JURNAL ILMIAH TEKNIK MESIN, ELEKTRO DAN KOMPUTER
Halaman Jurnal: http://ejurnal.stie-trianandra.ac.id/index.php/juritek
Halaman UTAMA Jurnal : http://ejurnal.stie-trianandra.ac.id/index.php

PENERAPAN BIG DATA UNTUK MENGATUR SISTEM ANALISIS DATA

Indra Ava Dianta[a], Ahmad Ashifuddin Aqham[b], Dwi Setiawan[c]
[a] Fakultas Ilmu Terapan / Teknik Komputer, indra@stekom.ac.id, Universitas STEKOM
b Fakultas Ilmu Terapan / Komputerisasi Akuntansi, ashif@stekom.ac.id, Universitas STEKOM
[c] Fakultas Ilmu Terapan / Teknik Komputer, dwisetiawan@stekom.ac.id, Universitas STEKOM

ABSTRAK

Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate to handle them. The challenges include analysis, retrieval, search, sharing, storage, transfer, visualization, query, and updating and privacy of information. However, this huge amount of data cannot be easily handled because most CS systems are relational, and adjustments are required prior to any processing. With the advent of Big Data, a new NoSQL system came to solve this relational data problem. So, we propose an approach to migrate historical CS data from relational to NoSQL systems, and use a distributed environment containing multiple nodes. As an experiment, we migrated the data generated by oil and gas CS to a suitable distributed NoSQL system, and we performed several data mining experiments on it to compare the results and prove the performance obtained.

Keyword : *Big data, Data mining, control systems, NoSQL, NewSQL*

Abstrak

Big data merupakan istilah yang digunakan untuk gudang data yang sangat besar atau kompleks sehingga aplikasi pemrosesan data tradisional tidak memadai untuk mengatasinya. Tantangannya meliputi analisis, pengambilan, pencarian, berbagi, penyimpanan, transfer, visualisasi, kueri, dan pembaruan serta privasi informasi. Namun, data yang sangat besar ini tidak dapat dengan mudah ditangani karena sebagian besar sistem CS bersifat relasional, dan diperlukan penyesuaian sebelum pemrosesan apa pun. Dengan munculnya Big Data, sistem NoSQL baru datang untuk mengatasi masalah data relasional ini. Jadi, kami mengusulkan pendekatan untuk memigrasi data CS historis dari relasional ke sistem NoSQL, dan menggunakan lingkungan terdistribusi yang berisi banyak node. Sebagai eksperimen, kami memigrasikan data yang dihasilkan oleh CS minyak dan gas ke sistem NoSQL terdistribusi yang sesuai, dan kami melakukan beberapa eksperimen data mining padanya untuk membandingkan hasil dan membuktikan kinerja yang diperoleh.

Kata Kunci: Big data, Data mining, Sistem kontrol, NoSQL, NewSQL, Analytics

1. PENDAHULUAN

Data besar adalah istilah untuk kumpulan data yang begitu besar atau kompleks sehingga aplikasi pemrosesan data tradisional tidak memadai untuk mengatasinya. Tantangannya meliputi analisis, pengambilan, kurasi data, pencarian, berbagi, penyimpanan, transfer, visualisasi, kueri, dan pembaruan serta privasi informasi. Istilah "data besar" sering merujuk hanya pada penggunaan analitik prediktif, analitik perilaku pengguna, atau metode analitik data lanjutan tertentu lainnya yang mengekstrak nilai dari data, dan jarang ke ukuran kumpulan data tertentu. “Ada sedikit keraguan bahwa jumlah data yang sekarang tersedia memang besar, tapi itu bukan karakteristik yang paling relevan dari ekosistem data baru ini.”

