

```
1 [ 0.0%] 65 [ 0.0%] 129 [ 100.0%] 1 [ 100.0%]
2 [ 0.0%] 66 [ 0.0%] 130 [ 0.0%] 2 [ 100.0%]
3 [ 0.0%] 67 [ 0.0%] 131 [ 100.0%] 3 [ 100.0%]
4 [ 1.0%] 68 [ 0.0%] 132 [ 0.0%] 4 [ 100.0%]
5 [ 1.0%] 69 [ 0.0%] 133 [ 0.0%] 5 [ 100.0%]
6 [ 1.0%] 70 [ 0.0%] 134 [ 0.0%] 6 [ 100.0%]
7 [ 0.0%] 71 [ 0.0%] 135 [ 100.0%] 7 [ 100.0%]
8 [ 1.0%] 72 [ 0.0%] 136 [ 0.0%] 8 [ 100.0%]
9 [ 2.1%] 73 [ 0.0%] 137 [ 100.0%] 9 [ 100.0%]
10 [ 1.0%] 74 [ 0.0%] 138 [ 0.5%] 10 [ 100.0%]
11 [ 1.0%] 75 [ 0.0%] 139 [ 0.0%] 11 [ 100.0%]
12 [ 1.0%] 76 [ 0.0%] 140 [ 0.0%] 12 [ 100.0%]
13 [ 1.0%] 77 [ 100.0%] 141 [ 0.0%] 13 [ 100.0%]
14 [ 0.0%] 78 [ 0.5%] 142 [ 100.0%] 14 [ 100.0%]
15 [ 0.0%] 79 [ 100.0%] 143 [ 0.0%] 15 [ 100.0%]
16 [ 1.0%] 80 [ 0.0%] 144 [ 0.0%] 16 [ 100.0%]
17 [ 1.5%] 81 [ 100.0%] 145 [ 0.0%] 17 [ 100.0%]
18 [ 1.0%] 82 [ 0.0%] 146 [ 0.0%] 18 [ 100.0%]
19 [ 1.0%] 83 [ 0.0%] 147 [ 0.5%] 19 [ 100.0%]
20 [ 1.0%] 84 [ 0.0%] 148 [ 0.0%] 20 [ 100.0%]
21 [ 4.0%] 85 [ 100.0%] 149 [ 0.0%] 21 [ 100.0%]
22 [ 0.0%] 86 [ 100.0%] 150 [ 0.0%] 22 [ 100.0%]
23 [ 0.0%] 87 [ 100.0%] 151 [ 0.0%] 23 [ 100.0%]
24 [ 1.0%] 88 [ 0.0%] 152 [ 0.5%] 24 [ 100.0%]
25 [ 0.0%] 89 [ 100.0%] 153 [ 0.0%] 25 [ 100.0%]
26 [ 0.0%] 90 [ 100.0%] 154 [ 0.0%] 26 [ 100.0%]
27 [ 0.5%] 91 [ 100.0%] 155 [ 0.0%] 27 [ 100.0%]
28 [ 1.0%] 92 [ 0.0%] 156 [ 0.0%] 28 [ 100.0%]
29 [ 0.0%] 93 [ 0.0%] 157 [ 100.0%] 29 [ 100.0%]
30 [ 1.0%] 94 [ 0.0%] 158 [ 0.0%] 30 [ 100.0%]
31 [ 2.0%] 95 [ 100.0%] 159 [ 0.0%] 31 [ 100.0%]
32 [ 1.0%] 96 [ 0.0%] 160 [ 0.5%] 32 [ 100.0%]
33 [ 0.0%] 97 [ 100.0%] 161 [ 0.0%] 33 [ 100.0%]
34 [ 0.0%] 98 [ 100.0%] 162 [ 0.0%] 34 [ 100.0%]
