

PENGEMBANGAN SISTEM MONITORING DAN ALERTING DENGAN MENGGUNAKAN TICK STACK PADA SISTEM OPERASI LINUX DI PT XXX

Abdi Marulita Sipahutar¹, IGN Mantra, M.Kom, MM²,
Institut Keuangan Perbankan dan Informatika Asia Program
Studi Sistem Informasi Perbanas
Jl. Cut Meutia No. 2, Sepanjang Raya, Bekasi
Indonesia
1713070012@perbanas.id

Abstrak. Tujuan dari penelitian ini adalah untuk mengembangkan sistem *monitoring* terdahulu yang telah digunakan pada PT XXX yaitu Nagios 3 dengan sebuah sistem *monitoring* baru yang lebih baik untuk memantau sistem aplikasi dan kesehatan infrastruktur jaringan perusahaan tersebut. *Paper* ini fokus untuk mengembangkan sistem *monitoring* yang ada saat ini (Nagios 3) dengan memilih salah satu dari 3 sistem *monitoring open source* terbaru berbasis *technology stack* yang sangat populer digunakan saat ini. *Tick stack* merupakan salah satu sistem *monitoring open source* berbasis *technology stack* yang banyak digunakan untuk memantau sistem aplikasi dan infrastruktur jaringan. Pada *paper* ini akan dijelaskan kelebihan dan kekurangan setiap sistem *monitoring* tersebut, mengapa peneliti memilih menggunakan *tick stack*.

Kata Kunci: Nagios, *monitoring* sistem, *alerting*, *tick stack*, *sensu*, *prometheus*, *Linux*, *Microsoft Windows*, *Mac OS*, *open source*, *Server*.

1 Pendahuluan

Teknologi Informasi (TI) adalah kombinasi teknologi komputer yang terdiri dari perangkat keras dan lunak untuk mengolah dan menyimpan informasi dengan menggunakan teknologi komunikasi untuk melakukan penyaluran informasi. Software lapisan pertama yang memiliki kontrol langsung dengan perangkat keras komputer disebut operating system (OS). OS yang banyak dikenal dan digunakan saat ini ada 3, yaitu *linux*, *mac-os*, dan *windows*. Teknologi komunikasi yang digunakan untuk menyalurkan informasi biasanya bersifat *client-Server*, dimana *server* berperan memberikan layanan dan *client* sebagai penerima layanan (pengguna). Karena kebutuhan terhadap informasi yang tinggi dan menuntut agar sistem dapat berjalan 24 jam maka dibutuhkan sebuah sistem lain yang dapat memantau sistem tersebut dan menjamin ketersediaan layanan dari *server* ke *client* selama 24 jam.

Dalam dunia IT suatu organisasi atau seseorang yang ingin memberikan layanan menggunakan sebuah sistem komputer maka penanggung jawab sistem tersebut harus memastikan sistem dapat berjalan terus menerus selama 24 jam. Terutama jika sistem layanan tersebut berkaitan dengan transaksi keuangan. Sebagai contoh sebuah sistem perbankan. Apabila sistem tiba-tiba tidak dapat memberikan layanan kepada nasabah maka pihak bank akan mengalami kerugian dan banyak keluhan dari nasabahnya. Sebuah bank biasanya memiliki banyak jumlah nasabah untuk bisa melayani seluruh nasabah tersebut pasti dibutuhkan banyak peralatan sistem. Termasuk sebuah sistem *monitoring* untuk memantau sistem layanan mereka.

Sistem *monitoring server* adalah suatu proses untuk memantau sumber daya sistem *server* tersebut seperti penggunaan *CPU*, penggunaan memori, *I/O (Input/output)*, jaringan, penggunaan disk (kapasitas penyimpanan), proses, dan lain-lain. *Monitoring server* ini membantu memahami penggunaan sumber daya sistem *server* yang dapat membantu seorang *system administrator* dalam menjamin ketersediaan sistem layanan, merencanakan peningkatan kapasitas dan memberikan hasil dan pengalaman yang lebih baik terhadap pengguna.

