

블록체인을 활용한 개인 경력 관리 및 유통 시스템 설계

배수환*, 신용태**

Design of Personal Career Records Management and Distribution using Block Chain

Su-Hwan Bae*, Yong-Tae Shin**

요약 본 논문은 직원 채용 시 개인의 경력을 위조하여 취업하는 경우가 발생하는 문제점을 해결하기 위해 개인 경력 정보를 블록체인을 통해 관리하고 검증해 줄 수 있는 시스템을 제안한다. 블록체인 네트워크는 프라이빗 네트워크를 사용하며, 블록 내부에는 사용자의 학력과 경력 정보를 보관한다. 블록체인의 기능은 블록 생성, 블록 내부 데이터 검색, 경력 및 학력 검증의 기능을 수행하며, 이는 체인코드를 통해 동작한다. 제안하는 시스템의 성능평가 결과 트랜잭션 당 처리시간은 약 110ms, 검색 시간은 10ms로 측정되었으며, 실제 시스템에 적용하여 사용가능함을 확인하였다.

Abstract This paper proposes a system that can manage and verify personal career information through a block chain to solve the problem of getting a job by forging an individual's career when hiring employees. Blockchain network uses private network, and inside the block, the user's academic and career information is kept. The functions of the block chain perform the functions of block creation, block internal data retrieval, career and academic verification, which works through chain code. As a result of the performance evaluation of the proposed system, the processing time per transaction was measured at approximately 110 ms and the search time was measured at 10 ms, and it was applied to the actual system to confirm that it was available.

Key Words : Block chain, Data management, Data distribution, Personal career records, Recruitment system

1. 서론

4차 산업혁명시대로의 발전은 다양한 산업 분야에 수많은 변화를 가져왔다. 이런 변화를 반영하듯 각 산업에서 필요로 하는 인력에도 변화가 일어나고 있다. 각 기업들은 변화하는 기술에 대처하기 위해 새로운 직원을 채용하는 방안을 모색하고 있다[1]. 하지만 경력을 위조하여 위장 취업하는 문제는 계속하여 발생하고 있으며, 이를 해결하기 위한 경력을 검증하여 제공할 수 있는 시스템이 부족한 상황이다. 또한 각 기업도 이를 해결하기 위해 많은 시간과 비용을 소비하는 문제가 발생하고 있다.

본 논문에서는 개인의 경력 위조 문제를 해결하기 위해 블록체인을 활용한 개인 경력 관리 및 유통 시스템을 제안한다. 제안하는 시스템은 블록체인에 개인의 경력을 저장하고 이를 제공하는 형태로 개인의 경력을 검증하여 기업에 전달할 수 있는 방식을 사용한다. 블록체인 네트워크는 프라이빗 네트워크 형태로 동작한다. 네트워크를 구성하는 노드는 관리노드, 사용자노드, 엔터프라이즈 노드로 구성된다. 블록체인의 공유 원장에는 개인의 학력과 경력을 기록하며, 이를 카프카 알고리즘을 사용하여 검증하고 블록으로 생성하는 형태로 동작한다. 제안하는 시스템에서 체인코드를 사용

*Department of Computing, Soongsil University

**Corresponding Author : Department of Computing, Soongsil University (shin@ssu.ac.kr)

Received June 10, 2020

Revised June 17, 2020

Accepted June 19, 2020

하여 사용자의 학력과 경력을 블록에 저장하고, 블록체인에 저장되어 있는 개인 경력 및 학력을 지원 시 제출한 이력서와 대조하여 위조 여부를 확인할 수 있도록 한다.

본 논문의 구성은 2장에서 기존의 채용 플랫폼, 개인 경력 조회 시스템, 하이퍼레저 합의 알고리즘의 운영 방식에 대해 설명한다. 3장에서는 제안 기법의 구성과 경력 관리 및 유통 과정을 설명한다. 4장에서는 블록체인에 저장된 데이터를 검색할 때의 성능과 트랜잭션 처리 시간을 측정하여 제안기술의 성능을 평가한다. 마지막으로 5장에서는 본 연구의 결론과 제안된 시스템의 향후 연구 계획에 대해 기술한다.

