

# MEMBANDINGKAN PERFORMA ANTARA HYPERLEDGER DAN MYSQL

Riko Herwanto<sup>1</sup>, Onno W. Purbo<sup>2</sup>, and Sriyanto<sup>3</sup>

<sup>1,2,3</sup>Fakultas Ilmu Komputer, Institut Informatika dan Bisnis Darmajaya

Jl. 2.A. Pagar Alam No. 93, Bandar Lampung - Indonesia 35142

Telp. (0721) 787214 Fax. (0721) 700261

e-mail : rikoherwanto@darmajaya.ac.id, onno@indo.net.id, sriyanto@darmajaya.ac.id

## ABSTRACT

*In this paper, we report the benchmarking results of Hyperledger, a Distributed Ledger, which is the derivation Blockchain Technology. Method to evaluate Hyperledger in a limited infrastructure is developed. The measured infrastructure consists of 8 nodes with a load of up to 20000 transactions/second.. The benchmarking of Hyperledger shows better than a database system in a high workload scenario. We found that the maximum size data volume in one transaction on the Hyperledger network is around ten (10) times of MySQL. Also, the time spent on processing a single transaction in the blockchain network is 80-200 times faster than MySQL. This initial analysis can provide an overview for practitioners in making decisions about the adoption of blockchain technology in their IT systems.*

**Keywords**— Blockchain, Distributed Ledger Technology, Hyperledger, MySQL

## ABSTRAK

*Dalam tulisan ini, kami melaporkan hasil benchmark dari Hyperledger, sebuah Ledger Terdistribusi, yang merupakan turunan dari Teknologi Blockchain. Metode untuk mengevaluasi Hyperledger dalam infrastruktur terbatas dikembangkan. Infrastruktur yang diukur terdiri dari 8 node dengan beban hingga 20.000 transaksi / detik. Hyperledger secara konsisten menjalankan semua evaluasi, yaitu, untuk 20.000 transaksi. Penentuan tolok ukur Hyperledger menunjukkan lebih baik daripada sistem database dalam skenario beban kerja yang tinggi. Kami menemukan bahwa volume data ukuran maksimum dalam satu transaksi di jaringan Hyperledger adalah sekitar sepuluh (10) kali dari MySQL. Juga, waktu yang dihabiskan untuk memproses satu transaksi di jaringan blockchain adalah 80-200 kali lebih cepat daripada MySQL. Analisis awal ini dapat memberikan gambaran umum bagi para praktisi dalam membuat keputusan tentang adopsi teknologi blockchain dalam sistem TI mereka.*

**Kata Kunci**— Blockchain, Hyperledger, MySQL, Throughput

## I. PENDAHULUAN

Dalam karya ini, Hyperledger Fabric [1],

implementasi Distributed Ledger Technology (DLT) [2, 3] dari Linux Foundation, dijadikan patokan. DLT

mengelola Ledger melalui jaringan peer-to-peer menggunakan mekanisme konsensus dan kontrak pintar. Hyperledger adalah implementasi kerangka kerja Blockchain yang digunakan untuk mengembangkan aplikasi dengan arsitektur modular [4], dan merupakan proyek Blockchain open-source dan proyek terkait.

Dengan demikian, DLT memberikan model baru kepercayaan dan peluang bisnis. Untuk alasan ini, DLT adalah teknologi yang muncul di banyak bidang, seperti Teknologi Keuangan (Fintech) [5], layanan kesehatan [6], termasuk organisasi pemerintah [7]. Sayangnya, karena interaksi peer-to-peer yang kompleks, kinerja DLT lebih sulit diakses daripada sistem terpusat [8]. Karya ini melaporkan upaya untuk membandingkan DLT dan membandingkannya dengan sistem database terpusat.

Dalam membandingkan blockchain terdistribusi dengan basis data relasional, secara teoritis, basis data relasional lebih mudah untuk dikembangkan. Dengan demikian, ada lebih banyak pilihan dalam kerangka kerja basis data relasional daripada kerangka kerja Blockchain yang diizinkan [9]. Blockchains diizinkan dalam tahap pengembangan, sehingga kemungkinan ada opsi yang lebih tersedia

untuk Blockchains yang diizinkan di masa depan. Dukungan dalam solusi cloud, ada lebih banyak untuk database relasional.

