JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 19, No. 3 : 529~536 Mar. 2015

암호화 성능 향상을 위한 다중장비 기반 분산 병렬 처리 방법

김현욱* · 박성은 · 어성율

A Method of Distributed Parallel Processing based on Multi-Server for Improving Encryption Performance

Hyun-wook Kim* · Sung-eun Park · Sung-yul Euh

Research Institute, KSIGN CO.,LTD., Seoul 135-515, Korea

요 약

최근 개인정보보호법의 시행으로 개인 정보를 암호화하여 저장하는 메커니즘이 보안 시스템에 적용되고 있다. 개인 정보에 대한 암 복호화 메커니즘을 적용할 경우 초기에 기 저장되어 있는 대용량의 개인 정보를 암호화해야 한다. 이때 서버의 자원 부족이 발생할 수 있다. 또한 많은 시간이 소요된다. 본 논문에서는 위와 같은 문제점을 해결하고자 저 사양 다중 장비를 사용하여 대량의 개인 정보를 분산 병렬처리로 암호화하는 방법을 제안하고 테스트 환경을 구축하여 성능을 측정하였다. 그리고 고 사양 장비의 성능과 비교하였다. 측정 결과 장비를 3대로 확장하여 분산 병렬처리를 수행하는 경우 약 128% 이상, 5대로 확장하였을 경우 158% 이상 성능이 향상되는 것을 확인할 수 있었다.

ABSTRACT

As personal information protection act was recently enforced, a mechanism which saves encrypted personal information has been used to Information Security systems. To use the mechanism, a millions of personal information which are already saved on the system first have to be encrypted. At the moment, it may cause a resource scarcity on server, and also take a lot of time. Thus, this paper suggests a way to encrypt millions of personal information by using multi-server with low specifications and measures its performance on test environment. And , I was compared with the performance of high- specification server. As a compared result, the mechanism with three devices by parallel and distributed processing improved its performance by 128%, and the mechanism with five devices by the same processing improved its performance by 158%.

키워드 : 암호화, 분산처리, 병렬처리, 정보 보안, 토큰화

Key word : Encryption, Distributed processing, Parallel processing, Information security, Tokenization

접수일자 : 2014. 11. 18 심사완료일자 : 2014. 12. 22 게재확정일자 : 2015. 01. 09

* **Corresponding Author** Hyun-Wook Kim(E-mail:khw@ksign.com, Tel:+82-02-564-0182)
Research Institute, KSIGN CO.,LTD., Seoul 135-515, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2015.19.3.529 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

개인 정보 유출 사고가 증가하고 개인정보보호법 시행 등 정보 보안의 중요성은 계속 증가하고 있으며, 개인정보보호법이 2011년 3월 29일 제정되어 2012년 3월 20일부터 개정되어 시행되었다. 이에 개인 정보를 관리하는 DB 서버에 보안 시스템이 갖추어지고 있다[1].

최근에는 개인 정보의 수집과 이용이 보편화됨에 따라 개인 정보의 수집 범위가 크게 증가하고, 데이터베이스가 대용량화되는 추세이다[2].

그러나 동시 트랜잭션이 많은 환경을 가지거나 대용량 데이터베이스 시스템에서의 암 복호화와 같은 개인정보 보호 메커니즘을 적용하는 것은 성능 저하의 원인이 된다[3].

성능 저하를 막기 위해서는 암 복호화를 수행하는 서버의 사양을 높이는 방법(Scale-Up)이 주를 이루고 있다. 하지만 고가의 예산, 서버 이관 작업등의 어려움이 있다.

기존 웹 서버 시스템의 경우 대량의 네트워크 트랜잭션을 처리하기 위해 확장성을 보장하는 클러스터를 구축하여 성능 향상 및 안정성을 높이는 기술이 연구되고 있다[4].

본 논문에서는 여러 대의 암호화 서버로 구성되는 다중 장비 기반의 분산 병렬 처리 시스템을 제안한다. 또한 특정 서버에 부하를 가중시키지 않고 효율적으로 작업을 분할하여 병렬 처리하는 방법을 제안한다. 그리고 대량의 개인 정보에 대한 암호화 성능을 측정하였다.