35 [ 0.0%] 99 [ 100.0%] 163 [ 0.0%] 35 [ 100.0%]
36 [ 1.0%] 100 [ 0.0%] 164 [ 0.0%] 36 [ 100.0%]
37 [ 0.5%] 101 [ 100.0%] 165 [ 0.0%] 37 [ 100.0%]
38 [ 0.0%] 102 [ 0.0%] 166 [ 100.0%] 38 [ 100.0%]
39 [ 0.0%] 103 [ 0.0%] 167 [ 0.0%] 39 [ 100.0%]
40 [ 1.0%] 104 [ 0.0%] 168 [ 0.5%] 40 [ 100.0%]
41 [ 1.5%] 105 [ 100.0%] 169 [ 0.0%] 41 [ 100.0%]
42 [ 0.0%] 106 [ 0.0%] 170 [ 100.0%] 42 [ 100.0%]
43 [ 0.0%] 107 [ 100.0%] 171 [ 0.0%] 43 [ 100.0%]
44 [ 3.0%] 108 [ 100.0%] 172 [ 100.0%] 44 [ 100.0%]
45 [ 0.0%] 109 [ 0.0%] 173 [ 0.5%] 45 [ 100.0%]
46 [ 1.0%] 110 [ 0.0%] 174 [ 0.0%] 46 [ 100.0%]
47 [ 2.0%] 111 [ 100.0%] 175 [ 0.5%] 47 [ 100.0%]
48 [ 0.5%] 112 [ 0.0%] 176 [ 100.0%] 48 [ 100.0%]
49 [ 3.1%] 113 [ 100.0%] 177 [ 0.0%] 49 [ 100.0%]
50 [ 0.0%] 114 [ 0.0%] 178 [ 100.0%] 50 [ 100.0%]
51 [ 0.5%] 115 [ 100.0%] 179 [ 0.0%] 51 [ 100.0%]
52 [ 0.0%] 116 [ 100.0%] 180 [ 0.0%] 52 [ 100.0%]
53 [ 1.0%] 117 [ 0.0%] 181 [ 0.0%] 53 [ 100.0%]
54 [ 1.0%] 118 [ 0.0%] 182 [ 0.0%] 54 [ 100.0%]
55 [ 0.0%] 119 [ 100.0%] 183 [ 0.0%] 55 [ 100.0%]
56 [ 0.0%] 120 [ 100.0%] 184 [ 0.0%] 56 [ 100.0%]
57 [ 0.5%] 121 [ 100.0%] 185 [ 0.0%] 57 [ 100.0%]
58 [ 1.0%] 122 [ 0.0%] 186 [ 0.0%] 58 [ 100.0%]
59 [ 1.0%] 123 [ 0.0%] 187 [ 0.0%] 59 [ 100.0%]
60 [ 1.0%] 124 [ 0.0%] 188 [ 0.0%] 60 [ 100.0%]
61 [ 1.0%] 125 [ 1.5%] 189 [ 0.0%] 61 [ 100.0%]
62 [ 1.0%] 126 [ 0.0%] 190 [ 0.0%] 62 [ 100.0%]
63 [ 1.0%] 127 [ 0.0%] 191 [ 0.0%] 63 [ 100.0%]
64 [ 0.0%] 128 [ 100.0%] 192 [ 0.0%] 64 [ 100.0%]
Mem [ 7.56G/94.3G] Tasks: 100, 176 thr; 65 running
Swap [ 0K/16.0G] Load average: 64.17 64.06 64.06
Uptime: 2 days, 07:53:52
```