Big Data dapat didefinisikan sebagai fenomena yang muncul yang mengacu pada praktik penanganan volume data yang besar dan kompleks, dengan sistem teknis terkait seperti algoritme yang digunakan untuk memvisualisasikan dan menganalisis secara real-time atau tidak (real-time atau batch) data masif ini , untuk menciptakan nilai tambah bagi organisasi. Secara paralel, Data Mining juga mengalami perkembangan pesat (jaringan saraf, algoritme genetik ...) dan sekarang dimungkinkan untuk membuat struktur TI yang belajar sendiri. Ini adalah Pembelajaran Mesin: analisis dan implementasi metode otomatis yang memungkinkan mesin (pada umumnya) untuk berkembang melalui proses pembelajaran, dan dengan demikian melakukan tugas-tugas yang sulit atau tidak mungkin diisi dengan cara algoritmik yang lebih konvensional.

Model data ON-RELATIONAL dan relasional berbeda. Model relasional mengambil data dan memisahkannya menjadi banyak tabel yang saling terkait yang berisi baris dan kolom. Tabel mereferensikan satu sama lain melalui kunci asing yang disimpan dalam kolom juga. Saat menanyakan data, informasi yang diminta akan dikumpulkan dari banyak tabel, seolah-olah pengguna bertanya: apa jawaban atas pertanyaan saya?

Model data non-relasional sering dimulai dari kueri khusus aplikasi, sebagai lawan dari model data relasional. Model data non-relasional kemudian akan digerakkan oleh pola akses khusus aplikasi. Pemahaman lanjutan tentang struktur data dan algoritme diperlukan [1], sehingga desain utamanya adalah mengetahui: pertanyaan apa yang cocok dengan data saya?

Dalam makalah ini kami akan memberikan pendahuluan, pratinjau dari berbagai model data non-relasional. Selanjutnya, kami akan fokus pada database dokumen di Sekte. 2 dan 3, dengan membahas kriteria evaluasi terkait model data ini. Terakhir, kami akan membahas hasil evaluasi di bagian kesimpulan.

Perusahaan industri dan pabrikan semakin dilengkapi dengan sistem kontrol (CS) yang menghasilkan data waktu nyata dalam jumlah yang sangat besar. Data ini digunakan oleh aplikasi khusus untuk menyediakan informasi kritis waktu nyata, grafik perkembangan waktu nyata, alarm waktu nyata, dll. Nanti, data ini disimpan untuk arsip historis, dan dalam banyak kasus sering dihapus nanti. Namun, banyak pengguna yang semakin tertarik dengan data historis ini untuk menggunakannya dalam banyak proses bisnis, dan terutama di area data mining, seperti mengekstraksi pengetahuan yang berguna, memberikan sarana umpan balik awal, meningkatkan permintaan di masa depan, dll. Namun demikian, akumulasi besar ini data secara progresif membutuhkan kapasitas dan daya untuk mendukung pemrosesan dan penyimpanan. Setiap mesin atau server mencapai batasnya untuk mendukung penyimpanan dan pemrosesan data yang sangat besar, apa pun kapasitas fisiknya, CPU, memori, atau disk. Selain itu, peningkatan atau ekstensi mesin tidak dapat dianggap sebagai solusi permanen. Jadi, platform terdistribusi berisi banyak server (node), seperti cluster, grid dapat digunakan secara efisien untuk menangani masalah ini.

Artikel ini terdiri dari lima bagian. Bagian pertama adalah pengantar, dan yang kedua, kami menyajikan masalah data relasional pada lingkungan terdistribusi dan beberapa pekerjaan terkait. Di bagian ketiga, kami memberikan pendekatan untuk menggunakan sistem NoSQL terdistribusi dan migrasi data. Di bagian keempat, kami menyiapkan data CS untuk perusahaan [2] di platform eksperimental. Di bagian kelima, kami melakukan beberapa kueri menarik untuk menguji dan membandingkan kinerja. Sebuah kesimpulan menutup makalah ini di bagian terakhir.