Blockchain relatif baru dibandingkan dengan database terdistribusi. Karya ini menunjukkan bahwa teknologi Blockchain sebanding dengan teknik yang lebih tua dalam hal latensi [10]. Dalam beberapa kasus, kinerjanya lebih baik, dan ketika mempertimbangkan model konsistensi, kemungkinan akan ada beberapa kasus penggunaan yang berarti segera di mana blockchain akan menjadi pilihan yang lebih baik daripada database yang didistribusikan [11].

Blockchain adalah teknologi baru untuk berbagi data transaksional dan perhitungan tanpa menggunakan fasilitas pihak ketiga. Blockchain menggunakan arsitektur yang berbeda dibandingkan dengan database atau protokol tradisional. Perbedaan dalam arsitektur menyebabkan perbedaan dalam kinerja, biaya, dan keamanan, tetapi sedikit yang memprediksi kinerja sistem berbasis blockchain [12]. Dalam tulisan ini, kami telah mengevaluasi Hyperledger Fabric v1.0. Penilaian menunjukkan bahwa Hyperledger Fabric v1.0 dengan lebih dari dua node memiliki kinerja yang lebih baik di semua metrik evaluasi dibandingkan dengan hanya satu. Kami juga tertarik untuk membandingkan kinerja platform

blockchain dan blockchain publik dengan database tradisional (MySQL).

## II. METODE PENELITIAN

Arsitektur DLT yang kompleks dibagi menjadi empat lapisan, mis., Network, Node, Ledger, dan Application Layers untuk memudahkan analisis lebih lanjut. Pada setiap lapisan, beberapa metrik dan faktor pengaruh ditentukan. Metrik yang berbeda dan dipengaruhi oleh berbagai faktor diukur menjadi tolok ukur.

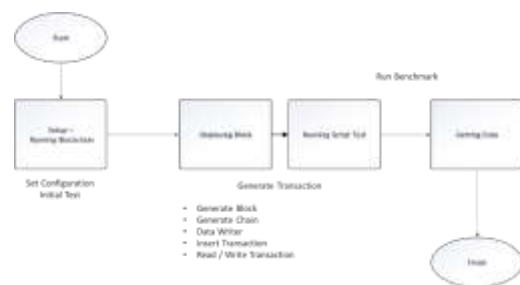
Konsep beban kerja DLT primer dan simulasi diperkenalkan. Berdasarkan analisis ini, sebuah kerangka kerja dirancang untuk membangun fondasi terdistribusi dari lingkungan pengujian dan melakukan pengukuran yang dapat direproduksi. Kerangka kerja ini dirancang agar teknologi dievaluasi dan lingkungan pengujian mudah dipertukarkan. Akibatnya, kerangka kerja desain diimplementasikan dengan alat benchmarking. Misalnya, pengukuran kinerja dan evaluasi, Hyperledger Fabric, percobaan dilakukan di lingkungan laboratorium yang terkontrol.

Evaluasi hasil pengukuran memberikan informasi tentang efek kinerja dari empat faktor, perubahan secara eksplisit dalam tingkat transaksi, beban kerja, ukuran blok dan juga dampak dari kehilangan paket.

Pengukuran menunjukkan bahwa faktor-faktor dari setiap lapisan dapat secara langsung mempengaruhi kinerja seluruh jaringan, meningkatkan tingkat transaksi, menuntut beban kerja, konfigurasi memori yang tidak menguntungkan, atau kondisi jaringan yang tidak menguntungkan.

Dalam karya ini, platform blockchain Hyperledger, Hyperledger Fabric v1.0 dievaluasi. Percobaan dilakukan pada Laptop i7, 8GB RAM, 500GB SSD, 3 Core 2 Duo CPU, 4GB RAM, 160GB HDD, dan menjalankan Ubuntu 18.04 LTE, ini digunakan untuk membandingkan karya Hyperledger dengan Relational Database, dalam hal ini, MySQL, dengan beban data yang sama