Performance

Carlos Bederián, carlos.bederian@unc.edu.ar
Nicolas Wolovick, nicolasw@famaf.unc.edu.ar

PID	USER	PRI	NI	VRT	RES	SHR	S	CPU%	MEM%	TIME+	Command	PID	USER	PRI
97938	spaz	20			128	R	99	0.1	30	05:39	/opt/nand/2.13-mpi-intel_2018/nand2 otfp.nand	97938	spaz	20
97922	spaz	20			136	R	99	0.1	30	05:39	/opt/nand/2.13-mpi-intel_2018/nand2 otfp.nand	97922	spaz	20
97928	spaz	20			99	R	99	0.1	30	05:39	/opt/nand/2.13-mpi-intel_2018/nand2 otfp.nand	97928	spaz	20
97885	spaz	20			98	R	99	0.1	30	05:39	/opt/nand/2.13-mpi-intel_2018/nand2 otfp.nand	97885	spaz	20
97917	spaz	20			120	R	99	0.1	30	05:40	/opt/nand/2.13-mpi-intel_2018/nand2 otfp.nand	97917	spaz	20
97906	spaz	20			164	R	99	0.1	30	05:41	/opt/nand/2.13-mpi-intel_2018/nand2 otfp.nand	97906	spaz	20

CCAD

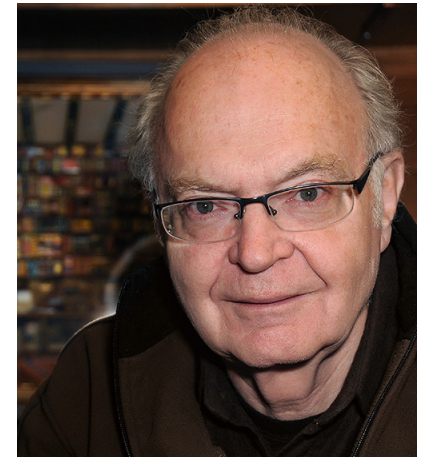
UNC

Motivación

"We should forget about small efficiencies, say about 97% of the time:

premature optimization is the root of all evil "

- Donald Knuth



Primero, antes que nada...

- Hasta dónde llega mi hardware?
- Cómo veo si es malo mi código?



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

CPU: Performance pico

Ejercicio teórico: GFLOPS que le puedo exprimir al procesador si se alinean todos los astros

1. Identificar las unidades de vectores en el diagrama de la arquitectura y su ancho
2. Ver si soportan FMA, que duplica la cantidad de operaciones por ciclo (caso contrario: 1 por unidad)
3. Buscar la frecuencia máxima de las unidades de vectores en el modo más ancho
4. Multiplicar todo:

$$GFLOPS_{64} = Cores * Unidades_{64} * \frac{Ancho}{64} * Frecuencia * FMA$$

$$GFLOPS_{32} = Cores * Unidades_{32} * \frac{Ancho}{32} * Frecuencia * FMA$$



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

CPU round up

Equipo	Procesador	Cores GHz		FPU's x ancho	FMA GFLOPS	
CCAD Cristina (2010)	2x Xeon E5420 (2007)	4	2.5	2 x 128	no	80
CCAD Mendieta (2013)	2x Xeon E5-2680 (2012)	8	2.7	2 x 256	no	346
CSC TUPAC (2015)	4x Opteron 6276 (2011)	8	2.3	2 x 128	sí	589
CCAD Eulogia (2017)	1x Xeon Phi 7210 (2016)	64	1.3	2 x 512	sí	2662
CCAD Mulatona (2018)	2x Xeon E5-2683 v4 (2016)	16	1.7	2 x 256	sí	870
SMN Huayra Muyu (2019)	2x Xeon Gold 6142 (2017)	16	1.6	2 x 512	sí	1638
Gaming PC (2019)	1x Core i9-9900K (2018)	8	3.6	2 x 256	sí	461



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Un segundo...

Equipo	Procesador	Cores GHz		FPUx x ancho	FMA GFLOPS	
CCAD Mulatona (2018)	2x Xeon E5-2683 v4 (2016)	16	1.7	2 x 256	sí	870
SMN Huayra Muyu (2019)	2x Xeon Gold 6142 (2017)	16	1.6	2 x 512	sí	1638