본 논문의 구성은 2장에서 기존 분산 처리 방법에 대한 사전 연구를 수행하고, 3장에서 다중 장비를 이용한 암 복호화 분산 병렬 처리 방법과 시스템을 설계하고, 4장에서는 테스트를 통해 성능을 고 사양 시스템과 본 논문에서 제한한 분산 병렬처리 시스템과의 성능을 비교 측정하고, 마지막으로 4장에서 결론을 맺는다.

Ⅱ. 관련 연구

기존 분산 병렬처리에 사용되는 기법으로는 Round Robin, Load Balancing, Active Standby, Divide Equally 등이 있다.

Round Robin기법은 복수개의 서버에 순차적으로 균등하게 작업을 할당하는 방법이다[5]. Round Robin방법은 부하 분산에 대한 효과는 있으나 작업에 대한 분할이 이루어지지 않기 때문에 병렬처리에 효과적이지 못하다. 또한 기존에 부하가 걸려있는 서버에 작업이 할당될 경우 해당 서버의 부하를 가중시킬 수 있는 단점이 있다.

Load Balancing은 작업을 일정 단위로 처리하는 방식으로 부하 분산에 효과가 크지만 스위칭 후 데이터를 재조립해야 하는 추가 작업이 발생하는 단점을 가진다[5].

Active standby 방식은 Master-Slaves 형태의 이중화 개념[6]으로 Fault-Tolerance를 만족시킬 수 있지만 부하 분산 효과가 없다.

Divide Equally는 사용 가능한 서버 수만큼 작업을 분할하여 할당하는 방법으로 부하 분산 및 병렬 처리에 효율적이지만, 특정 서버에 부하가 걸려있을 경우 부하가 가중되어 작업 처리시간이 증가한다는 단점이 있다.

Ⅲ. 분산 병렬 암 복호화 처리 방법 및 시스템

분산 병렬 암호화 처리를 위해서 제안하는 다중 장비 기반 분산 병렬처리 시스템은 아래 그림 1과 같이 전위(font-end)에 위치하는 코디네이터(Coordinator)와 후위(back-end)에 위치하는 복수개의 토큰 서버(Token Server)로 구성되는 클러스터로 구조를 가진다.

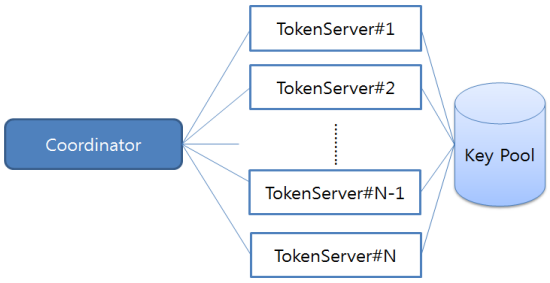

그림 1. 분산 병렬처리 시스템 구조
Fig. 1 Distributed Parallel System Structure

코디네이터는 암 복호화 작업(이하 Task)에 대한 입출력과 후위에 배치된 토큰 서버로의 Task 할당과 및

결과 취합 등의 역할을 수행한다.

토큰 서버는 코디네이터로부터 할당받은 Task를 수행한다. 모든 토큰 서버는 Task를 수행하기 위한 암호키를 풀(Pool)로 공유하는 구조를 가진다.

3.1. 코디네이터(Coordinator) 설계

코디네이터는 요청 Task에 대한 입·출력을 담당하는 노드로 수시로 토큰 서버들의 상태 값을 받아 작업부하(이하 Workload)를 계산하고, Task의 입력이 발생할 경우 계산된 Workload에 따라 Task를 실행하기 위한 토큰 서버를 선택하고, Task를 분할하여 토큰 서버로 Task를 할당한다. 그리고 처리 완료된 Task에 대한 취합을 수행하여 Task를 요청한 클라이언트로 결과를 전송해 주는 역할을 한다.