### 2. 1. Platform Eksperimen



Gambar 1. Desain Eksperimen

Dalam tulisan ini, kerangka evaluasi untuk Ledger yang didistribusikan pribadi dirancang dan dikembangkan. Untuk tujuan ini, berbagai lapisan telah ditentukan. Ini adalah lapisan jaringan, di atas lapisan simpul dan lapisan Ledger ke lapisan aplikasi. Untuk setiap lapisan ini, metrik, dan faktor telah diidentifikasi,

yang memungkinkan untuk mengukur / memengaruhi kinerja jaringan DLT. Juga, beban kerja telah ditentukan, yang menekankan aspek individual dari DLT atau mewakili kasus penggunaan yang realistis. Empat fase percobaan adalah:

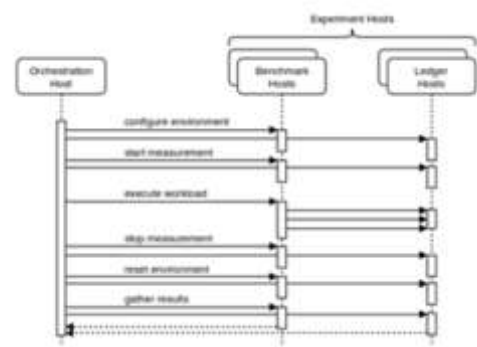
#### a. Fase Desain

Tujuannya adalah untuk menentukan kerangka kerja yang menjalankan eksperimen pada DLT. Ini akan mendukung konfigurasi teknologi dan pengukuran aktual serta evaluasi hasil. Fase desain adalah untuk menentukan berbagai tujuan, yaitu, Throughput dan Latency, dibahas, dibagi menjadi tiga fase, aplikasi, pengukuran, dan evaluasi, yang akan dibahas di bawah ini.

#### b. Fase Tesbed

Testbed disiapkan sebelum proses benchmarking. Ini mencakup instalasi dan konfigurasi perangkat lunak, mis., OS, mesin Docker, Jaringan, yang diperlukan untuk memfasilitasi langkah-langkah berikut. Setelah itu, Ledger Hosts, dilengkapi dengan bahasa GO dan Docker CE, diperlukan untuk melakukan benchmark DLT. Testbed mencakup banyak alat, seperti, Chaincode-Payload-Size, Chaincode-Skalabilitas, Skalabilitas Saluran, yang digunakan selama fase pengukuran untuk merekam host eksperimental.

#### c. Fase Pengukuran



Gambar 2. Desain Alur Kerja

Gambar 2. menjelaskan fase pengukuran, dimulai dengan langkah konfigurasi awal. Dalam menjalankan percobaan, setiap perubahan konfigurasi terakhir diterapkan ke Host percobaan, seperti gangguan jaringan. Setelah ini, inisialisasi semua monitor di host eksperimen tempat eksperimen berlangsung. Ini mungkin termasuk merekam lalu lintas jaringan atau menggunakan sumber daya, misalnya, Eksekusi beban kerja diprakarsai oleh Host Orkestrasi, yang tidak akan mengganggu proses dan menunggu sampai selesai. Host patokan mengambil alih dan mengoperasikan eksperimen berdasarkan definisi beban kerja pada host Ledger. Semua langkah konfigurasi dan inisialisasi sebelumnya memungkinkan sistem untuk menjalankan eksperimen tanpa intervensi eksternal. Semua perubahan selama percobaan diatur waktunya atau secara langsung diinduksi oleh eksekusi beban

kerja pada host Ledger. Setelah eksekusi beban kerja selesai, percobaan dimatikan. Monitor dihentikan pada host Ledger, dan gangguan apa pun yang terjadi selama langkah konfigurasi pengukuran dibatalkan. Misalnya, menghapus semua koneksi jaringan buatan yang hilang yang telah ditempatkan pada jaringan, untuk memungkinkan pekerjaan yang terus-menerus tidak terganggu pada testbed.

Akhirnya, setiap informasi yang dikumpulkan diambil dari host Ledger dan dikumpulkan di Host Orkestrasi. Itu memungkinkan untuk segera mengatur ulang host Ledger kecuali untuk Host Orkestrasi, untuk pengukuran lebih lanjut

#### d. Fase Evaluasi

Fase terakhir adalah fase evaluasi, di mana data yang dikumpulkan harus dievaluasi, dimulai dengan fase preprocessing. Hasil yang datang dari beberapa host harus melalui langkah-langkah berikut:

- a). Menyederhanakan pemrosesan lebih lanjut, seperti mengonversi ke format file standar.
- b). Membersihkan duplikasi lalu lintas yang direkam pada banyak host.
- c). Normalisasi.
- d) Integrasi dan hubungkan berbagai sumber data