Intel® Xeon® Processor E5-2683 v4

40M Cache, 2.10 GHz



Intel® Xeon® Gold 6142 Processor

22M Cache, 2.60 GHz



CCAD

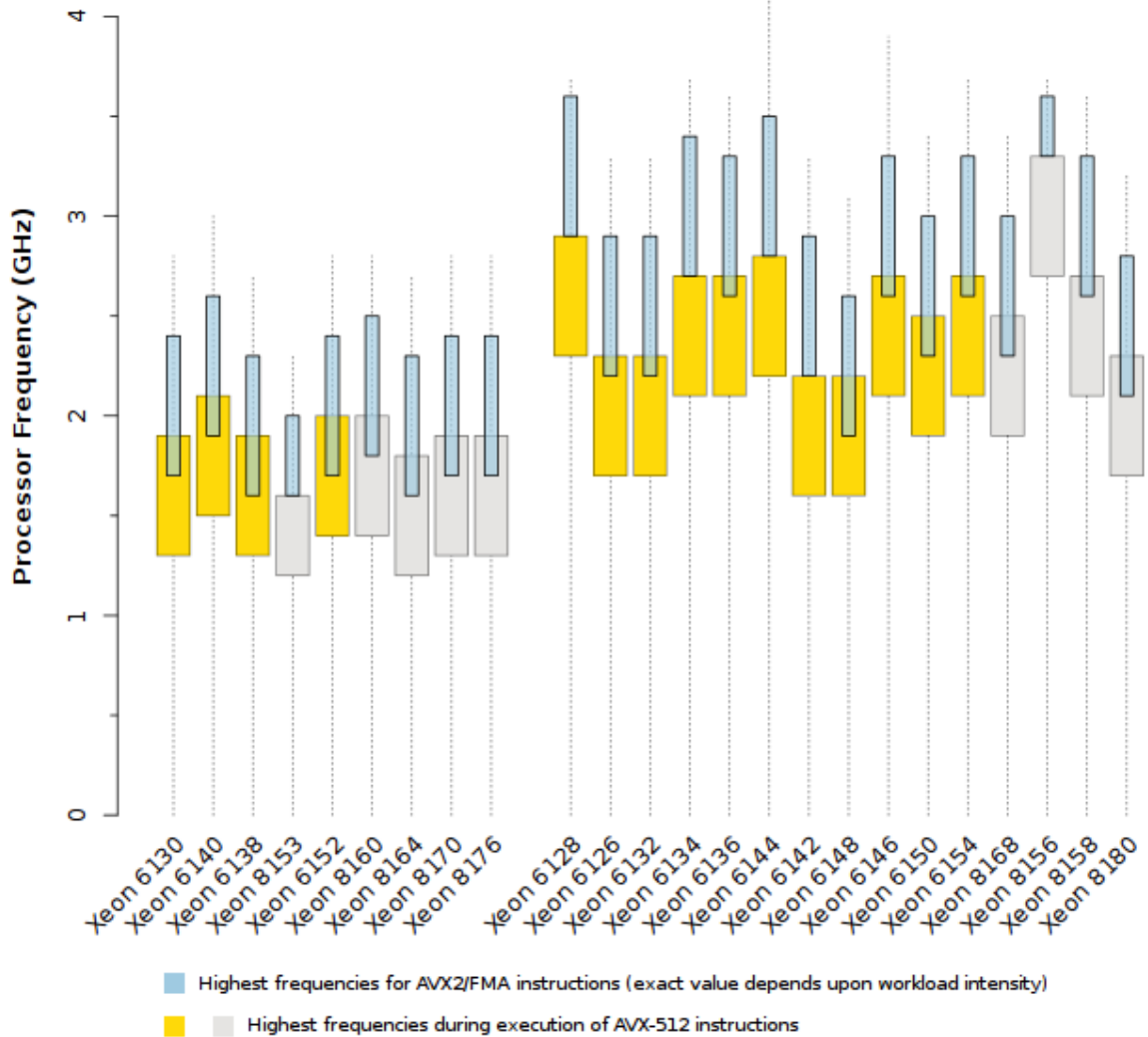
Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Comparison of All-Core Turbo Boost Ranges (AVX-512 vs. AVX2)



Microway Inc., Detailed Specifications of the “Skylake-SP” Intel Xeon Processor Scalable Family CPUs



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Throttling



Since a significant portion of the power consumption under heavy load is due to leakage current (which increases very rapidly with temperature), a more aggressive cooling solution that keeps the die temperature lower should also help the processor attain higher Turbo frequencies before hitting the power limit.