코디네이터는 아래 그림 2와 같이 Task를 관리하는 Task Management, 각 토큰 서버들의 부하 상태 값을 가지는 Token Server Workload Pool, 토큰 서버의 부하 상태 정보를 수집하여 Token Server Workload Pool에 반영하는 Token Server Workload Management로 구성된다.

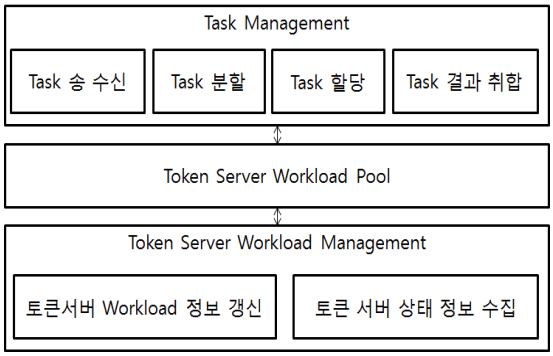

그림 2. 코디네이터 구조
Fig. 2 Coordinator Architecture

Token Server Workload Management는 토큰 서버로부터 실시간으로 상태정보(CPU Usage, IO Usage, Memory Usage)를 수집하여 Workload 계산하고, 이를 Token Server Workload Pool에 반영하는 작업을 반복한다. 토큰 서버의 상태정보는 SNMP(Simple Network Management Protocol)[7]를 기반으로 일정 간격으로 수신 받는다.

Workload 계산식은 아래 식 1과 같다.

$$Workload = (CPU\text{사용률}*0.7) + (IO\text{사용률}*0.2) + (MEM\text{사용률}*0.1) \quad (1)$$

계산된 Workload는 Task Management에서 Task를 할당하기 위한 토큰 서버를 선택하는데 사용된다.

Task Management에서는 Task를 수신하게 되면 Token Server Workload Pool에서 Workload값이 적은 토큰 서버를 선택한다.

선택되는 토큰 서버의 수는 최소 1개 이상이며, 아래 식 2를 만족하는 모든 토큰 서버를 Task를 수행하는 대상이 된다.

$$TS\text{노드 } Workload \leq Workload\text{평균} + workload\text{표준편차} \quad (2)$$

Task를 수행하는 토큰 서버가 선택되고 난 후에는 아래 그림 3과 같이 선택된 토큰 서버 수와 동일하게 Task를 분할하고, Task를 각 토큰 서버로 할당한다.

만약 Task의 크기가 작아 분할할 수 없을 경우 Lowest Workload 토큰 서버 순서로 Task가 할당된다.

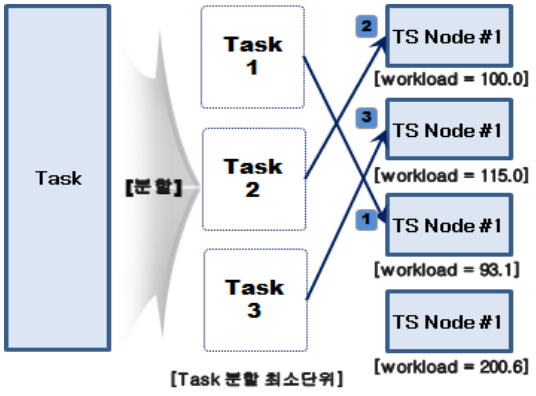

그림 3. 분산 병렬처리 방법
Fig. 3 Distributed parallel processing method

만약 수행되지 않거나 실패된 분할된 Task는 다른 토큰 서버에서 재처리하는 Fail-Over 기능을 수행한다.

3.2. 토큰 서버(Token Server) 설계

토큰 서버는 코디네이터로부터 할당받은 Task를 처리하는 서버로 N 대의 클러스터로 구성되어 있다. 동일

한 클러스터에 종속된 토큰 서버들은 암호 키를 풀(Pool)로 공유하고 있다.

토큰 서버에서 개인 정보에 대한 암 복호화는 [8, 9]에서 제안하는 블록 토큰 기법을 사용한다. 1개의 토큰 서버는 2개 이상의 클러스터에 종속될 수 없다.