2. 관련 연구

본 장에서는 제안하는 시스템의 기반이 되는 기존의 채용 플랫폼, NCS 채용 시스템, 블록체인, 체인코드, 하이퍼레저 합의 알고리즘에 대해 설명한다.

2.1 기존의 채용 플랫폼

기존의 채용 플랫폼은 사용자 자신이 개인의 학력과 경력을 직접 입력하고, 이력서를 작성하는 형태로 구성되어 있다. 특정 기업이 직원 채용공고를 하면 해당 기업에 작성한 이력서를 제출하는 절차를 진행하여 직원 모집을 실시한다. 이때, 기존의 채용 플랫폼에서는 개인 경력에 대해 추가적인 검증을 실시하지 않기 때문에, 악의적인 목적으로 개인의 학력과 경력을 위조하여 작성하여 기업에 제출하게 되면 위조된 사실은 기업에서 검증해야하는 문제가 발생한다.

또한 채용 분야에서 어떠한 일을 하는지 명확하게 인지하지 못한다는 문제가 존재한다. 이에 지원자가 불명확한 인지로 인해 잘못된 입사 지원을 할 수 있게 된다. 이로 인해 직원 채용 시 불필요한 비용과 시간이 발생하게 되어 효율적인 채용 절차를 진행하지 못하게 된다.

2.2 NCS 채용 시스템

NCS는 산업 현장에서 직무를 수행하기 적합한 인재를 선별하기 위해 요구되는 지식, 기술, 소양 등의

내용을 국가가 지정하여 체계화한 시스템을 일컫는다. NCS는 개인에게 반드시 요구되는 표준화 능력과 성취 여부를 판단할 수 있는 항목으로 구성되어 있다[2]. 또한, NCS 채용 시스템을 통해 채용 절차에서 실기 전형을 폐지하여 채용 비용, 재교육 비용, 생산성 제고 효과 등을 기대할 수 있게 되었다[3]. 그리고 익명 이력서를 통한 블라인드 채용을 실시하여 직무 능력 이외의 채용 담당자의 판단을 왜곡 할 수 있는 편견 요소를 배제시켜 필요한 인력을 채용할 수 있도록 하고 있다.

2.3 블록체인과 체인코드

2.3.1 블록체인

블록체인은 블록체인을 활용하여 복잡한 시스템을 단순화 하거나 혼동을 방지하기 위해 데이터를 동기화 하는데 사용된다[4]. 기존의 보안 방식과 블록체인의 차이점은 네트워크와 관련된 모든 사람들에게 동일한 정보를 제공함으로써 특정인이 데이터를 위조해도 전체 구성원이 알고 있는 정보와 다르기 때문에 인정받지 못한다는 것이다. 즉, 기존의 데이터 은닉과 다르게 공개를 통한 위조 방지를 수행하는 기술이다[5].

블록체인의 네트워크는 퍼블릭 네트워크와 프라이빗 네트워크로 구성되는데, 이는 아래 표 1과 같이 비교할 수 있다.

표 1. 퍼블릭 네트워크와 프라이빗 네트워크의 비교
Table 1. Compare of Public and Private Network

	Public	Private
Type of Participation	Everyone	Authorized User
Read Permission	Everyone	Authorized User
Chain Code Permission	Everyone	Authorized User
Permission management	Not used	Used
Example	Ethereum	Hyper-Ledger

2.3.2 체인코드

블록체인의 트랜잭션은 해시함수를 사용하여 허가된 경우에만 블록에 등록될 수 있다. 해당 과정에서 전

자서명을 사용하는데, 이를 사용하여 부인방지 기능을 제공하게 된다.