2. TINJAUAN PUSTAKA

2.1. Volume

Volume menggambarkan jumlah data yang dihasilkan oleh perusahaan atau individu. Big Data biasanya dikaitkan dengan fitur ini. Perusahaan di semua sektor perlu menemukan cara untuk mengelola volume data yang terus meningkat yang dibuat setiap hari. Katalog lebih dari 10 juta produk telah menjadi aturan daripada pengecualian. Beberapa pelanggan yang tidak hanya mengelola produk tetapi juga pelanggan mereka sendiri dapat dengan mudah mengakumulasi volume yang melebihi terabyte data

Heading pada level kedua dituliskan dengan boldface italics dengan menggunakan huruf besar dan huruf kecil. Heading dituliskan rata kiri.

2.2. Kecepatan

Kecepatan menggambarkan frekuensi di mana data dihasilkan, ditangkap, dan dibagikan. Karena perkembangan teknologi terkini, konsumen dan bisnis menghasilkan lebih banyak data dalam kerangka waktu yang jauh lebih singkat. Pada tingkat kecepatan ini, perusahaan hanya dapat memanfaatkan data ini jika dikumpulkan dan dibagikan dalam waktu nyata. Justru pada tahap inilah banyak analisis, CRM, personalisasi, titik penjualan, dan sistem lain gagal. Mereka hanya dapat memproses data dalam batch setiap beberapa jam, paling baik.

2.3. Keragaman

Perkembangan jenis data dari sumber seperti media sosial, interaksi Mesin ke Mesin, dan perangkat seluler menciptakan keragaman yang besar di luar data transaksional tradisional. Data tidak lagi menjadi bagian dari struktur yang jelas dan mudah digunakan. Jenis data baru termasuk konten, data geo-spasial, titik data perangkat keras, data geolokasi, data koneksi, data yang dihasilkan mesin, data pengukuran, data seluler, titik data fisik, proses, data RFID, data penelitian, data kepercayaan, data aliran, data media sosial, data teks, dan data berbasis web.

Mengapa penting untuk memahami semua ini?

JURNAL ILMIAH TEKNIK MESIN, ELEKTRO DAN KOMPUTER
Halaman Jurnal: http://ejurnal.stie-trianandra.ac.id/index.php/juritek
Halaman UTAMA Jurnal : http://ejurnal.stie-trianandra.ac.id/index.php

Karena Big Data membantu kita mendapatkan representasi yang lebih baik dari interaksi pelanggan dengan perusahaan. Ini memungkinkan pemahaman yang lebih baik tentang apa yang ingin dicapai pelanggan di setiap titik kontak. Ini meminimalkan risiko kehilangan pelanggan ini saat beralih dari satu titik kontak ke titik lain dan memastikan relevansi informasi yang dikirimkan kepada mereka. Jadi, untuk meningkatkan kualitas layanan, aspek kunci bagi pelanggan, dan tingkat transformasi pelanggan ini, penting bagi perusahaan untuk tidak melupakan Big Data.

2.4. Karakteristik Database NoSQL

Istilah "NoSQL" ditemukan pada tahun 2009 selama acara di database terdistribusi. Istilah ini tidak jelas, tidak benar (beberapa mesin NoSQL menggunakan varian bahasa SQL, misalnya Cassandra), tetapi memiliki keuntungan karena memiliki efek pemasaran dan polemik tertentu. Pada bagian ini, kita akan membahas karakteristik umum mesin NoSQL, secara historis, konseptual dan teknis, dalam kaitannya dengan database relasional, tetapi juga secara independen dari referensi ini.

2.5. Prinsip Database NoSQL

Database NoSQL, terutama yang berorientasi pada dokumen, mengabaikan kelebihan dari database relasional yaitu pengertian registrasi dan relasi antar elemen, agar dapat fokus pada pengertian suatu dokumen. Database NoSQL jauh lebih fleksibel dan lebih skalabel. Struktur organisasi tidak lagi terkait dengan skema relasional yang sulit untuk dimodifikasi, dan oleh karena itu dasarnya dapat tumbuh tanpa kendala. Di sisi lain, orientasi "dokumen" memfasilitasi penerapan database di beberapa mesin. Secara otomatis, tentu saja. Pengembang tidak peduli dengan lokasi dokumen, terbelah atau tidak. Ketika database menjadi terlalu besar, itu cukup untuk mendefinisikan mesin baru yang terhubung di jaringan, dan database NoSQL bertahan.