- John McCalpin a.k.a. Dr. Bandwidth



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Memoria: Performance pico

Ejercicio teórico: GBps que le puedo exprimir a la memoria si se alinean todos los astros

1. Identificar la velocidad (en gigatransferencias/s) de la memoria
 - Cada transferencia es de 64 bits
2. Ver la cantidad de canales de memoria que soporta cada procesador
 - Revisar que estén efectivamente poblados!
 - `sudo dmidecode -t memory`
3. Ver la cantidad de procesadores instalados
4. Multiplicar

$$GBps = \text{Procesadores} * \text{Canales} * \text{Velocidad} * 8$$



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

CPU: Performance práctica

El benchmark más utilizado en HPC es LINPACK

- Precisión doble
- Resuelve un sistema denso de ecuaciones lineales $Ax = B$
- Factorización LU con pivoting parcial e intercambio de filas (dgesv de LAPACK)
- $\frac{2}{3}N^3 + 2N^2$ operaciones con N arbitrario

Highly-Parallel LINPACK

- Versión distribuida de LINPACK usando MPI
- Llama a BLAS para operaciones matriciales



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

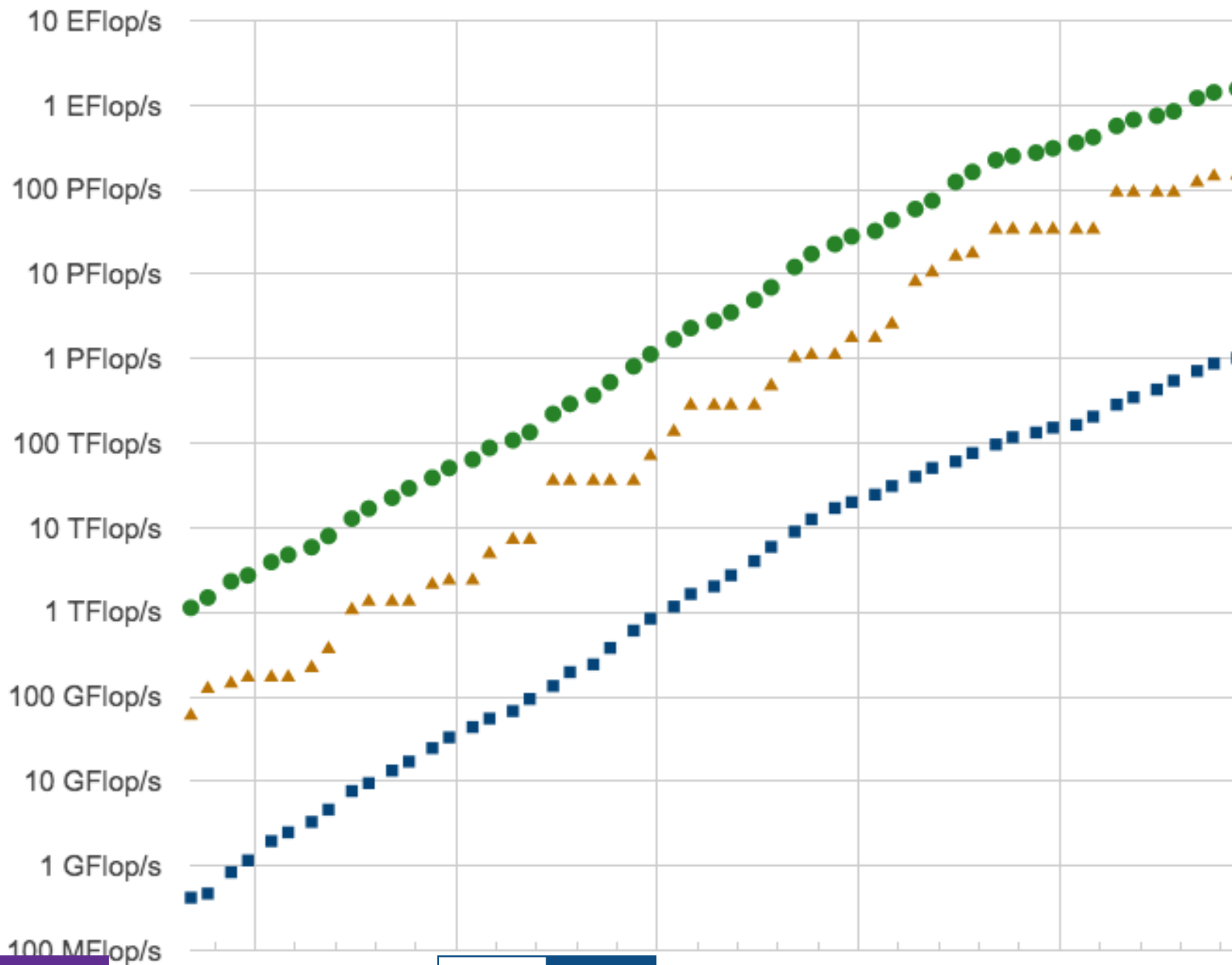
Universidad
Nacional
de Córdoba

Top500

Lista de las 500 supercomputadoras más potentes ordenadas por performance obtenida en HPL.

