증가시킬 수 있는 장점을 가지고 있으며 블록암호 알고리즘인 SEED의 기본 구조를 이루고 있다. 이처럼 SEED 알고리즘의 수행속도가 빠른 장점이 있어서 전체적인 시스템 성능을 고려하여 어카운팅 시스템을 설

계하였다.[1]

3.2 MySQL 암호·복호화 적용

클러스터 노드에 적재된 어카운팅 데이터를 MySQL 데이터베이스로 적재시 클러스터 노드로부터 한 레코드씩 읽어서 서버타입, 사용자이름, 날짜 필드를 복호화 한 후에 사용자 관리 테이블에서 해당 사용자 이름을 찾아서 Primary Key를 생성하고, 서버타입, 사용자이름, 날짜를 MySQL 암호화 함수를 이용하여 1차 암호화 시키고, 이 암호화 시킨 salt값을 다시 2차로 암호화를 시키는 구조를 갖고 있으며 그 형식은 [그림 3]과 같다.

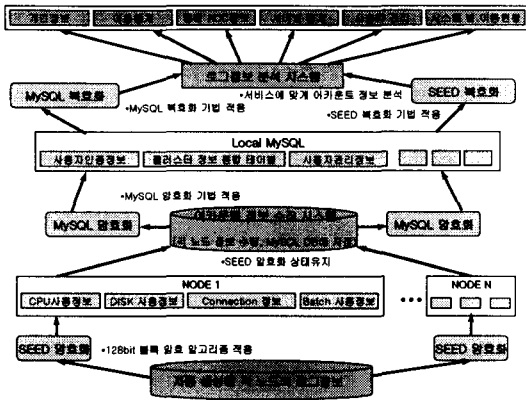
3.3 Web 기반 계정관리 및 통계

클러스터 시스템 이용에 대한 통계정보를 실시간으로 확인하고 접근할 수 있도록 Web 기반어카운팅 시스템이 개발 되어야 하며 효율적인 관리를 위해서는 표 8과 같은 기능을 갖는다.

[표 8] 어카운팅 시스템 기능

기능	설명
개인정보관리	개인정보 등록, 수정
사용자별 이용현황	1. CPU 사용량 2. 배치(Batch) 사용량 3. 디스크 사용량 4. 연결(Connection) 시간
시스템별 이용현황	1. CPU 사용량 2. 배치(Batch) 사용량 3. 디스크 사용량 4. 연결(Connection) 시간 5. 사용자, 기관, 분야별 사용량 6. 기간별 사용량 7. 각 클러스터별 사용량

WEB 기반 계정관리 시스템의 처리과정을 살펴보면, 검색을 요청하면 사용자 관리 테이블에서 한 레코드를 읽어서 salt2를 이용하여 salt1을 복호화 시키고 salt1으로 SEED 암호 처리된 부분을 제외한 나머지 부분을 다시 복호화해서 검색 요구사항에 맞는 사용자를 찾는다. 해당 SEED 암호화 부분을 복호화해서 요구사항을 처리한 후 최종 사용자에게 Display 시키는데 이 과정을 도식화 하면 그림 6과 같다.



▶▶ 그림 6. 웹 기반 어카운팅 시스템 데이터 흐름도

4. 결론

방대한 양의 정보를 계산하고 처리하기 위해 클러스터 시스템과 같은 초대형화 컴퓨팅 자원이 필요하게 되었다. 그러나 이러한 인프라 자원을 효율적으로 관리하고, 안전성을 보장하기 위해서는 다음과 같은 것들을 고려해야 한다.

첫째, 시스템에 대한 신뢰성 보장을 통한 사용자 확보 및 정보 유출을 방지함. 둘째, 타인의 시스템 자원을 임의의 사용을 금함으로서 형평성 있는 요금 부과를 원칙으로 함. 셋째, 실시간 시스템 자원 사용 현황 파악과 효율적인 시스템 자원의 활용을 위한 사용자 계정관리 체제 확립, 넷째, Web 기반 관리시스템 개발을 통해 사용자 접근성과 자원 관리의 편의성 등을 제공해야 한다.

따라서, 본 논문에서 제안한 클러스터 어카운팅 시스템은 보안을 고려한 128bit 국내 표준 암호 알고리즘인 SEED 기법을 적용하여 기존 어카운팅 시스템에서 고려되지 않은 정보의 비밀성에 대한 문제점을 보완하고 안전성을 향상 시킬 수 있도록 대칭키 암호 기법을 적용함으로써 정보의 비밀성이 향상되도록 설계 하였다. 또한, 클러스터 시스템 구조에서 산재된 여러 노드에 대한 통합 어카운팅 방법론을 제시하였고, 모든 어카운팅 로그정보들의 표준화와 웹 기반 통합 계정관리 시스템을 통해 다른 클러스터의 추가, 변경, 공동사용 등의 편리성 제공이 가능할 것이다.

참고문헌

- [1] 한국정보통신기술협회(TTA), "TTA.KO-12.0004 : 128 비트 블록암호알고리즘 표준", 1999.9
- [2] Seung-Jo Han, Heang-Soo Oh, Jongan Park, "The improved data encryption standard (DES) algorithm", Spread Spectrum Techniques and Applications Proceedings, 1996., IEEE 4th International Symposium on , Volume: 3 , 22-25 Sept. 1996 Page(s): 1310 -1314 vol.3
- [3] Robshaw, M.J.B. "Security estimates for 512-bit RSA", WESCON/'95. Conference record. 'Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies', 7-9 Nov. 1995 Page(s): 409
- [4] 엄성용, 이규원, 박선화, "SEED 블록 암호 알고리즘의 파이프라인 하드웨어 설계", VOL. 30 NO. 03 pp.0149-0159 2003 . 04
- [5] 최재웅, 김성천, "서버 클러스터에서의 인터넷 서비스를 위한 효율적인 연결 스케줄링 기법 ", VOL. 30 NO. 07 pp.0352-0358 2003.08
- [6] Aiko Pras, Bert-Jan van Beijnum, Ron Sprenkels, and Robert Parhonyi, University of Twente, "Internet Accounting", IEEE Communications Magazine, 2001.