Data yang diproses kemudian dapat dievaluasi berdasarkan metrik yang relevan. Beberapa metode evaluasi

didefinisikan dalam karya ini, dan format data yang diproses sebelumnya memungkinkan untuk menambahkan metode evaluasi lebih lanjut sehingga diperoleh dengan mudah:

- **Transaksi & Baca Latensi:** Mengukur waktu untuk transaksi yang dikeluarkan untuk diselesaikan dan tanggapan yang tersedia untuk aplikasi yang mengeluarkan transaksi. Maksimum, minimum, dan latensi untuk siklus tes disediakan.

- **Transaction & Read Throughput:** Mengukur laju aliran semua transaksi melalui sistem, dalam transaksi per detik, selama siklus.

Dalam percobaan, transaksi pertama kali dieksekusi untuk pra-mengisi rantai / Ledger dengan Blok. Ini adalah titik awal untuk memperkirakan bagaimana sistem berperilaku ketika cukup data di dalam system.

Kemudian, transaksi baca-tulis dijalankan di mana setiap transaksi membaca dan memodifikasi Blok secara acak. Dalam setiap percobaan, satu (atau dua) parameter bervariasi (ditandai dalam garis putus-putus), sambil menjaga parameter lainnya tetap. Cache sistem file OS tidak dihapus antara transaksi Insert dan transaksi baca-tulis dengan asumsi bahwa, dalam pengaturan praktis, sebagian besar data langsung akan berasal dari cache

sistem file. Percobaan menggunakan parameter berikut:

- 1). Total jumlah rantai (default - 10)
- 2). Jumlah transaksi yang disimulasikan secara paralel pada setiap rantai (default - 10)
- 3). Total Blok di seluruh rantai (kunci didistribusikan secara merata di seluruh rantai) (default - 20.000)
- 4). Jumlah kunci yang dibaca dan dimodifikasi secara acak oleh setiap transaksi (default - 4)
- 5). Nilai ukuran untuk setiap Blok (default - 200 byte)
- 6). Jumlah transaksi di setiap blok (default - 50)

Script Chaincode dikembangkan untuk mengatur eksperimen default. Script ditulis menggunakan JavaScript Object Notation (JSON), yang merupakan format ringkas untuk bertukar data komputer. Formatnya berbasis teks, dapat dibaca manusia, dan digunakan untuk merepresentasikan struktur data sederhana dan array asosiatif (disebut objek). JSON digunakan untuk mengirim data terstruktur melalui koneksi jaringan. Dalam karya ini, skrip untuk mengatur konfigurasi adalah sebagai berikut:

```

type
type configuration struct {
    chainMgrConf *chaingmt.ChainMgrConf
    batchConf    *chaingmt.BatchConf
    dataConf     *dataConf
    txConf       *txConf
}

func defaultConf() *configuration {

```

```

    conf := &configuration{}
    conf.chainMgrConf =
    &chaingmt.ChainMgrConf{DataDir:
    "/tmp/fabric/ledgerPerfTests", NumChains: 1}
    conf.batchConf =
    &chaingmt.BatchConf{BatchSize: 10,
    SignBlock: false}
    conf.txConf = &txConf{numTotalTx: 20000,
    numParallelTxPerChain: 2, numWritesPerTx: 4,
    numReadsPerTx: 4}
    conf.dataConf = &dataConf{numKVs: 20000,
    kvSize: 200}
    return conf
}

```

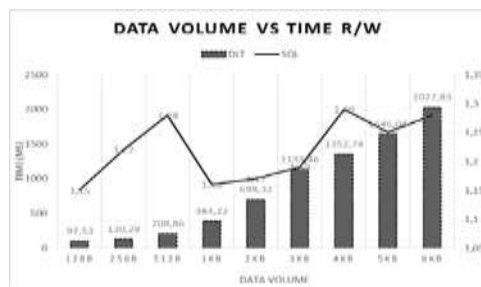
### III. HASIL DAN PEMBAHASAN

Pada bagian ini, Distributed Ledger dan Database Relasional, seperti MySQL, dengan dua cara, membaca dan menulis data akan dibandingkan. Perbandingan ini akan memberikan beberapa wawasan praktis untuk menggabungkan teknologi ledger terdistribusi dan database terpusat.

