ABSTRAK

Penerapan inferensi *Artificial Intelligence* (AI) pada arsitektur *cloud* terdistribusi rentan terhadap lonjakan trafik instan (*bursty*) yang memicu degradasi performa. Penelitian ini bertujuan menganalisis karakteristik kinerja serta mengevaluasi efektivitas konfigurasi *native* Google Kubernetes Engine (GKE) dalam menjaga stabilitas dan meminimalkan *overhead* beban kerja inferensi MobileNetV2. Metode yang digunakan adalah *empirical benchmarking* berbasis desain faktorial 3×3, menguji tiga skenario arsitektur klaster terhadap variasi beban 10, 50, dan 100 *concurrent users* menggunakan Locust. Hasil pengujian menunjukkan Skenario A mengalami *resource starvation* pada beban 100 users, menghasilkan *throughput* 14,40 RPS, latensi rata-rata 7.121,81 ms, dan *Error Rate* 70,67%. Skenario B menunjukkan stabilitas transaksi yang lebih baik (19,62 RPS, latensi 4.636,13 ms), di mana penurunan kinerja dipicu oleh kemacetan pada Ingress Gateway alih-alih saturasi komputasi CPU. Sementara itu, Skenario C mencatatkan *throughput* puncak tertinggi sebesar 19,82 RPS pada beban 100 users, namun mengalami anomali penurunan performa non-monoton pada beban menengah akibat osilasi replika (*pod thrashing*) di sekitar ambang batas *Horizontal Pod Autoscaler* (HPA). Analisis tingkat kernel pada Skenario C membuktikan bahwa pembatasan kaku Linux *cgroups* memicu *CPU Throttling* hingga 28,75%, kegagalan kontainer (*OOMKilled*), *cluster blackout* selama 3 menit, serta latensi pemulihan pod 6,40–8,54 detik dengan *Error Rate* global 71,18%. Disimpulkan bahwa konfigurasi *default* GKE kurang efektif dalam merespons beban AI yang fluktuatif ekstrem, sehingga direkomendasikan penerapan alokasi *resource burstable* dan *autoscaling* adaptif berbasis metrik kustom.

Kata Kunci: Cloud Terdistribusi, CPU Throttling, Google Kubernetes Engine, Inferensi MobileNetV2, Pod Thrashing.

ABSTRACT

Deploying Artificial Intelligence (AI) inference workloads on distributed cloud architectures is highly susceptible to instantaneous bursty traffic, which frequently triggers severe performance degradation. This study aims to analyze performance characteristics and evaluate the effectiveness of native Google Kubernetes Engine (GKE) configurations in maintaining stability and minimizing runtime overhead for MobileNetV2 inference workloads. The methodology employs an empirical benchmarking approach using a 3×3 full factorial design, evaluating three cluster architecture scenarios across 10, 50, and 100 concurrent users simulated via Locust. The experimental results reveal that Scenario A suffers from resource starvation under 100 users, yielding a throughput of 14.40 RPS, an average latency of 7,121.81 ms, and a 70.67% Error Rate. Scenario B exhibits superior transactional stability (19.62 RPS, latency of 4,636.13 ms), wherein performance bottlenecks stem from Ingress Gateway congestion rather than CPU compute saturation. Scenario C achieves the highest peak throughput of 19.82 RPS at 100 users; however, it displays a non-monotonic performance drop at medium loads due to pod thrashing around the Horizontal Pod Autoscaler (HPA) threshold. Furthermore, kernel-level forensics on Scenario C demonstrate that rigid Linux cgroups enforcement induces 28.75% CPU Throttling, container terminations (OOMKilled), a 3-minute telemetry blackout, and pod recovery durations of 6.40–8.54 seconds with an overall Error Rate of 71.18%. Consequently, default GKE configurations prove inadequate for handling extreme AI traffic surges, necessitating burstable resource allocations and custom metric-driven autoscaling.

Keywords: CPU Throttling, Distributed Cloud, Google Kubernetes Engine, MobileNetV2 Inference, Pod Thrashing.