- Datos desde 1993 (!!!)
- Información detallada de la configuración de cada cluster
- Dos ediciones anuales:
 - Junio (ISC)
 - Noviembre (SC)
- Nuevos benchmarks similares: Green500, HPCG, Graph500, IO500





Goodhart's Law

When a measure becomes a target, it ceases to be a good measure.

Double-precision FPUs in High-Performance Computing: an Embarrassment of Riches?

- La mitad del poder de cómputo de un Xeon Phi 72x0 se desperdicia, **incluso en HPL**
- Una unidad de precisión doble ocupa $\sim 3x$ más transistores que el equivalente en precisión simple



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

STREAM

Benchmark de ancho de banda de memoria escrito por John McCalpin

Para $i \in [1..N]$ con arreglos de N elementos que son mucho más grandes que la cache

- Copy: $C[i] = A[i];$
- Scale: $B[i] = \text{scalar} * C[i];$
- Add: $C[i] = A[i] + B[i];$
- Triad: $A[i] = B[i] + \text{scalar} * C[i];$



CCAD

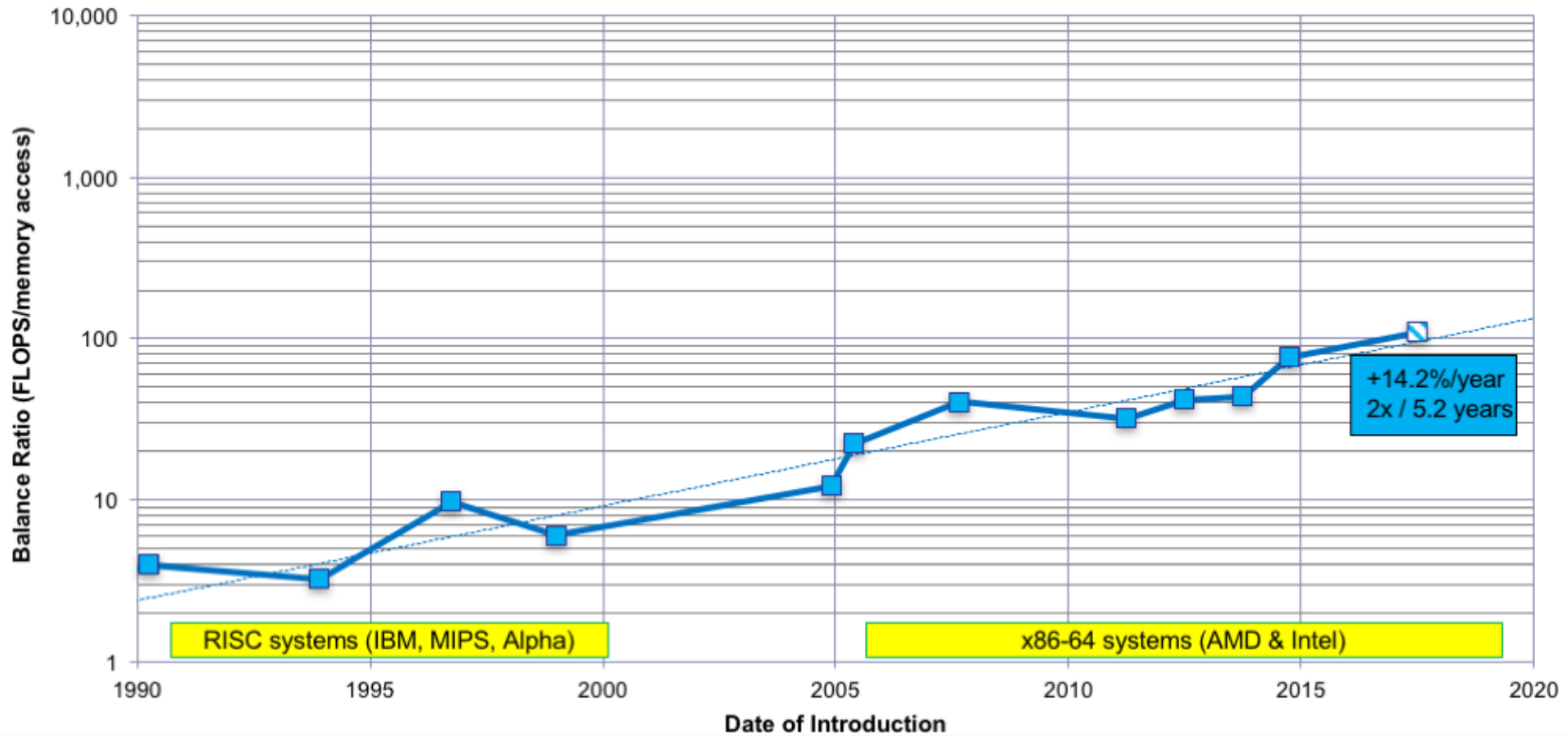
Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Memory Bandwidth is Falling Behind: (GFLOP/s) / (GWord/s)



