

**ACM Transactions on Architecture and Code Optimization**  
**Makalah ini disusun dalam rangka memenuhi Tugas Mata Kuliah**  
**“Organisasi dan Arsitektur Komputer”**



**GasMan**

<b>Devon Marvellous Loen</b>	<b>(221110226)</b>
<b>Raymond Tju</b>	<b>(221111452)</b>
<b>Dylan Pratama Khu</b>	<b>(221110781)</b>
<b>Edbert Yienson</b>	<b>(221110181)</b>
<b>Charles</b>	<b>(221110156)</b>
<b>Verianto Wijaya</b>	<b>(221111507)</b>
<b>Gilbert Sachio</b>	<b>(221113162)</b>

UNIVERSITAS MIKROSKIL  
Jl. M.H Thamrin No.140, Pusat Ps., Kec. Medan Kota, Kota Medan,  
Sumatera Utara 20212  
TA 2023/2024

## **Daftar isi**

Bab 1 Pendahuluan.....	3
1.1 Latar Belakang .....	3
1.2 Permasalahan.....	3
Bab 2 Pembahasan.....	4
2.1 Metode .....	4
2.2 Hasil.....	4
2.3 Pro dan Kontra .....	4
Bab 3 Lampiran.....	6
3.1 Link Website .....	6
Bab 4 Penutup .....	7
4.1 Kesimpulan.....	7
4.2 Rekomendasi .....	7

## **Bab 1**

### **Pendahuluan**

#### **1.1 Latar Belakang**

Prefetching data adalah teknik penting dalam arsitektur komputer yang bertujuan untuk mengurangi latensi akses memori dengan mengambil data yang diperlukan sebelum dibutuhkan oleh prosesor. Teknik ini menjadi semakin krusial seiring dengan perkembangan aplikasi modern seperti analitik grafis, pembelajaran mesin, dan basis data yang memiliki kebutuhan akses memori yang tinggi. Akses memori tidak langsung (IMA) adalah pola akses memori yang terjadi ketika alamat memori konsumen dihitung dari data yang dihasilkan oleh load producer. Pola IMA ini seringkali memiliki lokalitas yang buruk karena ketergantungan nilai yang kompleks, yang menyebabkan teknik prefetching konvensional kurang efektif dalam menangani latensi memori yang panjang. Oleh karena itu, diperlukan solusi prefetching yang lebih cerdas dan adaptif untuk meningkatkan kinerja aplikasi tersebut.

#### **1.2 Permasalahan**

Masalah utama dalam prefetching IMA adalah ketidakmampuan prefetcher konvensional untuk mengidentifikasi dan memprediksi pola akses memori yang kompleks dan nonlinier. Prefetcher perangkat keras tradisional sering kali hanya mengandalkan pola akses memori yang sederhana seperti stride atau pola linear. Akibatnya, mereka tidak mampu mengatasi variasi yang dihasilkan oleh ketergantungan nilai dalam pola IMA. Selain itu, teknik run-ahead yang ada sering kali tidak cukup efektif dalam mengantisipasi kebutuhan data di masa mendatang, sehingga menghasilkan permintaan prefetching yang sporadis dan tidak optimal. Tantangan lainnya adalah minimnya informasi yang dapat dimanfaatkan oleh prefetcher untuk membuat prediksi yang akurat, yang sering kali terbatas pada aliran alamat memori saja.

## **Bab 2**

### **Pembahasan**

#### **2.1 Metode**

Untuk mengatasi permasalahan ini, artikel ini memperkenalkan Tyche, sebuah prefetcher perangkat keras yang efisien dan umum untuk pola IMA. Tyche menggunakan mekanisme propagasi bilateral untuk menggali dan mencatat ketergantungan instruksi dengan tepat. Mekanisme ini memungkinkan Tyche untuk memprediksi pola akses memori dengan lebih akurat dibandingkan dengan prefetcher konvensional. Tyche terdiri dari tiga komponen utama:

- Propagation Table (PT): Digunakan untuk mencatat ketergantungan instruksi antara load producer dan consumer.
- Dependency Chain Table (DCT): Menyimpan informasi tentang rantai ketergantungan instruksi, memungkinkan Tyche untuk melacak dan memprediksi pola akses memori.
- Address Generation Queue (AGQ): Menghasilkan alamat prefetching berdasarkan informasi yang disimpan dalam PT dan DCT, memastikan bahwa data yang diperlukan tersedia sebelum dibutuhkan oleh prosesor.