Ⅳ. 테스트 환경 구성 및 성능 테스트

4.1. 테스트 환경 구성

테스트 환경 구성은 아래 그림 3과 같이 코디네이터 1대, 토큰 서버 5대로 구성하였다. 코디네이터에서는 데이터를 생성하여 암호화 요청 Task를 생성하는 Client S/W Module을 탑재하였다.

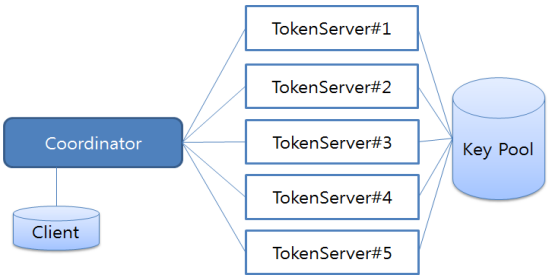

그림 4. 분산 병렬 처리 테스트 환경 구성
Fig. 4 Distributed Parallel Processing Test environment

각 노드는 가상 머신(VM, Virtual Machine)으로 구성되어있으며, Hypervisor로는 KVM(Kernel-based Virtual Machine을 사용하였다.

각 VM의 사양은 아래 표 1과 같다.

표 1. 가상 머신 사양
Table. 1 Virtual Machine specification

항목	사양
CPU	2vCore
Memory	4GB
NIC	1vNIC
O/S	CentOS6.5 minimal(64bit)

본 논문에서 제안하는 분산 병렬 처리 성능을 비교하기 위해 고 사양 서버로 구성된 단일 시스템은 아래 그림 5와 같이 구성하였으며, 토큰 서버의 사양은 아래 표 2와 같다.

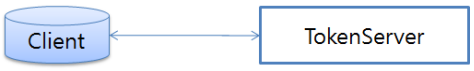

그림 5. 고 사양 단일 시스템 테스트 환경 구성
Fig. 5 High-Spec Single System Test Environment

표 2. 고 사양 단일 토큰 서버 사양
Table. 2 High-Spec single Token Server specification

항목	사양
CPU	16Core
Memory	32GB
NIC	10/100/1000Mbps
O/S	CentOS6.5 minimal(64bit)

4.2. 테스트 수행 방법

분산 병렬처리에 이용되는 단일 장비의 배치 데이터 처리 성능 수치를 확인하기 위한 테스트와, 토큰 서버 증가에 따른 배치 데이터 병렬 분산 처리 성능 수치를 확인하는 방법으로 진행하였다. 그리고 고 사양 토큰 서버에서의 암호화 성능을 측정하여 비교하였다.

테스트를 위해 사용된 데이터는 주민번호와 같은 13자리 숫자로 '000000-0000000'부터 '999999-9999999'까지 유일한 값으로 구성된다.

테스트 데이터 중 암호화 대상은 마지막 6자리로 정의하였으며, 암호화 방식으로는 암호화 대상 데이터의 길이가 변하지 않고 대체 값을 생성하는 Tokenization 기술인 Block Token[8, 9]을 사용하였다.

성능 측정은 1만 건부터 10만 건까지는 1만 건 단위로 데이터를 생성하여 수행하였고, 10만 건부터 100만 건까지는 10만 건 단위로, 100만 건부터 500만 건까지는 100만 건 단위로 수행하였다.

성능 데이터로는 데이터 수에 대한 암호화 시간(th.sec)과 초당 처리 가능한 데이터 수(dps, Data Per Seconds)를 추출하였다. 계산 아래 식 3과 같다.

$$\begin{aligned} th.\text{sec} &= \text{코디네이터와토큰서버간} networktriptime \\ &\quad + TS\text{노드에서의암호화시간} \\ dps &= datacount/th.\text{sec} \end{aligned} \tag{3}$$

4.3. 테스트 결과

테스트 수행은 데이터 단위별로 5회씩 수행하였다. 테스트 수행 결과의 평균은 아래 표 3과 같다.