트랜잭션 처리를 위한 방식으로 주로 사용되는 방식은 체인코드를 사용하는 방식이다. 체인코드는 블록 생성뿐만 아니라 블록체인을 활용하여 다양한 서비스를 제공할 수 있는 기능을 제공하는데 사용된다. 체인코드는 시스템 체인코드와 사용자 체인코드로 분류되는데, 각 체인코드의 구성은 다음의 표 2의 내용과 같다[6].

표 2. 체인코드의 비교 및 구성
Table 2. Composition and Contents of Chain code

Classify	Title	Role
System	Configuration	Peer and Process channel configuration
	Life Cycle	Chain code's life cycle management
	Query	Block data query
	Endoser	Signing transaction execution results
	Validator	transaction validation
User	User	Depends on user's use

2.4 하이퍼레저 합의 알고리즘

하이퍼레저 패브릭은 다른 블록체인 플랫폼들과 다른 트랜잭션 처리 방식을 사용하고 있다. 다른 블록체인 플랫폼의 경우 거래내용을 검증하여 체인과 연동하는 방식을 사용한다. 이와 대조적으로 하이퍼레저 패브릭은 Execute, Order, Validate, and Update state의 절차로 동작한다[7]. 기존의 하이퍼레저 패브릭의 경우는 PBFT(Practical Byzantine Fault Tolerance) 방식을 합의 방식으로 사용하였다. 하지만 초당 트랜잭션(tps) 성능의 하락이 문제점으로 발생하게 되어 현재 Kafka 알고리즘을 사용하여 합의 절차를 진행하고 있다. 향후에는 SBFT(Simple BFT)를 사용하여 적용할 예정으로 연구하고 있다[8].

3. 블록체인을 활용한 개인 경력 관리 및 유통 시스템

본 장에서는 제안하는 기술인 블록체인을 활용한 개인 경력 관리 및 유통 시스템에 대해서 다룬다. 제안 기술의 구성과 절차, 각 기능의 동작을 위한 체인코드의 의사 코드를 통해 제안 기술을 설명한다.

3.1 제안하는 시스템의 구성

제안하는 기술에서 네트워크는 블록체인 관리자, 기업, 사용자의 3개의 그룹으로 분류된다. 블록체인 관리자는 Full 노드로서 네트워크의 모든 권한을 가지며 기업과 사용자는 Light 노드로 제한된 기능을 가진 상태로 네트워크에 참여한다. 아래의 그림 1은 제안하는 기법에서의 네트워크 구조를 나타낸다.

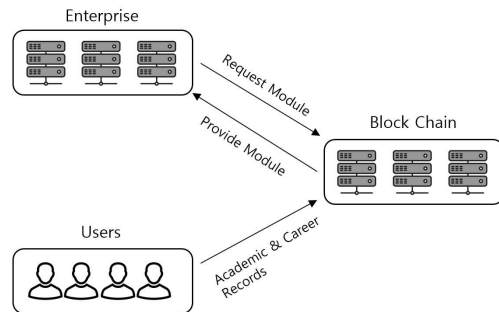


그림 1. 제안하는 시스템의 구조
Fig. 1. Structure of the proposed system

제안 기술에서의 블록 구성은 다음의 그림 2와 표 3과 같다.

		Number	Previous Hash	Data Hash	Block Header
Transaction Proposal		Version	Timestamp	Channel Id	Txid
		Chaincode path		Chaincode Name	Chaincode Version
		Chaincode Type	Input(chaincode function and arguments)		Timeout
		Endorser-1 Identity(certificat, public key)		Endorser-1 Signature	
		Endorser-2 Identity(certificat, public key)		Endorser-2 Signature	
		Endorser-N Identity(certificat, public key)		Endorser-N Signature	
Proposal Response		Read Set: <key, Version> read by Transaction			
		Write Set: List of <Key, Value> (UserID, Academic, Career)			