Ini adalah jawaban untuk aplikasi baru yang menuntut kecepatan pemrosesan dan kuantitas data yang dikelola. Kuantitas urutan beberapa ratus terabyte. Perhatikan bahwa ada juga basis NOSQL tipe "kolom" dan basis tipe "grafik". Basis kolom adalah solusi yang sangat baik untuk analisis besar-besaran. Dasar grafik, lebih rumit untuk dipahami, seperti yang ditunjukkan oleh denominasi mereka, lebih disesuaikan dengan resolusi pertanyaan organisasi dalam jaringan (struktur dalam busur dan simpul).

Sistem NoSQL menggunakan replikasi untuk mencapai berbagai tujuan.

a. Ketersediaan. Replikasi memastikan bahwa sistem selalu tersedia. Jika terjadi kegagalan server, node, atau disk, tugas yang dilakukan oleh komponen yang rusak dapat segera didukung oleh komponen lain. Teknik failover ini merupakan aset penting untuk memastikan stabilitas sistem yang dapat mencakup ribuan node, tanpa harus menelan anggaran yang besar dalam pemantauan dan pemeliharaan.
b. Skalabilitas (membaca). Jika data tersedia di beberapa mesin, permintaan untuk mendistribusikan (baca) di mesin ini menjadi mungkin. Ini adalah skenario tipikal untuk skalabilitas aplikasi Web.
c. Skalabilitas (menulis). Akhirnya, orang dapat berpikir untuk mendistribusikan juga permintaan secara tertulis, tetapi ada orang yang dihadapkan pada masalah potensial yang rumit dari persaingan tulisan dan rekonsiliasi.

Teknik ini sangat klasik dan digunakan oleh semua DBMS di dunia. Alih-alih berulang kali menulis ke disk tanpa urutan yang ditentukan sebelumnya (yang disebut akses "acak"), yang setiap kali memerlukan perpindahan kepala baca dan karena itu latensi beberapa milidetik, seseorang menulis secara berurutan dalam file Log (log) dan data juga ditempatkan di memori RAM.

2.6. Munculnya Big Data dan Database NoSQL

Evolusi perangkat lunak mengikuti evolusi material secara alami. DBMS pertama dibangun di sekitar mainframe dan bergantung pada kapasitas penyimpanan saat itu. Keberhasilan model relasional tidak hanya karena kualitas model itu sendiri tetapi juga untuk optimalisasi penyimpanan yang memungkinkan pengurangan redundansi data. Dengan meluasnya penggunaan interkoneksi jaringan, meningkatkan bandwidth Internet dan menurunkan biaya mesin yang cukup bertenaga, kemungkinan baru telah muncul di bidang komputasi terdistribusi dan virtualisasi, misalnya.

Pergeseran ke abad kedua puluh satu telah menyaksikan volume data yang dimanipulasi oleh beberapa organisasi, terutama yang terkait dengan Internet, meningkat secara dramatis. Data ilmiah, jaringan sosial, operator telepon, database medis, badan pertahanan teritorial nasional, indikator ekonomi dan sosial, dll., Peningkatan komputerisasi dari semua jenis pemrosesan menyiratkan peningkatan eksponensial dalam volume data Sekarang dalam petabyte. Inilah yang disebut oleh Anglo-Saxon sebagai Big Data. Mengelola dan memproses volume data ini dipandang sebagai

tantangan TI baru, dan mesin database relasional tradisional yang sangat transaksional tampaknya sudah ketinggalan zaman.