표 3. 성능 측정 결과
Table. 3 Performance measurement results

data count	고 사양 토큰 서버		토큰 서버 1대		토큰 서버 2대		토큰 서버 3대		토큰 서버4대		토큰서버 5대	
	th.sec	dps	th.sec	dps	th.sec	dps	th.sec	dps	th.sec	dps	th.sec	dps
10,000	0.058	172,414	0.084	118,878	0.056	178,571	0.046	217,391	0.043	232,558	0.032	312,500
20,000	0.124	161,290	0.166	120,417	0.123	162,602	0.090	222,222	0.084	238,095	0.066	303,030
30,000	0.178	168,539	0.240	125,333	0.166	180,723	0.156	192,308	0.138	217,391	0.114	263,158
40,000	0.233	171,674	0.325	123,588	0.204	196,078	0.204	196,078	0.184	217,391	0.168	238,095
50,000	0.287	174,216	0.404	123,921	0.288	173,611	0.247	202,429	0.214	233,645	0.253	197,628
60,000	0.345	173,913	0.500	121,020	0.345	173,913	0.266	225,564	0.268	223,881	0.269	223,048
70,000	0.400	175,000	0.640	114,941	0.405	172,840	0.326	214,724	0.317	220,820	0.311	225,080
80,000	0.455	175,824	0.634	126,286	0.455	175,824	0.328	243,902	0.308	259,740	0.353	226,629
90,000	0.513	175,439	0.705	127,612	0.522	172,414	0.421	213,777	0.409	220,049	0.395	227,848
100,000	0.569	175,747	0.784	127,625	0.580	172,414	0.432	231,481	0.493	202,840	0.394	253,807
200,000	1.139	175,593	1.578	126,762	1.130	176,991	0.981	203,874	0.870	229,885	0.776	257,732
300,000	1.710	175,439	2.355	127,380	1.739	172,513	1.489	201,478	1.542	194,553	1.097	273,473
400,000	2.277	175,670	3.145	127,181	2.165	184,758	2.015	198,511	1.612	248,139	1.462	273,598
500,000	2.844	175,809	4.048	123,584	3.065	163,132	2.576	194,099	2.386	209,556	2.056	243,191
600,000	3.551	168,966	4.698	127,733	3.287	182,537	2.800	214,286	2.190	273,973	2.311	259,628
700,000	4.105	170,524	5.500	127,294	4.026	173,870	3.276	213,675	3.232	216,584	2.595	269,750
800,000	4.591	174,254	6.299	127,005	4.594	174,140	3.821	209,369	2.973	269,088	3.136	255,102
900,000	5.157	174,520	7.065	127,409	5.339	168,571	4.405	204,313	4.122	218,341	3.383	266,036
1,000,000	5.728	74,581	7.861	127,214	5.708	175,193	4.490	222,717	3.867	258,598	3.904	256,148
2,000,000	11.448	174,703	15.726	127,187	10.834	184,604	9.370	213,447	7.872	254,065	6.943	288,060
3,000,000	17.204	174,378	23.606	127,100	16.418	182,726	13.876	216,201	12.739	235,497	10.580	283,554
4,000,000	23.282	171,807	31.596	126,607	21.203	188,653	16.096	248,509	17.314	231,027	13.933	287,088
5,000,000	30.102	166,102	40.148	124,652	27.746	180,206	25.115	199,084	22.775	219,539	20.540	243,427

측정 결과 표 1과 같은 사양을 가지는 토큰 서버를 기준으로 표 2와 같은 고 사양 토큰 서버와 비슷한 성능을 위해서는 저 사양 토큰서버 2대 이상이 필요하다는 것을 확인 할 수 있다.

100만 건 이상의 개인정보 데이터를 암호화 시 토큰 서버 2대를 사용하여 분산 병렬 처리하는 것이 처리 시간이 단축되는 것을 확인할 수 있었다. 데이터 수에 대한 암호화 시간을 기준으로 하는 암호화 효율은 아래 그림 6과 같다.

