INTISARI

Peningkatan akses internet mengubah pola hidup masyarakat dan mendorong perusahaan beradaptasi dengan teknologi digital. Salah satu kebutuhan utama dalam era ini adalah pengelolaan *big data*. Dari beberapa variasi jenis *big data*, *Mobile Positioning Data* (MPD) merupakan data yang paling banyak dimanfaatkan oleh beberapa perusahaan atau instansi. MPD yang bersifat pasif merekam data dari *log* panggilan pengguna jaringan seluler. Namun, dalam prosesnya, pengelolaan *big data* menghadapi tantangan seperti keterbatasan memori, stabilitas sistem, dan teknik manajemen data. Python, meski populer dalam analisis data, terbatas dalam pemrosesan *multi-threaded* karena *Global Interpreter Lock* (GIL). Untuk mengatasi hal ini, digunakan PySpark, API Python untuk Apache Spark, yang memungkinkan pemrosesan paralel dan efisien untuk data berukuran besar.

Penelitian ini menggunakan metode eksperimental untuk membandingkan performa komputasi lokal dengan komputasi terdistribusi dalam pengolahan MPD pasif, sekaligus menguji batasan komputasi lokal dalam memproses sebuah data. Proses pengujian dilakukan dengan mengimplementasikan arsitektur Kubernetes untuk mengelola kluster komputasi terdistribusi. Komputasi lokal diimplementasikan dengan Python dan Spark dalam mode lokal di mana proses hanya berjalan di satu *pod* atau satu *node* saja. Beberapa parameter yang diuji dalam eksperimen ini meliputi penggunaan waktu CPU, alokasi memori, dan waktu eksekusi. Data yang digunakan memiliki tujuh variasi ukuran, mulai dari data berbasis baris (10.000 - 10.000.000 baris) hingga data berbasis ukuran (1-15 GB), untuk mengetahui batasan performa dan kelebihan masing-masing lingkungan komputasi.

Secara kuantitatif, hasil perhitungan performa menunjukkan bahwa kluster Spark unggul secara signifikan dalam waktu eksekusi (50,16 detik) dan waktu pemrosesan CPU (50,45 detik), meskipun membutuhkan penggunaan memori yang lebih besar (8 GB). Sebaliknya, lokal Spark menonjol dalam efisiensi CPU (hanya 0,27 detik), sedangkan lokal Python menawarkan keseimbangan antara penggunaan memori (5 GB) dan waktu eksekusi (263,4 detik), menjadikannya pilihan yang layak dalam skenario data berukuran relatif kecil dan sederhana.

Setiap lingkungan komputasi memiliki keunggulan dan keterbatasan yang perlu dipertimbangkan berdasarkan kebutuhan spesifik. Penelitian ini memberikan kontribusi penting dalam pemilihan teknologi komputasi yang efektif untuk kebutuhan pengolahan *big data*, sekaligus dapat menjadi sebuah referensi dalam pengembangan infrastruktur komputasi terdistribusi di masa depan.

Kata kunci: Pemrosesan *Big Data*, Komputasi Terdistribusi, *Passive Mobile Positioning Data*, PySpark

ABSTRACT

The increasing access to the internet has transformed societal behavior, prompting companies to adapt to digital technologies. One of the primary needs in this era is big data management. Among various types of big data, Mobile Positioning Data (MPD) has become one of the most widely utilized datasets by organizations and companies. Passive MPD captures data from the call logs of cellular network users. However, in terms of its process, big data management faces challenges such as limited memory, system stability, and data management techniques. Although Python is popular for data analysis, it has limitations in multi-threaded processing due to the Global Interpreter Lock (GIL). To overcome these challenges, PySpark—an API of Python for Apache Spark—enables parallel and efficient processing of large-scale data.

This research adopts an experimental method to compare the performance of local and distributed computing to process the passive MPD, while also assessing the limitations of local computing. The testing process involves the implementation of a Kubernetes architecture to manage distributed computing clusters. Local computing is implemented with Python and Spark in local mode, where processes run within a single pod or node. The experiment evaluates several parameters, including CPU time, memory allocation, and execution time. The dataset used comprises seven size variations, ranging from row-based data (10,000–10,000,000 rows) to size-based data (1–15 GB), to assess the performance limits and advantages of each computing environment.

Quantitatively, the performance evaluation results demonstrate that the Spark cluster significantly outperforms in execution time (50.16 seconds) and CPU processing time (50.45 seconds), albeit requiring higher memory usage (8 GB). Conversely, local Spark excels in CPU efficiency (only 0.27 seconds), while local Python offers a balanced approach with moderate memory usage (5 GB) and execution time (263.4 seconds), making it a viable option for scenarios involving relatively small and simple datasets.

Each computational environment has its advantages and limitations that need to be carefully considered based on specific requirements. This study provides valuable insights into selecting effective computing technologies for big data processing needs and serves as a reference for developing distributed computing infrastructure in the future.

Keywords : Big Data Processing, Distributed Computing, Passive Mobile Positioning Data, PySpark