2.7. Masalah Data Terdistribusi dan Pekerjaan Terkait

Di bagian ini, kami menyajikan kendala model relasional di dunia terdistribusi, teknologi NoSQL, dan beberapa pekerjaan terkait yang diberikan untuk menangani masalah ini. Lingkungan terdistribusi adalah sekumpulan mesin fisik (node) yang berpartisipasi bersama untuk menyelesaikan pemrosesan dan penyimpanan paralel. Semua sumber daya node (CPU, memori, disk) dapat dibagikan atau tidak. Jumlah node dapat berbeda dari beberapa hingga ribuan node. Jadi, kita dapat menemukan cluster sederhana dengan beberapa node yang biasanya berbagi disk, atau lebih kompleks seperti grid yang berisi ratusan atau ribuan node di mana sumber daya sering tidak digunakan bersama. Bagian utama dari database CS didasarkan pada model relasional yang dibangun di atas konsep tabel (relasi antar data) dan operasi set-aljabar.

Model ini cocok untuk kebutuhan transaksional karena sifat ACID (Atomicity, Consistency, Isolation, dan Durability) [3], dan bekerja dengan sangat baik di lingkungan node tunggal. Sebaliknya, data relasional tidak dapat didistribusikan dan diterapkan dengan baik dalam lingkungan multi-node terdistribusi. Properti ACID menjadi kendala untuk model dan kemudian mencegah data didistribusikan secara efektif antara node [4].

Perhatikan juga bahwa kendala ACID, meskipun mereka memastikan konsistensi, dalam beberapa kasus dapat menjadi faktor pemblokiran [5, 6]. Misalnya, penyidik di internet seringkali tertarik untuk memberikan tanggapan langsung meskipun tanggapan tersebut tidak mutakhir. Akibatnya, sistem baru perlu dibuat untuk mendistribusikan dan mengelola data secara dinamis antara node dengan lebih efisien dan kegunaan. Sistem baru untuk Big Data telah muncul seperti NoSQL dan NewSQL.

3. METODOLOGI PENELITIAN

Model penelitian yang digunakan untuk mengevaluasi model orientasi dokumen ada beberapa kriteria antara lain :

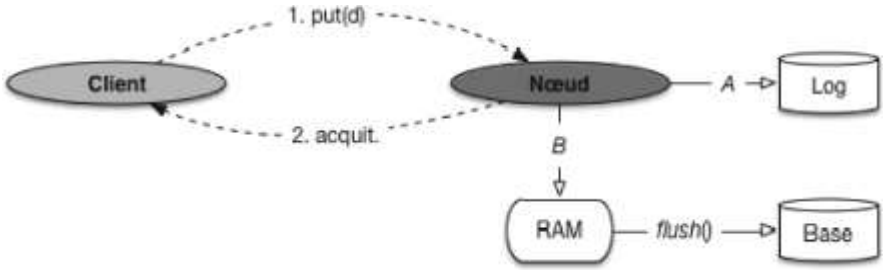
a. Sifat data (terstruktur, semi-terstruktur).
b. Hubungan data (integritas referensial, hubungan hierarki).
c. Siklus hidup data (versi, TTL).
d. Dataview (operasi CRUD).
e. Konsistensi data (properti ACID).
f. Kinerja (pengindeksan, partisi).
g. Volume penyimpanan (BigData).
h. Analisis data (agregat data).
i. Persistensi dan toleransi kesalahan (replikasi data).
j. Keamanan data (hak akses, enkripsi data).

Secara umum, data dapat terstruktur atau semi-terstruktur, bergantung pada use case yang terkait. Dalam model data relasional, tabel berisi sekumpulan baris di mana setiap baris mengelompokkan sekumpulan nilai. Hubungan yang ada didasarkan pada gabungan antar baris dalam tabel yang berbeda dan tidak didasarkan pada hubungan semantik antara kata kunci (nilai).