Contoh kasus yang paling di alami *system administrator* pada hari kerja: Suatu hari pada hari senin pagi kira-kira pukul sepuluh. Manajer cabang sangat marah karena dia menunggu email penting yang belum terkirim. Lalu *system administrator* menemukan dengan sigap menemukan sumber masalahnya yaitu pesan tidak masuk dalam antrian pesan untuk dikirim. *System administrator* juga melakukan pengecekan pada *file log* tapi tidak menemukan

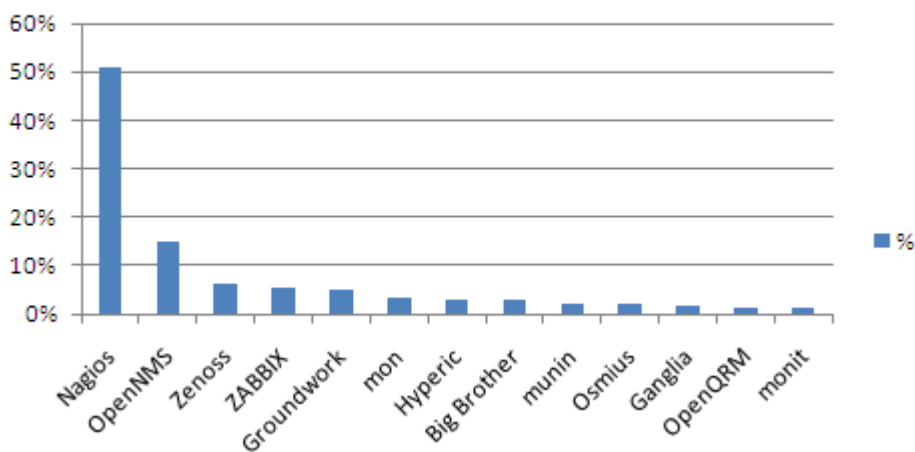
referensi apa pun baik itu error atau pun pengirim telah terkirim atau gagal terkirim. Jadi dimana masalahnya? *Server* utama email juga tidak memberikan respon terhadap program *ping*. Dengan gagalnya *Server* utama email memberikan respon terhadap program *ping* mungkin merupakan sumber dari masalah tersebut. Tetapi departemen TI membantah bahwa situasi ini bukan kesalahan mereka dan bersikukuh bahwa jaringan berjalan dengan baik di kantor pusat dan menuduh bahwa masalahnya ada di jaringan kantor cabang. Pencarian kesalahan berlanjut dan akhirnya ditemukan bahwa jalur *VPN* ke kantor pusat tidak beroperasi karena jalur cadangan berfungsi untuk mengambil alih jalur *VPN* yang rusak. Pada akhirnya banyak waktu yang terbuang hanya untuk menemukan kesalahan pada sistem.

Oleh karena itu dibutuhkan suatu sistem yaitu sebuah sistem pemantau yang dapat memantau seluruh infrastruktur dan layanan sistem untuk mencegah terjadi masalah seperti dijelaskan pada contoh diatas serta dapat membantu pengguna menganalisa penyebab atau sumber masalah kemudian dapat mengirimkan pesan peringatan dan notifikasi jika terindikasi akan menyebabkan kerusakan pada sistem yang sedang berjalan. Salah satu sistem monitoring tersebut adalah sistem monitoring *tick stack*. Dengan menggunakan sistem *monitoring* tersebut maka pengguna dapat dengan segera mencari solusi dan menangani masalah tersebut atau bahkan mencegahnya sebelum terjadi.

2 Sistem Monitoring Nagios

Pengembangan program dimulai pada tahun 1999. Nama asli proyek ini adalah *Netsaint*. Proyek ini selesai pada tahun 2002 dan selanjutnya dirubah dengan nama baru yang selanjutnya dikenal dengan nama *Nagios*. Penulis dari *nagios* ini adalah Mr. Ethan Galstad, yang saat ini juga adalah presiden *Nagios Enterprises*. *Nagios* adalah sistem pemantauan yang sangat populer pada masanya. Fakta ini juga menegaskan simposium yang dirilis pada forum diskusi 2010 untuk para penggemar distribusi *linux*.

Monitoring Solution Votes



Gambar. 1. Poling sistem *monitoring* terbaik tahun 2010

2.1 Kelemahan Nagios

User-Unfriendly GUI. Desain antarmuka pengguna berbasis *web nagios* tidak cukup menarik, karena tampilan pada umumnya terdiri dari tabel dan teks. Tampilan grafik tersedia di bagian "tren" yang hanya menampilkan kondisi *host* dan layanan (mis., Ok, peringatan, kritis) dari waktu ke waktu.