그림 2. 제안하는 시스템의 블록 구조
Fig. 2. Block structure of the proposed system

표 3. 제안하는 시스템의 블록 구성
Table 3. Block Structure of proposed system

Classify	Title	Role
Header	Block Number	Order of the block in the block chain
	Previous Hash Value	Attaches blocks to the chain using hash values from the previous block
	Data Hash Value	Validate the integrity of the transaction by entering the data hash value of the transaction
	Block Chain Version	Indicates the version of the current block chain
Transaction	User ID	Unique ID to identify users
	Academic Background	The academic background of a validated user with a verification node
	Career Records	The career records of users validated with the validation node

3.2 블록 생성 절차

제안 기법에서 블록 생성은 사용자가 블록체인 네트워크에 참여하거나 사용자의 학력과 경력을 변경할 때 수행된다. 그림 3은 블록 생성 절차를 나타낸다.

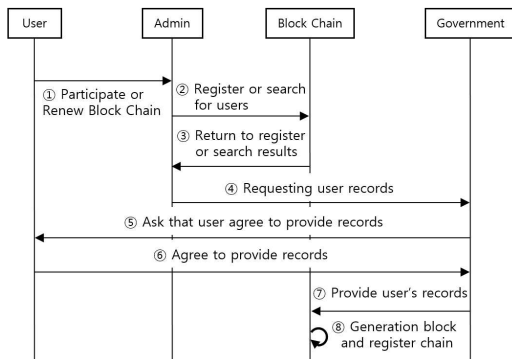


그림 3. 블록 생성 절차
Fig. 3. Process of create block

- 1) 사용자는 관리자에게 블록체인 네트워크에 참여하여 자신의 정보를 블록체인에 등록할 때와 기존의 정보 변경을 할 때 블록 생성을 요청한다.
- 2) 관리자는 등록 요청을 받으면 사용자가 제공해준 정보를 바탕으로 블록 생성 절차를 시작한다. 이때, 해당 사용자가 기존에 있었는지 여부를 확인하기 위해 블록체인에 해당 사용자가 등록되어 있는지 여부를 검색한다.

- 3) 관리자의 요청에 따라 사용자가 블록체인에 등록되어 있는지 여부를 확인하고 이를 관리자에게 전달한다.
- 4) 관리자는 사용자의 정보의 위조 여부를 확인하기 위해 사용자의 학력 및 경력을 정부에 정보를 요청한다.
- 5) 정부에서는 사용자에게 학력, 경력 정보를 제공해도 되는지 여부를 확인한다.
- 6) 사용자는 개인정보 제공 동의 여부를 전달하여 자신의 정보가 블록체인에 저장될 수 있도록 한다.
- 7) 정부에서는 사용자 정보 검증여부를 전달한다.
- 8) 블록체인에 사용자에게 대한 정보를 블록으로 생성하여 저장한다.

3.3 개인 학력 및 경력 검증 절차

제안 기법의 학력 및 경력 제공 절차는 사용자가 기업에 이력서를 제출하고, 해당 이력서의 내용이 위조되었는지 여부를 확인하기 위해 기업에서 요청하기 위해 사용된다. 그림 4는 경력 검증 절차를 나타낸다.

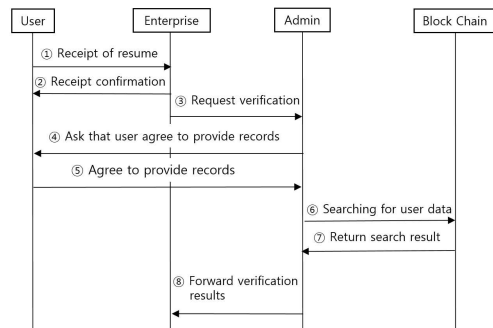


그림 4. 경력 정보 검증 절차
Fig. 4. Process of verifying career data

- 1) 지원자가 기업에 이력서를 제출할 때, 블록체인에 등록되어 있는 자신의 ID를 함께 제출한다.
- 2) 기업은 지원자의 이력서를 수령하고 수령 여부를 통지한다.
- 3) 기업은 블록체인 관리자에게 지원자의 이력서 내부의 학력과 경력의 위조 여부에 대한 검증을 요청한다.