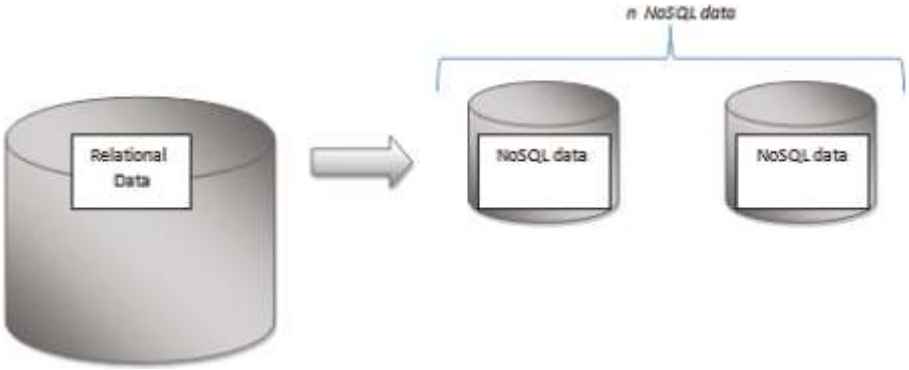
3.1. Pengajuan

Sebagai ajuan, disarankan untuk memigrasi data CS ke sistem NoSQL yang sesuai. Kami memilih sistem NoSQL sesuai dengan kriteria tertentu yang sebagian besar didasarkan pada penyesuaian NoSQLclass ke data CS. Selain itu, karena data CS harus konsisten, jenis sistem NoSQL yang dipilih harus CP (Consistency / Partitioning). Data NoSQLCS yang dimigrasi akan didistribusikan pada platform yang terdiri dari banyak node. Sharding adalah properti NoSQL yang mendistribusikan dan mengalokasikan sumber daya secara dinamis antara node dan menyesuaikannya sesuai dengan kebutuhan data. “Gambar 1,” mewakili skema untuk migrasi yang kami usulkan. Ini dapat sangat meningkatkan kinerja baik pada penyimpanan data atau waktu pemrosesan untuk kueri ad hoc (Gbr. 2).

JURNAL ILMIAH TEKNIK MESIN, ELEKTRO DAN KOMPUTER
Halaman Jurnal: http://ejurnal.stie-trianandra.ac.id/index.php/juritek
Halaman UTAMA Jurnal : http://ejurnal.stie-trianandra.ac.id/index.php



Gambar 1. Menulis dengan logging



Gambar 2. Pendekatan yang diusulkan

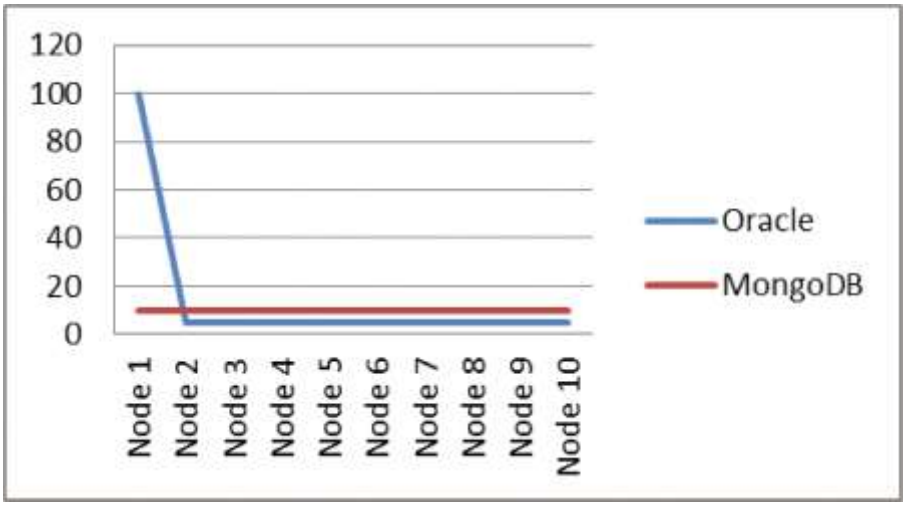
3.2. Penyusunan Data CS Target

Untuk menyiapkan data CS, kami mulai menerapkan platform terdistribusi, dan kami melakukan migrasi data dari Oracle 12c ke MongoDB versi 2.61. Beberapa percobaan dan hasilnya akan dilakukan pada bagian selanjutnya. Pertama, kami mengkonfigurasi platform terdistribusi yang dikelola langsung oleh MongoDB