처리 효율을 계산하는 식은 아래 식 4와 같다.

$$\text{처리 효율} = \text{고사양토큰서버} th.\text{sec} \div \text{분산병렬처리} th.\text{sec} \times 100\% \quad (4)$$

표 2와 같은 사양의 토큰 서버 암호화 처리 효율 대비 표 1과 같은 사양의 토큰 서버 1대를 사용하였을 경우 약 72%의 처리 효율을 가지는 것으로 측정되었다.

토큰 서버 2대를 사용하였을 경우 약 102%, 3대를 사용하였을 경우 약 123%, 4대를 사용하였을 경우 약 134%, 5대를 사용하였을 경우 약 149%로 처리 효율이 향상되는 것을 확인하였다. 토큰 서버 증설에 따른 분산 병렬 암호화 처리 시간은 아래 그림 7과 같다. 데이터의 크기가 증가하였을 때, 토큰 서버 대수에 따라 암호화 시간이 크게 단축되는 것을 확인할 수 있다.

아래 그림 8에서는 토큰 서버 증설에 따른 초당 데이터 처리량(DPS, Data Per Second)을 비교하였다.

초당 데이터 처리량은 그림 8과 같이 표 3과 같은 고 사양 장비에서는 약 172,887건으로 측정되었으며, 표 2와 같은 장비 1대에서 처리량은 약 125,075건, 2대에서 분산 병렬 처리하였을 경우 약 176,821건, 3대에서 분산 병렬 처리하였을 경우 약 213,019건, 4대에서 분산 병렬처리하였을 경우 약 231,533건, 5대에서 분산 병렬 처리하였을 경우 약 257,722건으로 측정되었다. 고 사양 토큰 서버 대비 다중 토큰 서버를 이용한 분산 병렬 처리 효율에 대한 분석 결과는 아래 표 4와 같다.

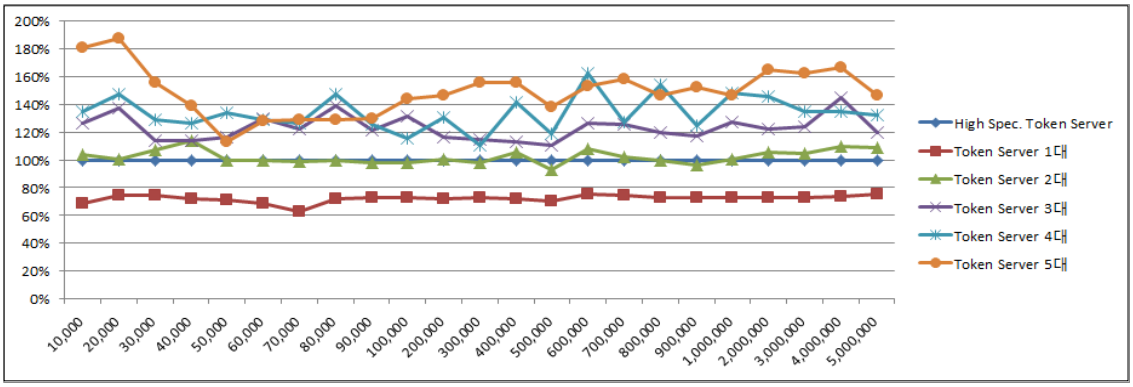

그림 6. 분산 병렬 암호화 처리 효율 비교
Fig. 6 Comparison of processing efficiency for distributed parallel encryption processing