THE UNIVERSITY OF TEXAS AT AUSTIN
TEXAS ADVANCED COMPUTING CENTER

Fuente: John McCalpin, Memory Bandwidth and System Balance in HPC Systems



Centro de
Computación
de Alto
Desempeño



Universidad
Nacional
de Córdoba

Intensidad aritmética

Para operar sobre datos es necesario transportar datos desde memoria al procesador, y viceversa.

Para un problema dado, la intensidad aritmética es el ratio entre la cantidad de operaciones y los bytes transferidos para realizarlas.

- Intensidad baja: Suma de dos matrices en precisión simple, $O(n)$ operaciones

1 op, 12 bytes (2 lecturas, 1 escritura) por elemento = 0.083 ops/byte

- Intensidad alta: Multiplicación de dos matrices $N \times N$ en precisión simple (optimista), $O(n^3)$ operaciones

$N^3 - N$ ops, $4 * 3N^2$ bytes transferidos $\approx \frac{N}{12}$ ops/byte

Para problemas de intensidad aritmética baja, el límite no es el poder de cómputo sino la capacidad de transferir datos.



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Roofline Performance Model

Modelo del límite efectivo de un nodo de cómputo.

1. Medimos el poder de cómputo máximo (R_{max} , en GFLOPS) con **HPL**
 - Techo para problemas de alta intensidad aritmética
2. Medimos el ancho de banda de memoria ($MemBW$, en GBps) con **STREAM**
 - Techo para problemas de baja intensidad aritmética
3. $\frac{R_{max}}{MemBW}$: Intensidad aritmética mínima para utilizar todo el poder de cómputo del procesador
4. Graficar GFLOPS máximo respecto a la intensidad aritmética