4. HASIL DAN PEMBAHASAN

Database CS MongoDB baru sekarang siap; tujuan kami adalah untuk membuktikan kinerja yang diperoleh dengan menggunakan elastisitas yang diberikan oleh NoSQLdata terdistribusi. Elastisitas ini menandakan kemampuan untuk mendistribusikan data dan pemrosesan kueri pada beberapa node, bertentangan dengan kekakuan data relasional. Jadi secara teori, diharapkan kinerja yang baik. Berikut ini, kami menggunakan kueri data mining yang sama untuk data MongoDBdata terdistribusi dan data mono-node Oracle.

Kami akan membandingkan dan menganalisis hasil penyimpanan, run time, dan pemrosesan kueri bersama dengan memvariasikan setiap kali jumlah saham dari 2 hingga 10. Sebagai percobaan, kami menggunakan koleksi yang berisi data waktu nyata yang sering digunakan dalam mencari kumpulan item yang sering. Untuk penyimpanan data, kita dapat melihat pada Gambar 3 sebuah grafik yang menunjukkan bahwa MongoDB telah membagi data CS secara otomatis pada shard yang berbeda, tetapi untuk Oracle hanya satu node yang mendukung penyimpanan data, node lainnya tetap diam.



Gambar. 3. Perbandingan partisi ulang penyimpanan data CS antara MongoDB dan Oracle

5. Kesimpulan

Dari penjelasan yang telah diuraikan pada bab – bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut :

a. Sebagai percobaan, setelah mengkonfigurasi lingkungan terdistribusi dengan sejumlah node, kami telah menginstal MongoDB dan memigrasikan data CS. Properti NoSQLsharding memungkinkan partisi ulang penyimpanan dan pemrosesan antara semua node. Beberapa eksperimen query Data Ming telah dilakukan untuk membandingkan kinerja hasil antara data NoSQL multi-node dan data relasional mono-node. Secara keseluruhan, hasil akhir menunjukkan peningkatan yang menarik dalam waktu berjalan, sebagai tambahan dari penyimpanan data yang diperoleh dengan menggabungkan semua node disk.
b. Terakhir, dalam perspektif kami berharap dapat memperluas eksperimen ini di lingkungan platform yang tersebar luas dengan banyak node seperti platform Hadoop. Selain itu, karena NoSQL adalah model baru dan masih dalam pengembangan, kami mendengar sistem NoSQL baru yang potensial untuk mengujinya dan menemukan yang paling sesuai untuk data CS. Cluster berorientasi dokumen menyediakan arsitektur yang sangat skalabel dan ketersediaan sistem yang lebih baik. Karena alasan ini, model ini adalah salah satu model NoSQL yang paling banyak digunakan di seluruh dunia.

Daftar Pustaka

Kaur, K., Rani, R.: Modeling and querying data in NoSQL databases. In: BigData, 2013 IEEE International Conference. INSPEC Accession Number 13999217 (2013)

Hashem, H., Ranc, D.: An integrative Modeling of BigData Processing. Int. J. Comput. Sci. Appl. Print ISSN 0972-9038 (2014)

Sharma, V., Dave, M.: SQL and NoSQL Databases. Int. J. Adv. Res. Comput. Sci. Softw. Eng. 2(8), 2–8 (2012). ISSN:2277 128X. Research paper available: www.ijarcsse.com

Degroodt, N.: L'élasticité des bases de données sur le Cloud Computing. Master thesis in Sciences computer, FreeUniversity of Bruxelles, pp. 12–20 (2011)

Li, Y., Manoharan, S.: A performance comparison of SQL and NoSQLdatabases. In: Communications, Computers and Signal Processing, 2013 IEEE Pacific Rim Conference (2013). ISSN 1555-5798

Farhaoui, Y.: Big data and NoSQL system for control system. IJEFT 14(2) (2017)