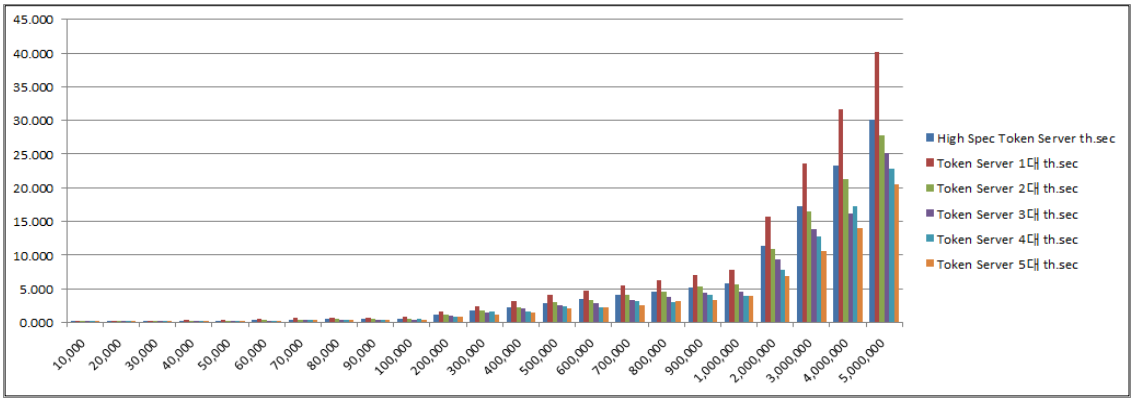

그림 7. 분산 병렬 암호화 처리 시간 비교
Fig. 7 Comparison of distributed parallel encryption processing time

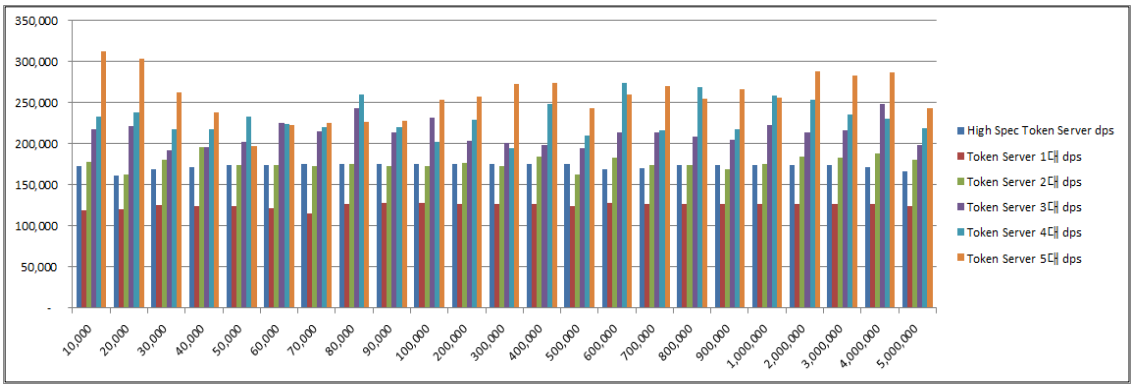

그림 8. 분산 병렬 처리에 따른 초당 데이터 처리량 비교
Fig. 8 Comparison of Data Per Second from distributed parallel encryption processing

표 4. 분산 병렬 처리 테스트 분석 결과
Table. 4 Encryption Performances analytics result

구분	1대	2대	3대	4대	5대
평균	72%	102%	123%	134%	149%
표준 편차	3%	5%	9%	12%	17%
신뢰구간 MIN(99%)	71%	100%	119%	127%	140%
신뢰구간 MAX(99%)	74%	105%	128%	141%	158%

성능 분석 결과 위 표 4와 같이 표 3과 같은 고 사양 장비 대비 표 2와 같은 저 사양 장비 2대를 사용하여 분산병렬처리하였을 경우 약 100 ~ 105%, 3대를 사용하였을 경우 약 119 ~ 128%, 4대를 사용하였을 경우 127 ~ 141%, 5대를 사용하였을 경우 140 ~ 158% 의 성능 향상 효과가 있음을 확인하였다.

V. 결 론

본 논문에서는 저사양의 다중 토큰 서버를 이용한 암호화 분산 병렬 처리 방법을 제안하고 테스트를 통해 토큰 서버를 증설하여 분산 병렬 처리할 경우 성능이 향상됨을 확인하였다. 본 논문에서 제안하는 분산 병렬 처리 방법 및 시스템은 토큰 서버의 작업부하(Workload) 값을 기준으로 분산 병렬 처리를 수행하기 때문에 특정 토큰 서버의 부하를 가중시키지 않는다는 장점이 있다. 이에 안정적으로 대량의 개인 정보를 빠르게 암 복호화 할 수 있다. 토큰 서버를 Scale-out 형식으로 확장하여 사용할 수 있기 때문에 기 구축된 개인정보 보호 시스템을 수정하지 않고 확장할 수 있다는 장점이 있다.

