Proyecto Semestral

Análisis comparativo de TowerSketch y Count-Min Sketch multinivel: Evaluación de precisión

Nombre:
Roberto Artigues Escobar

Matrícula:
2019082094

Docente:
Cecilia Hernández

TOPICOS EN MANEJO DE GRANDES VOLUMENES DE DATOS 503302-1

Índice

1. Resumen

El monitoreo de flujos de red es fundamental para la gestión y seguridad de las redes modernas. Este trabajo presenta un análisis comparativo entre dos estructuras de datos probabilísticas: TowerSketch y una implementación multinivel del Count-Min Sketch. Utilizando datos reales de CAIDA, se evalúa el rendimiento de ambos esquemas en términos de precisión y eficiencia de memoria.

Los resultados demuestran la superioridad significativa de TowerSketch, alcanzando errores relativos cercanos a cero con mayores cantidades de memoria, mientras que el Count-Min Sketch multinivel mantiene errores más elevados incluso en condiciones óptimas. Este estudio proporciona evidencia empírica sobre las ventajas de la estructura jerárquica de TowerSketch para el monitoreo eficiente de flujos de red.

2. Introducción

2.1. Contexto y Motivación

En el contexto actual de las redes de computadores, el monitoreo eficiente de flujos de red se ha convertido en un desafío crítico debido a:

- El crecimiento exponencial del volumen de tráfico de red
- La necesidad de procesar y analizar datos en tiempo real
- Las limitaciones de memoria en dispositivos de red
- Los requerimientos de alta precisión en la estimación de estadísticas

Las estructuras de datos probabilísticas han emergido como una solución prometedora para estos desafíos, ofreciendo un compromiso entre precisión y uso de memoria. Existen propuestas recientes que han demostrado ventajas significativas en términos de eficiencia y precisión, como TowerSketch, que utiliza una estructura jerárquica para mejorar la estimación de flujos.

2.2. Objetivos

Este trabajo tiene como objetivos principales:

- Realizar un análisis comparativo exhaustivo entre TowerSketch y una implementación multinivel del Count-Min Sketch
- Evaluar el rendimiento de ambas estructuras bajo diferentes configuraciones de memoria
- Analizar la precisión en la estimación de flujos tanto frecuentes como infrecuentes

2.3. Alcance

El estudio se centra en:

- Implementación y evaluación de TowerSketch y Count-Min Sketch multinivel
- Análisis utilizando trazas de red reales de CAIDA
- Evaluación comparativa de métricas de error relativo y absoluto
- Comparación del comportamiento con diferentes configuraciones de memoria (4KB a 64MB)

3. Marco Teórico

3.1. Sketches Probabilísticos

Los sketches probabilísticos son estructuras de datos que permiten estimar estadísticas sobre grandes conjuntos de datos utilizando memoria sublineal. Sus principales características incluyen:

- Uso de memoria constante independiente del tamaño de entrada
- Garantías probabilísticas de error
- Trade-off ajustable entre precisión y uso de memoria

3.2. TowerSketch

TowerSketch representa un avance significativo en el diseño de sketches probabilísticos, implementando una estructura jerárquica innovadora:

- Estructura de múltiples niveles con tamaños de contador optimizados
- Cada nivel utiliza 2^{i+1} bits por contador, donde i es el nivel
- Funciones hash independientes por nivel para mejor distribución
- Empaquetamiento eficiente de contadores en palabras de 32 bits

La eficacia de TowerSketch se basa en:

- Minimización de colisiones mediante múltiples niveles
- Optimización del espacio mediante contadores de tamaño variable
- Estimación precisa basada en el mínimo valor entre niveles
- Balance eficiente entre precisión y uso de memoria

3.3. Count-Min Sketch Multinivel

El Count-Min Sketch multinivel es una extensión del CM tradicional que introduce:

- Múltiples niveles con diferentes capacidades de conteo
- Sistema de promoción basado en umbrales
- Seguimiento del nivel actual de cada elemento
- Altura fija por nivel para maximizar filas de contadores

La estructura implementa:

$$\text{Promoción si: } valor\_contador \geq umbral\_nivel \times factor\_overflow \tag{1}$$

Aunque este enfoque busca mejorar la precisión mediante la separación de flujos, los resultados experimentales demuestran que TowerSketch logra una mejor eficiencia general en términos de precisión y uso de memoria.