Lack of Database and Performance Records. *Nagios* tidak menggunakan *database* mungkin dengan maksud untuk membuatnya agar tetap ringan saat dijalankan. Sehingga *nagios* hanya memberikan informasi status dan pemberitahuan yang diarsipkan dalam *file log* berbasis teks. *Nagios* hanya menampilkan data kinerja saat ini (misalnya, *loss rate*, *round-trip delay*, *load*), tetapi tidak mengumpulkan data kinerja *historis*.

Difficult Configuration. Konfigurasi *nagios* hanya untuk memantau beberapa perangkat dan layanan saja sudah memerlukan prosedur yang panjang. Karena *nagios* membutuhkan *file* konfigurasi berbasis teks.

Notification using SMS, Pager, or email. Media pengiriman pesan notifikasi atau peringatan masih terbatas dan belum mampu terintegrasi dengan media pengiriman pesan terbaru yang jauh lebih baik seperti *slack*, *telegram* dan sebagainya.

3 Sistem Monitoring Berbasis Teknologi *Stack*

Technology stack merupakan singkatan dari *libraries*, *programming language*, *framework*, *patterns*, *server*, solusi *UI/UX* mana yang akan digabungkan untuk mengembangkan aplikasi seluler atau *platform* berbasis web. Setiap kali suatu produk berbasis *stack* atau layanan dirancang, ada beberapa hal yang perlu diputuskan termasuk:

- ┆ *Hosting*
- ┆ *Nodes*
- ┆ *Front-end*
- ┆ *API*
- ┆ *Database*
- ┆ *Storage*

Hosting, lingkungan dimana teknologi *stack* tersebut akan di pasangkan. *Nodes*, *operating system* apa yang akan digunakan untuk menjalankan teknologi *stack*. *Front-end*, disain antarmuka yang akan dilihat oleh pengguna teknologi *stack*. *API*, antarmuka aplikasi menggunakan teknologi *stack* yang mana. *Database*, sebagiknya untuk sistem *monitoring* menggunakan *database* seperti apa yang paling sesuai. *Storage*, seberapa besar kapasitas yang dibutuhkan untuk sistem *monitoring*. Kolaborasi teknologi tersebut membuat solusi sistem *monitoring*. Ada berbagai macam sistem *monitoring solution* berbasis *stack*, dalam *paper* ini penulist memilih 3 sistem *monitoring* berbasis *stack* yang cukup populer digunakan yaitu *tick stack*, *prometheus* dan *sensu*.

3.1 Tick Stack

Tick stack adalah kumpulan komponen berbasis *open source* yang bergabung untuk menghadirkan platform agar mudah menyimpan, memvisualisasikan, dan memantau data deret waktu seperti metrik dan kejadian (*event*). Komponennya terdiri dari: *telegraf* agen *server* untuk mengumpulkan dan melaporkan metrik; *Influxdb time-series database* berkinerja tinggi; *chronograf* antarmuka pengguna untuk platform; dan *kapacitor* mesin pengolah data yang dapat memproses, mengalirkan, dan mengumpulkan data dari *influxdb*.

3.2 Prometheus

Prometheus adalah sebuah *open source* sistem monitoring berbasis metrik. *Prometheus* memiliki model data dan *query language* yang simpel dan sangat kuat sehingga memungkinkan pengguna untuk menganalisa bagaimana sebuah sistem aplikasi berjalan serta bagaimana performa sistem infrastruktur dalam menjalankan sistem aplikasi tersebut.