청한다.

- 4) 블록체인 관리자는 지원자에게 블록체인 내의 경력 확인 동의 여부를 확인한다.
- 5) 지원자는 블록체인 관리자에게 본인의 데이터를 조회할 수 있도록 권한을 부여하는 메시지를 발송한다.
- 6) 블록체인 관리자는 체인코드를 실행하여 블록체인의 내부 데이터를 확인하고 이때 사용자의 고유 ID를 사용하여 사용자 데이터를 검색한다.
- 7) 블록체인 관리자는 블록체인에 등록된 데이터를 회사에 전달하여 위변조 여부를 확인할 수 있도록 한다.

3.4 체인코드의 동작

제안하는 시스템에서는 개인정보 제공 동의 여부, 블록 내부 데이터 검색, 블록 생성 및 수정, 경력 정보 검증을 위한 체인코드를 사용하여 기능을 구현하여 타 시스템과의 차별점을 구현하였다.

3.4.1 개인정보 제공 동의 여부 확인

사용자의 경력 및 학력 정보는 개인정보에 포함되는 사항이기 때문에, 사용자의 정보 동의 여부를 확인하는 절차가 필요하다. 해당 체인코드를 사용하는 주체는 사용자 정보의 검증을 도울 수 있는 정부에서 사용한다. 이때, 사용자에게 정보 제공 동의를 받는 방법은 메시지나 전자 우편의 형태로 동의 여부를 확인받을 수 있도록 한다. 사용자가 정보 제공에 동의하는 경우에만 블록 생성 및 검증 작업을 수행할 수 있도록 한다. 이는 그림 5의 체인코드와 같이 구현한다.

```

1 //User approval function
2 function getUserApprove(userData) {
3 //Create user object for
4   query approval request
5   User := User(userData)
6 //Request for inquiry approval
7   userApprove := User.requestApprove(data.enterprise)
8   if (userApprove == ture) {
9 //Grant permission if the user has agreed
10    return useApprove.approveInfo()
11 } else {
12 //Do note grant permission if use declined
13    return userApprove.rejectMessage()}}
    
```

그림 5. 개인정보 제공 동의 여부 확인 체인코드
Fig. 5. User approval function chain code

3.4.2 사용자 데이터 검색

사용자 검색에 사용되는 체인코드는 개인정보 제공 동의 확인 후 사용가능하다. 이는 실제로 블록체인 내부의 데이터를 얻는 작업을 수행하는 체인코드이기 때문이다. 데이터 검색 체인코드는 QueryString, txType의 두 가지 방식을 사용하여 검색을 수행 가능하다. QueryString 방식은 사용자 ID만을 사용하여 검색을 수행하는 방법이다. txType 방식은 사용자 ID와 추가적인 조건을 부여하여 검색을 수행하는 방식이다. 제안하는 시스템 내에서는 해당 체인코드를 사용하지 않는 경우에는 블록체인에서 정보를 확인할 수 없도록 구성되어 있다. 각각의 검색 방식에 대한 체인코드는 다음의 그림 6, 7과 같다.

```

1 //Generate a string to user for searching
2 queryString := "{\ selector\":{\
3   + column + \"\":\":" + value + \"\"}}\"
4 //Importing data from block chain
5 resultsIterator := Blockchain.getQueryResult(queryString)
6 //Convert data to JSON format
7 queryStrRes := resultsIterator.json()
8 if (queryStrRes != null) {
9 //Returns data if result is not null
10    return queryStrRes
11 } else {
12 //Error occurs if result is null
13    return error.Error("no results")}}
    
```

그림 6. QueryString 방식을 사용한 체인코드
Fig. 6. Chain code using QueryString

```

1 //Importing data from blockchain using keys
2 data = Blockchain.getState(key)
3 if (data != null) {
4 //Returns data if result is not null
5   return data
6 } else {
7 //Error occurs if result is null
8   return error.Error("no results")}}
    
```