본 논문에서 제안하는 다중 노드 기반의 분산 병렬 처리 방법은 암호화 처리 시스템에 제한되지 않고 클라우드 환경을 기반으로 Scale-out이 필요한 많은 시스템에 응용되어 사용될 수 있을 것으로 기대한다.

지속적인 연구를 통해 전위에 위치한 코디네이터 분산처리를 통한 성능 개선에 대한 연구를 수행할 것이다. 또한 관련 연구에서 설명한 기존 병렬 분산 처리 방법들과의 성능 비교를 진행할 예정이다.

감사의 글

This work was supported by the IT R&D program of MSIP/IITP. [10041579, Development the Personal Information Security service solution using tokenization technology]

REFERENCES

[1] M. S. Seo, D. W. Park, "A Study on Quantitative Security Assessment after Privacy Vulnerability Analysis of PC", *Proceedings of the Korean Institute of Information and Communication Sciences Conference*, Busan, pp 456-460, 2012.

[2] H. W. Kim, S. E. Park and S. Y. Euh, "The Distributed Encryption Processing System for Large Capacity Personal Information based on MapReduce", *Journal of the Korea Institute of Information and Communication Engineering*, vol. 18, no. 3, pp. 576-585, Mar.2014.

[3] J. W. Kang, "A Study of Effective Privacy Protection System on High Concurrent Transaction Database System", *Jouranl of Information and Security*, vol.12, no. 2, pp. 107-113, May, 2012.

[4] G. C. Park, K. Sung and S. S. Kim, “Traffic Distributed Processing System Implementation on the Web Server Networking", *Journal of the Korea Institute of Information and Communication Engineering*, vol.8, no.4, pp. 846-853, Aug, 2004.

[5] K. Sung, S. S, Kim, and G. C. Park, "A high speed processing method of web server cluster through rout robin load balancing", *Journal of the Korea Institute of Information and Communication Engineering*, vol.8, no.7, pp. 1524-1531, Nov, 2004.

[6] J. W. Choi, “Foundation Techniques and Fault-tolerance Tests of Active-Active Duplicated Domain Name Servers”, *Journal of the Korea Institute of Information and Communication Engineering*, vol.17, no.1, pp. 90-100, Jan, 2013.

[7] HISTORIC Network Working Group, A Simple Network Management Protocol (SNMP) [Internet]. Available: https://tools.ietf.org/html/rfc1157

[8] S. Y, Euh et al, *METHOD FOR DESIGNING BLOCK TOKENS BY USING DATA BLOCKING,* KR-A-1020130046604, to KSIGN, Patent and Trademark Office, Seoul, 2014.

[9] K. K. Lee,"Design of block token scheme for improving encryption performance in database", M.S. Theses, Soongsil University, 2014.

김현욱(Hyun-Wook Kim)

2009년 : 광운대학교 컴퓨터공학과 공학사
2013년 : 광운대학교 일반대학원 전자공학과 석 · 박사통합과정 수료
2013년~현재 : (주)케이사인 정보보안연구실 주임연구원
※관심분야 : 개인정보보안, 클라우드 컴퓨팅, 하둡, 기계학습, 분산처리

박성은(Sung-Eun Park)

1998년 : 아주대학교 컴퓨터공학과 공학사
2000년 : 아주대학교 컴퓨터공학과 석사
2002년 : 아주대학교 컴퓨터공학과 박사과정 수료
2009년~현재 : (주)케이사인 정보보안연구실 연구실장
※관심분야 : 클라우드 컴퓨팅, Virtual Appliance, Software-Defined Security

어성율(Seong-yul Euh)

1997년 : 아주대학교 컴퓨터공학과 공학사
1999년 : 아주대학교 컴퓨터공학과 석사
2001년~현재 : (주)케이사인 개발본부 총괄
※관심분야 : 클라우드 컴퓨팅, 빅데이터, 개인정보 암호화 처리