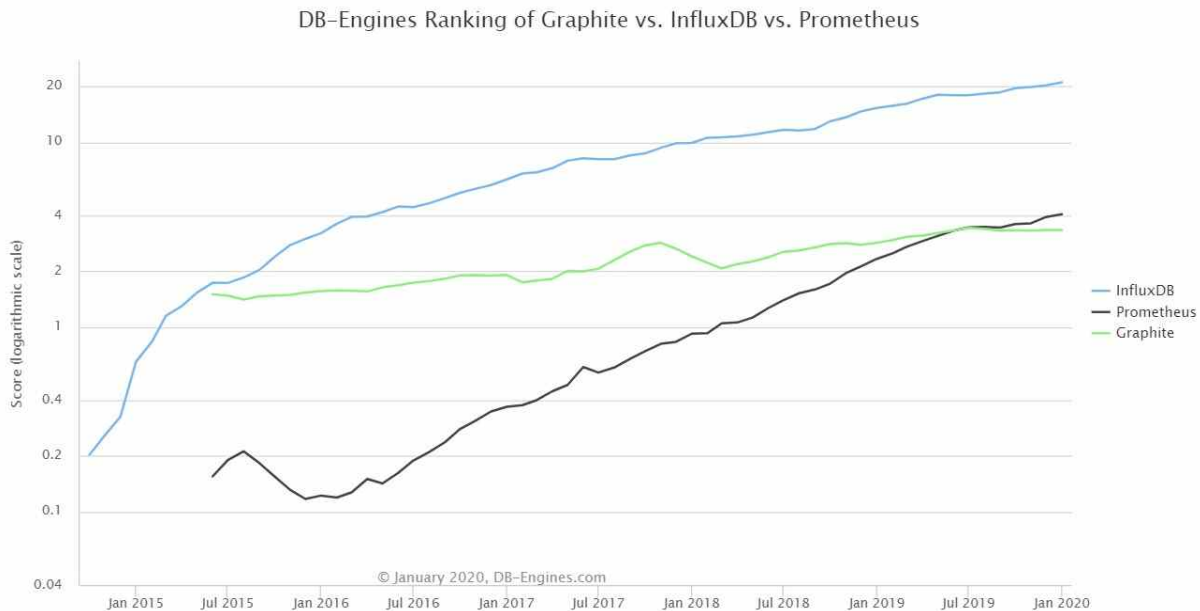
3.2 Sensu

Sensu adalah *open source monitoring event pipeline* yang dibangun untuk mengurangi beban operator dan membuat *developer* serta pemilik bisnis tanpa khawatir keadaan sistem layanan saat sedang bekerja. *Sensu* mulai dikembangkan pada tahun 2011, pendekatan *sensu* dalam menyediakan layanan *monitoring* terbilang fleksibel

karena *sensu* memecahkan tantangan dalam memantau *cloud hybrid* dan infrastruktur sementara dengan alur kerja yang terukur dan terotomatisasi serta integrasi dengan alat yang sudah digunakan oleh pengguna.

4 Tick Stack vs Sensu vs Prometheus

Untuk melihat perbandingan yang lebih jelas antara ketiga *monitoring* tersebut maka fokus perbandingannya pada *paper* ini adalah *database* yang digunakan masing-masing sistem *monitoring*. Karena ketiga sistem *monitoring* tersebut menggunakan *time-series database*. Seperti namanya, *time-series database* dirancang untuk menyimpan data yang berubah seiring waktu. Hal ini bisa berupa data apa pun yang dikumpulkan dari waktu ke waktu. Bisa itu berupa metrik yang dikumpulkan dari beberapa sistem - semua sistem yang sedang tren adalah contoh dari data deret waktu.



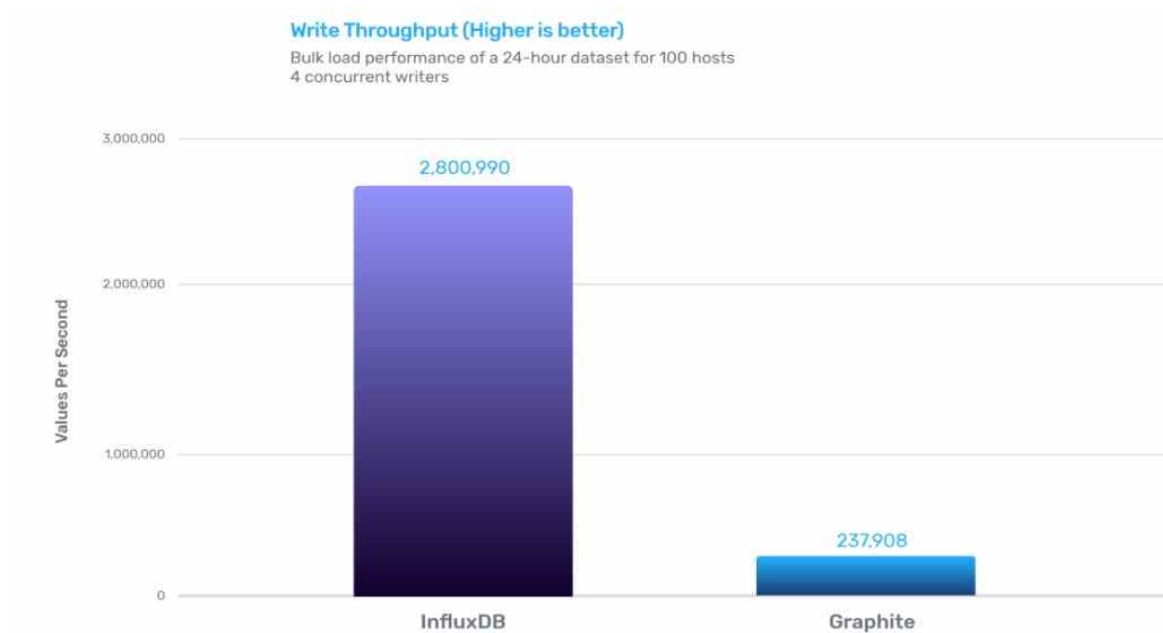
Gambar. 2. Tren popularitas *time-series database*