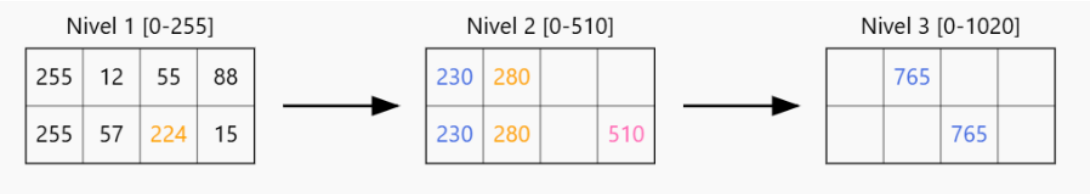

Figura 1: Estructura de ejemplo, Count-Min Sketch multinivel

4. Metodología

4.1. Diseño Experimental

El experimento fue diseñado para realizar una comparación exhaustiva entre TowerSketch y Count-Min Sketch multinivel (MCM) bajo condiciones controladas. El entorno experimental consistió en:

- **Hardware**:
  - Procesador: Intel(R) Core(TM) Ultra 7 165H (22) @ 5.00 GHz
  - Memoria RAM: 32 GB
- **Parámetros de Evaluación**:
  - Rango de memoria: 4KB a 64MB
  - Número de ejecuciones por configuración: 10 (REP_TIME)
  - Máximo de paquetes procesados: 20 millones
  - Factor de overflow para MCM: 0.50

4.2. Configuración de Estructuras

4.2.1. TowerSketch

- **Parámetros Estructurales**:
  - Número de niveles (d): 5
  - Bits por contador por nivel: [2, 4, 8, 16, 32] bits
  - Máscaras de nivel: [0x3, 0xf, 0xff, 0xffff, 0xffffff]
- **Características por Nivel**:
  - Bits de desplazamiento (cs): [1, 2, 3, 4, 5]
  - Contadores por palabra (cpw): [4, 3, 2, 1, 0]
  - Máscaras de bit bajo (lo): [0xf, 0x7, 0x3, 0x1, 0x0]

4.2.2. Count-Min Multinivel

- **Parámetros Estructurales**:
  - Profundidad (ML_DEPTH): 5 niveles
  - Altura fija (ML_HEIGHT): 5 filas
  - Bits por contador por nivel: [2, 4, 8, 16, 32] bits
  - Máscaras de nivel: [0x3, 0xf, 0xff, 0xffff, 0xffffff]
- **Características por Nivel**:
  - Bits de desplazamiento (ML_CS): [1, 2, 3, 4, 5]
  - Contadores por palabra (ML_CPW): [4, 3, 2, 1, 0]
  - Máscaras de bit bajo (ML_LO): [0xf, 0x7, 0x3, 0x1, 0x0]

4.3. Conjuntos de Datos

Se utilizó el dataset CAIDA 2018 como fuente principal de datos:

- **Características del Dataset**:
  - Formato: Registros binarios de 13 bytes por flujo
  - Tamaño: 20 millones de paquetes
  - Identificación: Tuplas únicas de 13 bytes por flujo

4.4. Métricas de Evaluación

Se implementaron cuatro métricas principales:

- **Error Relativo Promedio (ARE)**:

$$ARE = \frac{1}{n} \sum_{i=1}^{n} \frac{|e_i - t_i|}{t_i} \tag{2}$$

- **Error Absoluto Promedio (AAE)**:

$$AAE = \frac{1}{n} \sum_{i=1}^{n} |e_i - t_i| \tag{3}$$

- **Tasa de Utilización de Contadores**:

$$\text{Utilización} = \frac{\text{Contadores\_Usados}}{\text{Total\_Contadores}} \times 100 \tag{4}$$

- **Análisis por Percentiles**:
  - Separación de flujos al 80^{o} percentil
  - Evaluación independiente de ARE y AAE por grupo

4.5. Implementación de Funciones Hash

Ambas estructuras utilizan MurmurHash3 para el mapeo de flujos:

- Semillas aleatorias independientes por nivel
- Distribución uniforme para minimizar colisiones

4.6. Recolección de Datos

El proceso experimental genera cuatro archivos de resultados:

- `sketch_results.csv`: Métricas generales ARE/AAE
- `percentile_results.csv`: Análisis por tipo de flujo
- `tower_level_info.csv`: Estadísticas de TowerSketch
- `mcm_level_info.csv`: Estadísticas de MCM

5. Resultados

Los resultados detallados del experimento, incluyendo gráficos interactivos y tablas completas, están disponibles en `graficos.oracle.rartigues.com`. A continuación, se presenta un análisis de los datos obtenidos.