Dalam karya ini, Hyperledger Fabric untuk blockchain dan database MySQL untuk database relasional digunakan. Adapun pengaturan seperti itu, kami memiliki dua standar. Yang pertama adalah apakah objek yang diuji secara fungsional lengkap atau tidak. Kedua, Fabric Hyperledger untuk mengimplementasikan berbagai fungsi dan eksekusi [16]. Juga, MySQL dapat digunakan untuk menyimpan berbagai jenis data.

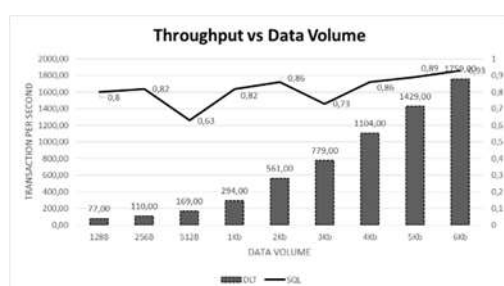
Gambar 3 menunjukkan waktu pelaksanaan setiap transaksi, Blockchain menghabiskan lebih banyak waktu pada volume data yang lebih besar, mencapai 2027 ms dengan data 6KB, waktu

membaca dan menulis rata-rata adalah 1,22 ms. Namun, kami menemukan bahwa waktu yang digunakan untuk membaca dan menulis data yang dihabiskan oleh Blockchain adalah 1660 kali lebih tinggi dari MySQL.



Gambar 3. Grafik Perbandingan Data Volume vs Waktu R/W.

Gambar 3. menunjukkan hubungan antara throughput dan volume data per transaksi antara Blockchain dan database MySQL. Ketika volume data meningkat dan melampaui ambang 2KB, throughput Hyperledger meningkat



Gambar 4. Grafik perbandingan Throughput vs Data Volume

Membandingkan hasilnya, Hyperledger menunjukkan kinerja yang lebih tinggi dengan volume data yang jauh lebih besar dan hubungan linier

dengan data. Ditemukan bahwa Hyperledger adalah 80-200 lebih cepat dari MySQL. Dengan peningkatan volume data dalam satu transaksi, waktu pemrosesan yang dibutuhkan meningkat. Waktu dan volume data yang digunakan menghasilkan hubungan eksponensial. Perhatikan bahwa ada delapan node dalam blockchain, hubungan linear mungkin merupakan hasil dari jumlah node

#### IV. SIMPULAN

Berikut Pendekatan eksperimental terstruktur untuk mengkarakterisasi kinerja platform Blockchain Hyperledger. Kerangka kerja yang dikembangkan dalam makalah ini dibangun untuk diperluas. Ini termasuk menambahkan faktor, metrik, beban kerja, dan lebih banyak faktor yang dapat diperluas dengan berbagai keterbatasan jaringan seperti kecepatan jaringan terbatas atau keterlambatan jaringan. Metrik harus memasukkan informasi tentang beban setiap node, untuk memungkinkan membuat pernyataan yang tepat tentang kemacetan Ledger yang didistribusikan. Dalam karya ini, fitur ini telah diterapkan sebagian. Dengan demikian, jumlah beban kerja dapat dengan mudah ditingkatkan.

Kinerja, yaitu, throughput, waktu eksekusi, dan latensi, menunjukkan bahwa

Hyperledger secara konsisten lebih baik daripada SQL. Kami menemukan bahwa volume data maksimum dalam satu transaksi di jaringan Hyperledger adalah sekitar sepuluh (10) kali dari MySQL. Juga, waktu yang dihabiskan untuk memproses satu transaksi pada jaringan blockchain adalah 80-200 kali lebih cepat daripada MySQL. MySQL berkinerja konstan daripada delapan-node blockchain. Throughput Hyperledger didapati bekerja lebih baik dibandingkan MySQL

Secara umum, Hyperledger menghasilkan kinerja yang sama untuk sejumlah node, terlepas dari bebannya. Namun, kinerja Hyperledger dipengaruhi karena jumlah node berubah, dan, dengan demikian, jumlah Blok, ukuran Blok, dan Jumlah Transaksi. Untuk mencapai throughput yang lebih tinggi, efisiensi yang lebih besar, interval blok harus dibuat sesedikit mungkin. Kami telah menemukan bahwa interval blok untuk protokol Blockchain berbasis Hyperledger tidak boleh kurang dari 12 detik. Ini akan memastikan penyebaran lebih cepat dan latensi rendah.