Dalam kategori *time-series database*, *influxdb* berada pada urutan pertama sebagai *time-series database* terbaik. Bahkan secara keseluruhan jenis *database* yang ada, diantara ketiga *database* tersebut *influxdb* tetap menjadi yang terbaik. Performa *database* sangat berperan besar dalam menyediakan layanan *monitoring* yang bersifat *real-time*. Karena jika performa *database* lambat maka hal ini sangat bertolak belakang dengan sifat *real-time*.

Editorial information provided by DB-Engines			
Name	Graphite X	InfluxDB X	Prometheus X
Description	Data logging and graphing tool for time series data	DBMS for storing time series, events and metrics	Open-source TimeSeries DBMS and monitoring system
Primary database model	Time Series DBMS	Time Series DBMS	Time Series DBMS
DB-Engines Ranking Trend Chart	Score 3.35 Rank #83 Overall #4 Time Series DBMS	Score 21.14 Rank #32 Overall #1 Time Series DBMS	Score 4.08 Rank #72 Overall #3 Time Series DBMS

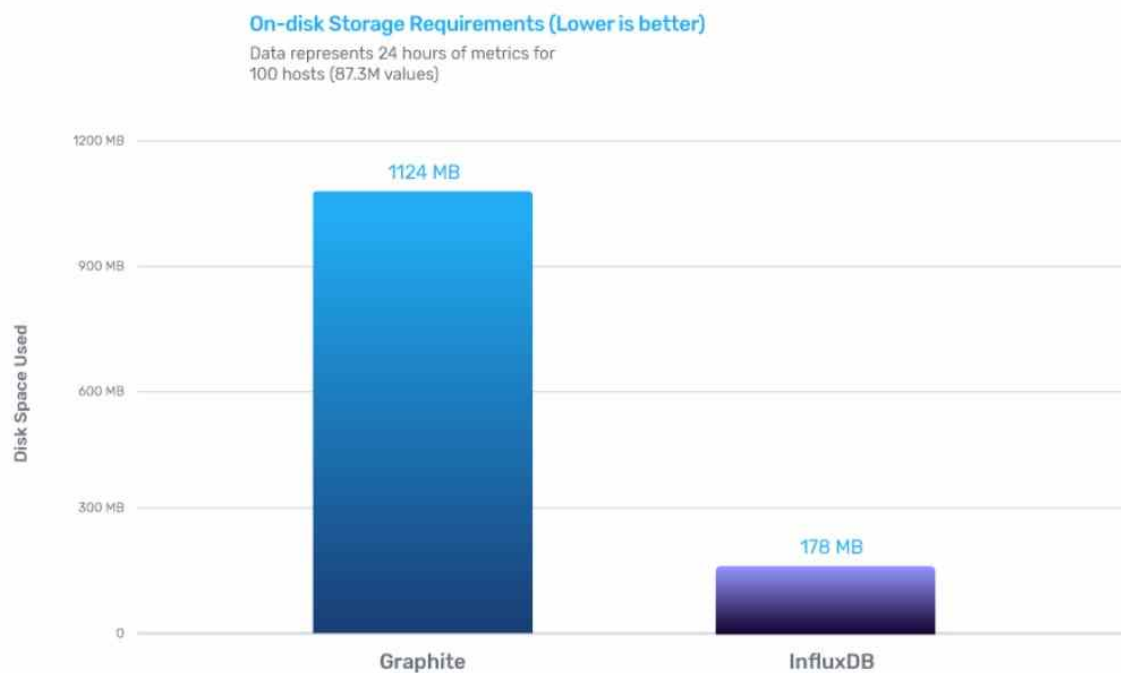
Gambar. 3. Tren popularitas *time-series database*