#### **2.2 Hasil**

Evaluasi kinerja Tyche menunjukkan bahwa prefetcher ini mampu meningkatkan kinerja rata-rata sebesar 16,2% dibandingkan dengan prefetcher spasial terkini, Berti. Tyche juga menunjukkan peningkatan kinerja yang signifikan dibandingkan dengan prefetcher IMA lainnya seperti IMP, Gretch, dan Vector Runahead, dengan peningkatan masing-masing sebesar 15,9%, 12,8%, dan 10,7%. Selain itu, Tyche memiliki overhead penyimpanan yang sangat rendah, hanya sebesar 0,57 KB, yang membuatnya lebih efisien dibandingkan prefetcher lainnya. Dari segi efisiensi energi, Tyche mengonsumsi rata-rata hanya 1,1% lebih banyak energi dibandingkan prefetcher lain, yang merupakan kompromi yang layak mengingat peningkatan kinerja yang diperoleh.

#### **2.3 Pro dan Kontra**

##### **Pro :**

- Peningkatan kinerja yang signifikan: Tyche mampu memberikan peningkatan kinerja yang substansial dibandingkan dengan prefetcher lain, terutama dalam menangani pola IMA yang kompleks.
- Overhead penyimpanan yang rendah: Dengan overhead penyimpanan hanya sebesar 0,57 KB, Tyche sangat efisien dalam hal penggunaan memori.
- Efisiensi energi yang baik: Tyche hanya mengonsumsi sedikit lebih banyak energi dibandingkan prefetcher lain, menunjukkan bahwa peningkatan kinerja tidak datang dengan biaya energi yang tinggi.
- Kemampuan mengidentifikasi pola IMA nonlinier: Mekanisme propagasi bilateral Tyche memungkinkan untuk mengidentifikasi dan memprediksi pola akses memori yang kompleks dan nonlinier dengan lebih akurat.

**Kontra :**

- Konsumsi energi lebih tinggi: Meskipun efisien, Tyche masih mengonsumsi lebih banyak energi dibandingkan beberapa prefetcher lainnya, yang dapat menjadi masalah dalam lingkungan dengan batasan energi yang ketat.
- Kompleksitas implementasi: Struktur dan mekanisme Tyche yang canggih mungkin memerlukan usaha dan biaya tambahan dalam pengembangan dan implementasi lebih lanjut.

**Bab 3**  
**Lampiran**

**3.1 Link Website**

**[ACM Transactions on Architecture and Code Optimization](#)**

## **Bab 4**

### **Penutup**

#### **4.1 Kesimpulan**

Tyche merupakan solusi inovatif dan efisien untuk masalah prefetching IMA, dengan peningkatan kinerja yang signifikan dan efisiensi energi yang baik. Meskipun masih ada ruang untuk peningkatan, Tyche menawarkan pendekatan yang menjanjikan untuk mengatasi tantangan dalam prefetching data pada aplikasi modern. Artikel ini memberikan kontribusi yang berharga dalam bidang arsitektur komputer dan diharapkan dapat menjadi dasar untuk penelitian dan pengembangan lebih lanjut. Dengan optimasi dan peningkatan lebih lanjut, Tyche memiliki potensi untuk menjadi prefetcher standar dalam industri teknologi informasi.

#### **4.2 Rekomendasi**

Untuk peningkatan lebih lanjut, disarankan untuk:

- Mengoptimalkan efisiensi energi: Melakukan penelitian lebih lanjut untuk mengurangi konsumsi energi Tyche tanpa mengorbankan kinerja, sehingga lebih cocok untuk aplikasi dengan batasan energi.
- Mengembangkan framework yang lebih luas: Memperluas kemampuan Tyche agar dapat diterapkan pada berbagai jenis aplikasi dan skenario penggunaan yang berbeda.
- Meningkatkan kompatibilitas: Meningkatkan kompatibilitas Tyche dengan berbagai jenis perangkat keras dan perangkat lunak untuk memperluas adopsi dan penerapan prefetcher ini dalam industri.