Investigación
Métricas y Ensamblador

Joseph David Jimenez Zuñiga, *2016133677, josephdjz@estudiantec.cr*

1. Respuesta 1

Pregunta: Basado en este paper(Scalability in Computing and Robotics) explique y contraste las principales leyes de escalabilidad vigentes a la fecha (apartado 1.1 Laws of Scalability).

Basándonos en el paper propuesto y en el apartado 1.1. Se pueden deducir leyes de escalabilidad vigentes hasta la fecha, entre ellas se menciona la ley de Amdahl, la ley de Gustafson y la ley universal de escalabilidad de Gunther. Las 3 leyes utilizan en parte simbología parecida pero cambiando en su sigma.

Donde:

$S_{Throughput}(N)$ es el aumento de rendimiento o Speedup al utilizar N unidades de procesamiento.

N es el número de unidades de procesamiento.

1.1. Ley de Amdahl

Propuesta por Gene Amdahl en 1967, establece que la mejora en el rendimiento de un sistema mediante la paralelización está limitada por la fracción de tiempo dedicada a la parte secuencial o serial del proceso. La ecuación de Amdahl se expresa como:

$$S_{throughput}(N) = \frac{N}{1 + \sigma(N-1)} \quad (1)$$

Donde:

σ es la fracción de tiempo que se gasta en la parte secuencial del proceso, con $0 \leq \sigma \leq 1$.

1.2. Ley de Gustafson

Propuesta por John Gustafson en 1988, Esta ley asume que el impacto de la parte secuencial en el rendimiento disminuye a medida que aumenta el tamaño del sistema. La ecuación de Gustafson se expresa como:

$$S_{throughput}(N) = N + (1 - N)\sigma \quad (2)$$

σ es la fracción de tiempo que se gasta en la parte secuencial del proceso, con$0 \leq \sigma \leq 1$.

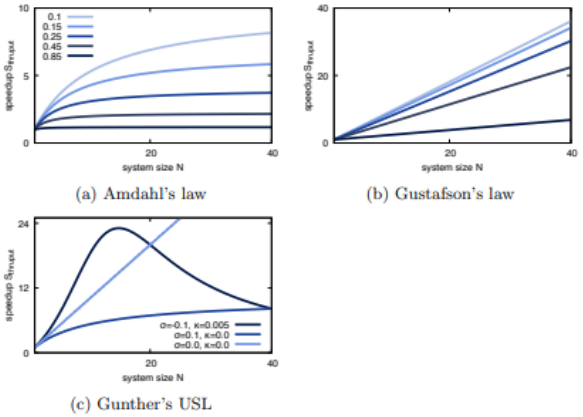
1.3. Ley universal de escalabilidad de Gunther

Propuesta por Neil J. Gunther en 1993, considera la posibilidad de que los costos de coordinación para un sistema grande puedan ser mayores que el trabajo que pueden realizar las unidades adicionales. La ecuación de la USL es:

$$S_{throughput}(N) = \frac{N}{1+\sigma(N-1)+\kappa N(N-1)} \quad (3)$$

Donde: σ representa la contención o tiempo perdido en colas de recursos compartidos, con$0 \leq \sigma \leq 1$.

K Es un parámetro adicional que tiene en cuenta el retraso de coherencia debido a la interacción de cada unidad con todas las demás unidades.



(a) Amdahl's law (b) Gustafson's law

(c) Gunther's USL

Figura 1. Leyes de escalabilidad vigentes hasta la fecha

1.4. Contraste

La Ley de Amdahl predice un aumento de rendimiento limitado a medida que se agregan más unidades de procesamiento, alcanzando un valor constante aproximado de σ^{-1} para valores grandes de N. Esto significa que aunque se aumente la cantidad de unidades de procesamiento, siempre habrá una fracción del proceso que se ejecutará de forma secuencial, lo que limita el aumento total del rendimiento. La Figura 1a muestra cómo el rendimiento, medido como speedup $S_{Throughput}$, se satura a medida que N aumenta. En contraste, Gustafson's Law predice un crecimiento ilimitado del rendimiento al aumentar el número de unidades

- *Instituto Tecnológico de Costa Rica, Área Académica de Ingeniería en Computadores, CE4301 — Arquitectura de Computadores I.*
- *Profesor. Luis Alberto Chavarría Zamora.*

de procesamiento. La ecuación (2) muestra que a medida que N crece, la influencia de la parte secuencial en el rendimiento disminuye, lo que lleva a un aumento continuo del rendimiento a medida que se agregan más unidades. Esto implica que un sistema con una alta paralelización puede obtener un rendimiento significativamente mayor a medida que se aumenta el número de unidades, como se muestra en la Figura 1b. Por otro lado, la Ley de Escalabilidad Universal de Gunther (USL) proporciona una perspectiva más completa y realista de la escalabilidad en sistemas grandes. Además de considerar la parte secuencial σ, la ecuación (3) también tiene en cuenta la contención κ y la demora de coherencia para coordinar las unidades. La Figura 1c ilustra las curvas de rendimiento obtenidas variando los valores de σ y κ en la ecuación.

Mientras Amdahl's Law advierte sobre las limitaciones de la paralelización, Gustafson's Law resalta el potencial de un aumento ilimitado del rendimiento con una alta paralelización. Sin embargo, la Ley de Escalabilidad Universal de Gunther ofrece una visión más completa, considerando tanto la parte secuencial como los costos de coordinación, lo que permite un análisis más detallado y realista del rendimiento en sistemas de gran escala.

2. RESPUESTA 2

Pregunta: Basado en el suite de punto flotante de SPEC (Standard Performance Evaluation Corporation) y explique tres de esos Benchmarks (indique lo que realiza y lo que mide o su propósito y objetivo).