5.1. Análisis de Precisión

El análisis de precisión se realizó mediante la evaluación del Error Relativo Promedio (ARE) y Error Absoluto Promedio (AAE) para diferentes configuraciones de memoria:

- **Comportamiento General**: TowerSketch muestra una mejora significativa en precisión con el aumento de memoria, alcanzando un ARE de 0.00002 % con 64MB, mientras que MCM mantiene un error más alto de 8.13 %.
- **Escalabilidad**: Los resultados muestran que TowerSketch escala mejor con el aumento de memoria:
  - Con 4KB: TowerSketch ARE $\approx$ 30.597 %, MCM ARE $\approx$ 26.965 %
  - Con 64MB: TowerSketch ARE $\approx$ 0.00002 %, MCM ARE $\approx$ 8.13 %
- **Error Absoluto**: El AAE sigue un patrón similar, con TowerSketch mostrando una reducción más pronunciada del error al aumentar la memoria.

5.2. Análisis por Tipos de Flujo

El análisis de percentiles revela comportamientos distintos para flujos frecuentes e infrecuentes:

- **Flujos Frecuentes**:
  - TowerSketch muestra mejor rendimiento en general
  - La diferencia se acentúa con mayor memoria disponible
  - Convergencia más rápida hacia errores bajos
- **Flujos Infrecuentes**:
  - Ambas estructuras muestran errores más altos
  - TowerSketch mantiene ventaja en precisión
  - Mayor variabilidad en las estimaciones

5.3. Análisis de Uso de Niveles

El comportamiento de los niveles muestra patrones distintivos:

5.3.1. TowerSketch

- Distribución gradual de la utilización entre niveles
- Reducción progresiva del uso de niveles inferiores con mayor memoria
- Mejor balance en la utilización de contadores entre niveles

5.3.2. CM-Multinivel

- Alta concentración en niveles superiores
- Caída abrupta en la utilización del nivel 4 (32-bit)
- Menor eficiencia en la distribución de contadores

5.4. Eficiencia de Memoria

El análisis de eficiencia de memoria muestra que:

- TowerSketch logra mejor utilización del espacio disponible
- La estructura jerárquica de TowerSketch permite una mejor adaptación a diferentes patrones de tráfico
- MCM muestra saturación más rápida de niveles superiores

5.5. Hallazgos Clave

Los resultados principales del estudio son:

- TowerSketch supera consistentemente a MCM en precisión, especialmente con mayor memoria disponible
- TowerSketch maneja mejor tanto flujos frecuentes como infrecuentes
- La distribución de contadores es más eficiente en TowerSketch

Se realizaron los experimentos con datos autogenerados con sesgo y sin sesgo, y se encontró que TowerSketch mantiene una precisión superior en ambos casos, lo que sugiere una mayor robustez en entornos reales.

6. Conclusiones

Este estudio comparativo entre TowerSketch y Count-Min Sketch multinivel revela diferencias significativas en rendimiento y diseño estructural. TowerSketch demostró superioridad en términos de:

- **Precisión**: Mejores resultados tanto en ARE como AAE, especialmente con memoria superior a 1MB, alcanzando errores cercanos a cero con 64MB mientras CM-Multinivel mantiene errores superiores al 8 %.
- **Eficiencia Estructural**: La distribución jerárquica de TowerSketch (32,768 contadores en nivel base para 256KB) permite un mejor aprovechamiento de memoria que la distribución uniforme de CM-Multinivel (257x51 por nivel).
- **Manejo de Flujos**: Mayor efectividad tanto en flujos frecuentes como infrecuentes, evidenciado en el análisis por percentiles.

Las principales limitaciones identificadas incluyen:

- CM-Multinivel subutiliza sus niveles superiores
- Ambas estructuras muestran sensibilidad con memoria inferior a 16KB
- CM-Multinivel requiere más contadores totales (65,535 vs 63,488 en 256KB) para un rendimiento inferior

7. Trabajo Futuro

Cualquier trabajo futuro en Multilevel Count-Min Sketch debería considerar las siguientes áreas de mejora:

- Optimizar la utilización de memoria en CM-Multinivel
- Evaluar el rendimiento con diferentes datasets
- Investigar umbrales de promoción alternativos

La arquitectura actual de ML CM no logra aprovechar eficientemente la memoria disponible, lo que sugiere la necesidad de un rediseño estructural para mejorar su rendimiento.

Referencias

[1] Yang, T., Yang, H., Liu, A. X., Meng, J., Sha, Z., & Li, B. (2021). *TowerSketch: A compressed sketch for better heavy-hitter identification.* In 2021 IEEE 29th International Conference on Network Protocols (ICNP) (pp. 1-11). IEEE.

A. Código Fuente

Todo el código fuente utilizado en este trabajo se encuentra disponible en el repositorio de GitHub: `https://github.com/rartigues/ProyectoVolumDatos`. Detalle acerca de la implementación del proyecto se encuentra en el archivo README.md.