Didapatkan juga bahwa:

- Blockchain ekuivalen dengan database.
- Blockchain tidak bisa di ubah jadi lebih aman daripada database.

- Blockchain untuk pola transaksi yang fix, bisa lebih cepat daripada database

Singkatnya, penggunaan terbaik untuk database meliputi yang berikut ini.

- Aplikasi atau sistem yang memanfaatkan aliran data yang berkelanjutan.
- Menyimpan informasi rahasia
- Pemrosesan transaksi online yang harus cepat
- Aplikasi atau sistem di mana verifikasi data tidak diperlukan.
- Data relasional
- Aplikasi mandiri

Singkatnya, penggunaan terbaik untuk blockchain meliputi yang berikut:

- Nilai transfer
- Nilai penyimpanan
- Transaksi moneter
- Verifikasi data tepercaya
- Sistem pemilihan
- Aplikasi terdesentralisasi

Hasil ini menyiratkan bahwa blockchain mungkin lebih cocok untuk aplikasi / sistem intensif data.

## DAFTAR PUSTAKA

- [1] Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, De Caro A,



- Enyeart D, Ferris C, Laventman G, Manevich Y, Muralidharan S. *Hyperledger fabric: a distributed operating system for permissioned blockchains*. Proceedings of the Thirteenth EuroSys Conference. Porto, Portugal. 2018; p. 30.
- [2] H. Sukhwani, N. Wang, K. S. Trivedi and A. Rindos. *Performance Modeling of Hyperledger Fabric (Permissioned Blockchain Network)*. 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA). Cambridge, MA. 2018: pp. 1-8.
- [3] Walport M. *Distributed ledger technology: Beyond blockchain*. UK Government Office for Science. Tech. Rep; 2016.
- [4] Davidson S, De Filippi P, Potts J. *Disrupting governance: The new institutional economics of distributed ledger technology*. 2016.
- [5] Chishti S, Barberis J. *The FinTech book: the financial technology handbook for investors, entrepreneurs and visionaries*. New York. John Wiley & Sons; 2016.
- [6] Cunningham J, Ainsworth J. *Enabling patient control of personal electronic health records through distributed ledger technology*. Stud Health Technol Inform. 2018; 245: 45-48.
- [7] Genkin D, Papadopoulos D, Papamanthou C. *Privacy in decentralized cryptocurrencies*. Communications of the ACM. 2018; 61(6):78-88.
- [8] Kocsis I, Pataricza A, Telek M, Klenik A, Deé F, Cseh D. *Towards Performance Modeling of Hyperledger Fabric*. In International IBM Cloud Academy Conference (ICACON). 2017.
- [9] D. Chays and Yuetang Deng, *Demonstration of AGENDA tool set for testing relational database applications*. Proceedings of 25th International Conference on Software Engineering, 2003. Portland, OR, USA. 2003: pp. 802-803.
- [10] R. Yasaweerasinghelage, M. Staples and I. Weber. *Predicting Latency of Blockchain-Based Systems Using Architectural Modelling and Simulation*. 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg. 2017: pp. 253-256.
- [11] Rauchs M, Glidden A, Gordon B, Pieters GC, Recanatini M, Rostand F, Vagneur K, Zhang BZ. *Distributed ledger technology systems: a conceptual framework*. Cambridge, UK. 2018.