2.1. SPEC Cloud® IaaS 2018

El SPEC Cloud® IaaS 2018 es una suite de pruebas desarrollada por SPEC para medir el rendimiento de plataformas de infraestructura como servicio (IaaS) en la nube, ya sean públicas o privadas. El objetivo de este benchmark es evaluar tanto el aprovisionamiento como los aspectos de tiempo de ejecución de una nube utilizando cargas de trabajo intensivas en E/S y CPU dentro del cómputo en la nube.

2.2. SPECjEnterprise® 2018 Web Profile Benchmark

El SPECjEnterprise® 2018 Web Profile es un benchmark de SPEC diseñado para medir el rendimiento de servidores de aplicaciones Java EE 7 Web Profile y su infraestructura de soporte, como JVM, base de datos, CPU, disco y servidores. El objetivo principal del benchmark es proporcionar un método estándar para medir la capacidad de una plataforma de virtualización para modelar un entorno virtual dinámico de un centro de datos.

2.3. SPECvirt® Datacenter 2021 Benchmark

El SPECvirt® Datacenter 2021 es el benchmark de virtualización de próxima generación de SPEC para medir el rendimiento de un centro de datos a gran escala. Es un benchmark de varios hosts que utiliza cargas de trabajo simuladas y reales para medir la eficiencia general de las soluciones de virtualización y sus entornos de gestión. El objetivo evalúa el rendimiento de un centro de datos virtualizado con un enfoque en la optimización del servidor, la flexibilidad y la disponibilidad de aplicaciones, mientras se reducen los costos.

3. RESPUESTA 3

Pregunta: Explique en que consiste el Benchmark CoreMark, interprete los resultados de la tabla 3 de este enlace.

El Benchmark CoreMark es una herramienta de evaluación de rendimiento diseñada específicamente para medir la funcionalidad de un núcleo de procesador. Produce un puntaje de un solo número que permite a los usuarios realizar comparaciones rápidas entre diferentes procesadores.

La tabla 1 nos permite comparar la eficiencia relativa de los dos benchmarks en términos de la cantidad de instrucciones necesarias para realizar el mismo trabajo.

ISA or Implementation	Benchmarks			
	Dhrystone (100 loops)		CoreMark (1 loop)	
	Instruction Counts	Conversion Ratio	Instruction Counts	Conversion Ratio
RISC-V Compiled by GCC	32711	\	306242	\
ARMv6-M Compiled by ARMCC	28105	\	442604	\
Only Binary Interpretation	128608	4.58	1963327	4.44
Optimize Flags	52005	1.85	654065	1.48
Optimize Flags and Branch	45605	1.62	492766	1.11

Figura 2. El número de instrucciones ejecutadas en el ciclo principal de dos puntos de referencia bajo diferentes ISA y optimizaciones

RISC-V Compiled by GCC: En este caso, Dhrystone ejecuta 32,711 instrucciones en 100 loops, mientras que CoreMark ejecuta 306,242 instrucciones en un solo loop. CoreMark ejecuta significativamente más instrucciones que Dhrystone (casi 10 veces más) para lograr el mismo trabajo. Esto indica que CoreMark es menos eficiente que Dhrystone en este escenario particular.

ARMv6-M Compiled by ARMCC: En este caso, Dhrystone ejecuta 28,105 instrucciones en 100 loops, mientras que CoreMark ejecuta 442,604 instrucciones en un solo loop. CoreMark ejecuta aproximadamente 15 veces más instrucciones que Dhrystone para lograr el mismo trabajo.

Only Binary Interpretation: Esta implementación de Dhrystone requiere 128,608 instrucciones para ejecutar el mismo trabajo que CoreMark en un solo loop, CoreMark sigue siendo más eficiente que Dhrystone, ya que Dhrystone requiere 4.58 veces más instrucciones para ejecutar el mismo trabajo.

Optimize Flags: En este caso, Dhrystone requiere 52,005 instrucciones para ejecutar el mismo trabajo que CoreMark en un solo loop. CoreMark sigue siendo más eficiente que Dhrystone, ya que Dhrystone requiere 1.85 veces más instrucciones para realizar el mismo trabajo.

Optimize Flags and Branch: En esta implementación, Dhrystone requiere 45,605 instrucciones para ejecutar el mismo trabajo que CoreMark en un solo loop, lo que representa una relación de conversión de 1.62. Una vez más, CoreMark es más eficiente que Dhrystone.CoreMark es más eficiente que Dhrystone, ya que Dhrystone requiere 1.62 veces más instrucciones para lograr el mismo trabajo.

Se puede concluir que CoreMark es más eficiente que Dhrystone en todos los casos presentados en la tabla.

REFERENCIAS

[1] Hamann, H; Reina, A. (2021, June 14). Scalability in Computing and Robotics. Institute of Computer Engineering, University of Lübeck, Lübeck, Germany; IRIDIA, Université Libre de Bruxelles, Brussels, Belgium; Department of Computer Science, University of Sheffield, Sheffield, UK.

[2] SPEC (Standard Performance Evaluation Corporation). (2023). Floating-Point Benchmark Suite. Recuperado de http://www.spec.org/floating-point-benchmarks

[3] Cheng, Y., Huang, L., Cui, Y., Ma, S., Wang, Y., Sui, B. (2020). Efficient Multiple-ISA Embedded Processor Core Design Based on RISC-V. En Proceedings of the International Workshop on Computer Architecture Research with RISC-V (CARRV) (pp. xx-xx). Recuperado de https://carrv.github.io/2020/papers/CARRV2020$_paper_4Cheng.pdf$.