그림 7. txType을 사용한 체인 코드
Fig. 7. Chain code using txType

3.4.3 블록 생성 및 수정

블록 생성 및 수정 체인코드는 실질적으로 블록의 내용에 관여하는 역할을 수행한다. 이 체인 코드는 사용자에게 의해서 실행된다. 사용자는 블록체인 관리자에게 자신이 블록체인 네트워크에 참여하거나 자신의 기존에 등록된 내용의 변경을 요청하기 위해 해당 체인코드를 사용한다. 블록체인 관리자는 사용자 블록의 존

재 여부를 확인하고 블록을 생성하거나 기존의 내용을 변경하는 역할을 수행하게 된다. 그림 8은 블록 생성 및 수정 체인코드이다.

```

1 //Join blockchain network or request data modification
2 Admin.requestParticipateOrRenew(userInfo)
3 //Register or search for block chain
4 results := Admin.registerOrSearch(userInfo)
5 //Request user unformation from certification authority
6 Admin.requestUserData(Government)
7 //Prompt user for query approval
8 userApprove := Government.getUserApprove(userInfo)
9 if (userApprove == true) {
10 //Provide data if user approves
11     userData := Government.getUserdata(userApprove)
12 //Recording information in blockchain
13     Blockchain.register(userInfo, userData)
14 } else {
15 //Return error if note approved by user
16     error.Error("user did not approve") }
    
```

그림 8. 블록 생성 및 수정 체인코드
Fig. 8. Create and modify function chain code

3.4.4 사용자 정보 검증

사용자의 학력과 경력을 검증하기 위해 사용하는 체인코드이다. 지원자가 기업에 이력서를 제출하고 해당 이력서 내용의 유효성을 검증하는데 사용된다. 블록체인 관리자는 사용자에게 정보이용 동의 체인코드를 사용하여 지원자의 정보를 확인할 수 있는 권한을 획득하고 블록체인 내부의 사용자 데이터를 검색한다. 검증 과정은 블록체인에 포함된 정보가 기업으로부터 받은 지원자의 정보와 일치하는지 여부를 확인하여 검증을 실시하게 된다. 이후 검증 결과를 회사에 제공하여 지원자의 경력 위조 여부를 알려주도록 한다. 이는 그림 9의 체인코드와 같이 구현한다.

```

1 //Submit resume and confirm receipt
2 confirm := EnterPrise.resumeResume(resume)
3 //Request verification from administrator
4 userInfo := EnterPrise.getUserVerification(user)
5 //Requeset user data query approval
6 userApprove := Admin.getUserApprove(userInfo)
7 if (userApprove == ture) {
8 //Chech data in blockchain if user approved
9     userData := Blockchain.getData(userInfo)
10 } else {
11 //Returns null if note authorized by user
12     userData := null }
13 //Return results
14 return userData
    
```

그림 9. 경력정보 검증 체인코드
Fig. 9. Verify function chain code

4. 성능 평가

본 장에서는 제안하는 기술의 성능 평가를 위해 QueryString방식과 txType의 검색 성능 및 블록생성 시간을 측정하여 실제 시스템으로 활용 가능한지에 대한 여부를 평가하였다.

4.1 성능 평가 환경

성능 평가 시 테스트베드로 사용한 환경과 블록체인 구현 요소들의 내용은 다음의 표 4, 5와 같다.

표 4. 체인코드의 비교 및 구성
Table 4. Composition and Contents of Chain code

Category	Content
CPU	Inter® Core™ i7-7500 CPU@2.7GHz
RAM	8GB
HDD	SSD 120GB

표 5. 체인코드의 비교 및 구성
Table 5. Composition and Contents of Chain code

Category	Content
OS	Ubuntu(18.04 LTS)
Language	GO Language(1.10.4) Java Script (ES6) Node.js (8.15.1)
Platform	Hyper-ledger Fabric (1.4) Docker (18.09.5) Fabric SDK (1.4)

제안된 시스템은 Ubuntu 18.04 LTS 운영체제에 하이퍼레저 패브릭 버전 1.4를 설치하여 구현하였다. 패브릭 SDK를 활용하여 블록체인 네트워크와 DAPP을 연동하여 실제적인 동작을 할 수 있도록 하였다. 체인코드는 node.js와 Go Language를 사용하여 구현하여 상호 연동하도록 구성하였다.