- [12] Qassim Nasir, Ilham A. Qasse, Manar Abu Talib, and Ali Bou Nassif. *Performance Analysis of Hyperledger Fabric Platforms*. Security and Communication Networks, vol. 2018, Article ID 3976093, 14 pages, 2018
- [13] Gaur N, Desrosiers L, Ramakrishna V, Novotny P, Baset SA, O'Dowd A. *Hands-On Blockchain with Hyperledger: Building decentralized applications with Hyperledger Fabric and Composer*. Packt Publishing Ltd; 2018.
- [14] A. Baliga, N. Solanki, S. Verekar, A. Pednekar, P. Kamat and S. Chatterjee, Performance Characterization of Hyperledger Fabric. *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, Zug. 2018; pp. 65-74.
- [15] White B, Lepreau J, Stoller L, Ricci R, Guruprasad S, Newbold M, Hibler M, Barb C, Joglekar A. *An integrated experimental environment for distributed systems and networks*. Proceedings of the 5th symposium on Operating systems design and implementation. Boston, Massachusetts. 2002;36(SI):255-270.
- [16] Buchman E. *Tendermint: byzantine fault tolerance in the age of blockchains*. MA thesis. Department Engineering Systems and Computing. University of Guelph. Ontario, Canada. 2016.
- [17] Dhillon V, Metcalf D, Hooper M. *The hyperledger project*. In *Blockchain enabled applications*. Apress, Berkeley, CA. 2017: 139-149
- [18] Nakamoto S. *Bitcoin: A peer-to-peer electronic cash system*. 2008
- [19] Nasir Q, Qasse IA, Abu Talib M, Nassif AB. *Performance analysis of hyperledger fabric platforms*. Security and Communication Networks. Vol. 2018; 14.
- [20] Cachin C. *Architecture of the hyperledger blockchain fabric*. In Workshop on distributed cryptocurrencies and consensus ledgers. 2016;. 310: 4-11.
- [21] C. Decker and R. Wattenhofer. *Information propagation in the Bitcoin network*. *IEEE P2P 2013 Proceedings*, Trento, 2013, pp. 1-10.
- [22] Cunningham J, Ainsworth J. *Enabling patient control of personal electronic health records through distributed ledger technology*. *Stud Health Technol Inform*. 2018; 245:45-8.
- [23] Genkin D, Papadopoulos D, Papamanthou C. *Privacy in decentralized cryptocurrencies*. *Communications of the ACM*. 2018; 61(6):78-88.

- [24] Bolze R, Cappello F, Caron E, Daydé M, Desprez F, Jeannot E, Jégou Y, Lanteri S, Leduc J, Melab N, Mornet G. *Grid'5000: A large scale and highly reconfigurable experimental grid testbed*. The International Journal of High Performance Computing Applications. 2006; (4):481-494.
- [25] Croman K, Decker C, Eyal I, Gencer AE, Juels A, Kosba A, Miller A, Saxena P, Shi E, Siler EG, Song D. *On scaling decentralized blockchains*. In International Conference on Financial Cryptography and Data Security 2016). Springer, Berlin, Heidelberg. 2016; 106-125.
- [26] Maull R, Godsiff P, Mulligan C, Brown A, Kewell B. *Distributed ledger technology: Applications and implications*. Strategic Change. 2017; 26(5): 481-489.
- [27] N. Papadis, S. Borst, A. Walid, M. Grissa and L. Tassiulas. *Stochastic Models and Wide-Area Network Measurements for Blockchain Design and Analysis*. IEEE INFOCOM 2018 - IEEE Conference on Computer Communications. Honolulu, HI. 2018; pp. 2546-2554.
- [28] Sajana P, Sindhu M, Sethumadhavan M. *On Blockchain Application: Hyperledger Fabric and Ethereum*. International Journal of Pure and Applied Mathematics. 2018;118(18): 2965-2970.
- [29] S. Pongnumkul, C. Siripanpornchana and S. Thajchayapong. *Performance Analysis of Private Blockchain Platforms in Varying Workloads*. 2017 26th International Conference on Computer Communication and Networks (ICCCN). Vancouver, BC. 2017; pp. 1-6..
- [30] P. Thakkar, S. Nathan and B. Viswanathan, *Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform*, IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS). Milwaukee, WI. 2018; pp. 264-276..
- [31] Dinh TT, Wang J, Chen G, Liu R, Ooi BC, Tan KL. *Blockbench: A framework for analyzing private blockchains*. Proceedings of the 2017 ACM International Conference on Management of Data. 2017; pp. 1085-1100
- [32] Valentin Hauner. *Trustworthy Configuration Management with Distributed Ledgers*. MA thesis. Department of Informatics, Technical University of Munich; 2018.

- [33] Vukolić M. *Hyperledger fabric: towards scalable blockchain for business*. Trust in Digital Life. 2016.
- [34] M. Shbair, M. Steichen, J. François and R. State, *Blockchain orchestration and experimentation framework: A case study of KYC. NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. Taipei. 2018; pp. 1-6.