4.1 Performa Penulisan Ke Database



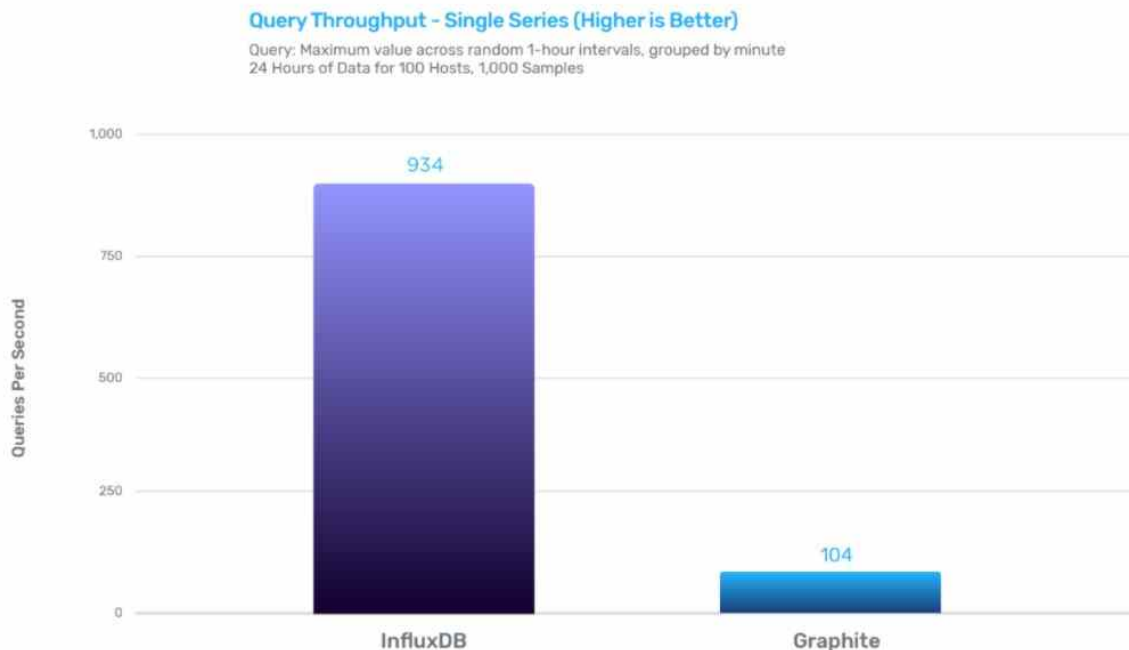
Gambar. 4. Influxdb mengungguli dibanding graphite dengan 12x untuk *data ingestion*.

4.2 On-disk Compression



Gambar. 5. Influxdb unggul 6x dibanding *graphite* untuk *data ingestion*.

4.3 Performa Query



Gambar. 6. Influxdb unggul 9x dibanding untuk *query performance*.

5 Kesimpulan

Ketiga sistem *monitoring* tersebut memiliki kelebihan masing-masing dalam memantau sistem aplikasi. Setelah membandingkan ketiga sistem *monitoring* diatas terutama kemampuan database yang mereka gunakan maka penulis memutuskan untuk menggunakan *tick stack* sebagai sistem *monitoring* yang akan digunakan pada PT XXX.

6 Referensi

- [1] Bader Andreas, Kopp Oliver, Falkenthal Michael, 2017. *Survey and Comparison of Open Source Time Series Databases*. Lecture Notes in Informatics (LNI), Gesellschaft für Informatik
- [2] Brazil Brian,. (2018). Prometheus: Up & Running Infrastructure and Application Performance Monitoring.
- [3] Churilo Chris (2019). InfluxDB vs. Graphite for Time Series Data & Metrics Benchmark, (May), <https://www.influxdata.com/blog/influxdb-outperforms-graphite-in-time-series-data-metrics-benchmark>