4.2 블록 생성 시간 성능 평가

블록 당 트랜잭션 수를 변경해 가면서 블록이 생성되는 시간을 측정하고, 제안하는 기법이 실제 환경에서 사용할 수 있는지 여부를 확인하였다. 테스트 횟수는

총 50회의 블록 생성을 통해 진행하였으며, 한 번에 처리하는 트랜잭션의 개수를 15, 20, 25, 30개의 조건으로 실험을 진행 하였다. 아래의 표 6에는 각 횟수에서 발생한 시간들의 평균치를 계산하여 표기하였다.

표 6. 블록생성 시간 성능평가 결과
Table 6. Result of block creating time

Processing Time(ms)			
15 Trans	20 Trans	25 Trans	30 Trans
1820	2507	2954	3367
1819	2512	2960	3377
1819	2515	2964	3374
1823	2510	2963	3380
1828	2516	2964	3379
1832	2519	2974	3383
1827	2515	2973	3378
1833	2516	2968	3377
1837	2522	2974	3389
1837	2523	2971	3382
2735	2639	3083	3447
2734	2636	3087	3453
2836	2642	3092	3448
2732	2637	3088	3448
2734	2640	3102	3451
	2636	3102	3454
	2636	3101	3456
	2649	3098	3462
	2648	3099	3459
	2649	3102	3454
		3802	3593
		3801	3604
		3790	3608
		3805	3607
		3796	3604
			3606
			3606
			3601
			3606
			3611

위의 표에서 볼 수 있듯이 트랜잭션의 처리는 10개씩 묶음으로 블록을 생성하여 처리되는 것을 확인할 수 있었다. 이를 통해 트랜잭션 하나의 평균 처리 시간을 계산하면 약 110ms의 처리시간을 가진다.

또한, 블록 생성 시 대기시간이 존재하는 것을 확인

하였다. 이는 위의 표에서 트랜잭션 처리 개수가 15개, 25개일 때의 처리시간과 20개, 30개일 때의 처리 시간의 비교를 통해 확인 가능하다. 트랜잭션이 20개, 30개일 때가 트랜잭션이 15개, 25개일 때의 처리시간보다 트랜잭션의 개수가 많음에도 더 빠른 처리시간을 가지는 것을 확인하였다. 이를 통해 블록이 생성 될 때 10개의 트랜잭션이 존재하면 곧바로 블록 생성을 시작하지만, 10개의 트랜잭션이 존재하지 않는 경우에는 일정 시간의 대기시간이 존재하는 것을 확인할 수 있었다.

4.3 데이터 검색 성능 평가

제안하는 기법에서 블록 내부의 데이터를 검색하는 방식은 QueryString 방식과 txType 방식이 존재한다. 데이터 검색 속도를 측정하기 위해 블록체인 내부에 보관하고 있는 데이터의 양을 5천개에서 1만개까지 변경하면서 블록 내의 데이터양에 따른 검색시간을 측정하였다. 그림 10은 데이터 검색 성능을 평가하여 그래프로 나타낸 수치이다.

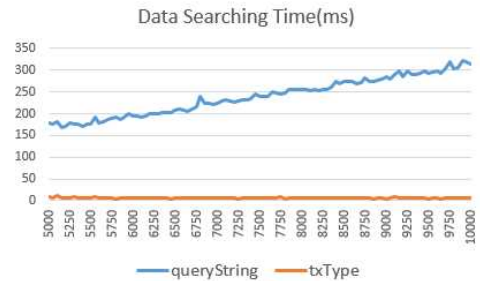


그림 9. 데이터 검색 시간 측정 결과

Fig. 9. Result of data searching time

위의 그래프에서 볼 수 있듯 QueryString 방식의 경우 사용자 ID 조건 하나를 사용하여 모든 데이터를 검색하기 때문에, txType 방식에 비해 많은 시간이 소요되는 것을 확인하였다. 또한 데이터양이 증가함에 따라 검색시간 또한 증가하는 것을 확인할 수 있었다. 이와 대조적으로 txType의 경우에는 사용자 ID를 포함한 추가 정보로 검색을 수행하기 때문에, 데이터양이 검색 시간에 크게 영향을 주지 않는 것을 확인할 수 있었다.

5. 결론

본 논문에서는 블록체인을 활용한 개인 경력 관리 및 유통 시스템을 제안하였다. 개인 경력의 확인을 할 수 없는 기존 채용 시스템의 한계점을 극복할 수 있도록 설계하였으며, 이를 위해 블록체인에서 기능을 담당하는 블록생성, QueryString 검색, txType 검색, 경력 검증 기능을 수행하는 코드를 작성하여 구현하였다. 제안 기법의 성능평가를 위한 환경으로 Inter core I7-7500, 8G RAM의 성능을 가진 PC에 Ubuntu 18.04 LTS 운영체제와 Hyper-Ledger Fabric 블록체인을 활용하여 구현하였다. 성능평가 항목으로 트랜잭션의 처리 시간과 블록체인 내부의 검색 시간을 평가하였으며, 평가 결과 트랜잭션 처리 시간은 약 110ms, 검색시간은 10ms이하의 결과를 나타내었다. 이를 통해 알 수 있듯이 실제 시스템에 적용하여 사용하는 경우에도 정상적인 서비스를 제공할 수 있음을 확인하였다. 또한, 서비스를 제공하는 서버의 사양에 따라 더 빠른 처리시간과 검색시간을 가질 수 있음을 확인했다. 다만, 현재 하이퍼레저 플랫폼에서 데이터의 양이 많아짐에 따라 블록의 크기가 커지는 현상이 있어, 향후에는 이를 보완하는 방법에 대한 연구를 진행할 것이다.

REFERENCES

[1] Recruiteze, "What The Expert Say About The Fourth Industrial Revolution", January. 2019.

[2] J.Kim, H.Jeong, Y.Kim, Y.Cho, "A Study on Application of NCS Recruiting Systems in Public Organization - Based on NCS Recruiting Performance During the First Half Year in 2015", Journal of Skills and Qualifications, 4(1), 65-84. March, 2015

[3] S.Lee, G.Ryu, Y.Shin, M.Kang, J.Oh, A.Lee, "A Survey on Unbiased Employment and Blind Employment and Performance Analysis", Human Ressources Development Service of Korea, November. 2018

[4] N.Yukio. "The Impact of the Block Chain", BookSta, 2017.

[5] J.Lee, "IT Science Story-Block Chain", Road Book, 2018.

[6] IBM, "An Introduction to Hyperledger", 2018

[7] L.Sankar, M.Sindhu, M.Sethumandhavan, "Survey of consensus protocols on blockchain applications", ICACCS, January. 2017

[8] M.Valenta, P.Sandner, "Comparison of Ethereum, Hpyerledger Fabric and Corda", FSBC Working Paper, June. 2017

저자약력

배수환 (Su-Hwan Bae)

[정회원]



- 2016.03 - 2018.02. 숭실대학교 일반대학원 융합소프트웨어학과 석사
- 2018.03 - 현재 숭실대학교 일반대학원 컴퓨터학과 박사과정

〈관심분야〉 블록체인, 정보보안, 컴퓨터 통신, 5G

신용태 (Yong-Tae Shin)

[정회원]



- 1991 - 1994 University of Iowa 컴퓨터학과 공학 박사
- 1995.03 - 현재 숭실대학교 컴퓨터학부 교수

〈관심분야〉 컴퓨터 네트워크, 분산 컴퓨팅, 인터넷 프로토콜, 초고속통신망, 전자상거래 기술