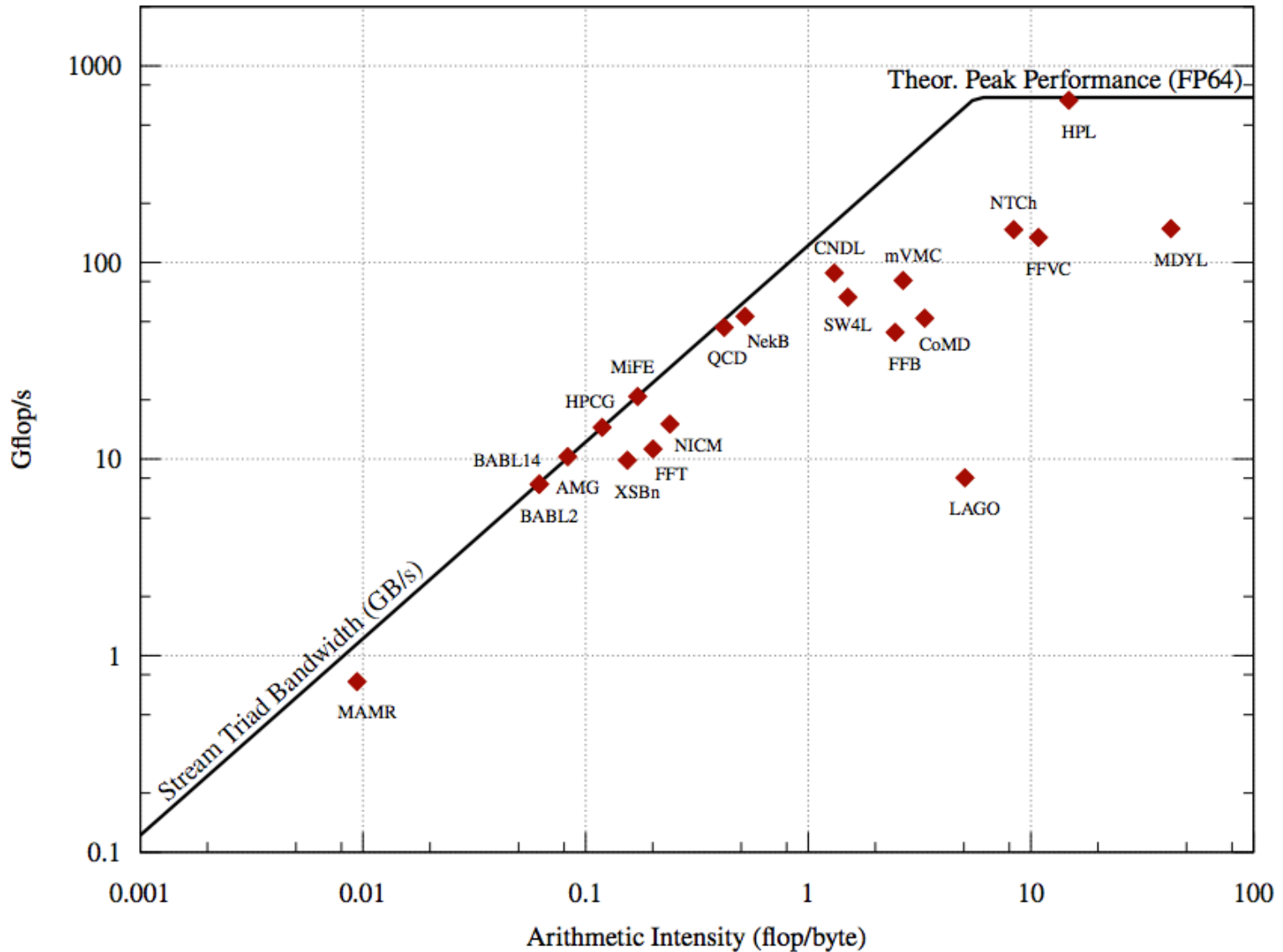
CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba



Fuente: Double-precision FPUs in High-Performance Computing: an Embarrassment of Riches?

¿Cómo procedo?

- Obtener el gráfico
 - Sin renegar: usar Intel Advisor
- Si no llegamos al techo, subir
 - Exprimir más el procesador (vectorización, paralelización)
 - Nota: El techo para problemas que no pueden usar Fused Multiply-Add (FMA) es la mitad
- Si el techo nos limita, ir hacia la derecha
 - Aumentar la localidad del código para operar más en cache



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Escalabilidad: Speedup

Speedup se define como el ratio de mejora sobre una implementación de referencia:

$$S = \frac{T_{orig}}{T_{new}}$$

Donde T_{orig} y T_{new} son tiempos de ejecución.

- Se usa para comparar una implementación nueva con una de referencia, generalmente una paralela contra la secuencial original:

$$S = \frac{T_1}{T_N}$$



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Escalabilidad: Eficiencia

Eficiencia muestra cuánto perdimos de lo teóricamente posible a la práctica:

$$Eff = \frac{M_{efectivo}}{M_{teorico}}$$

En el caso de paralelización:

$$Eff = \frac{S}{N}$$



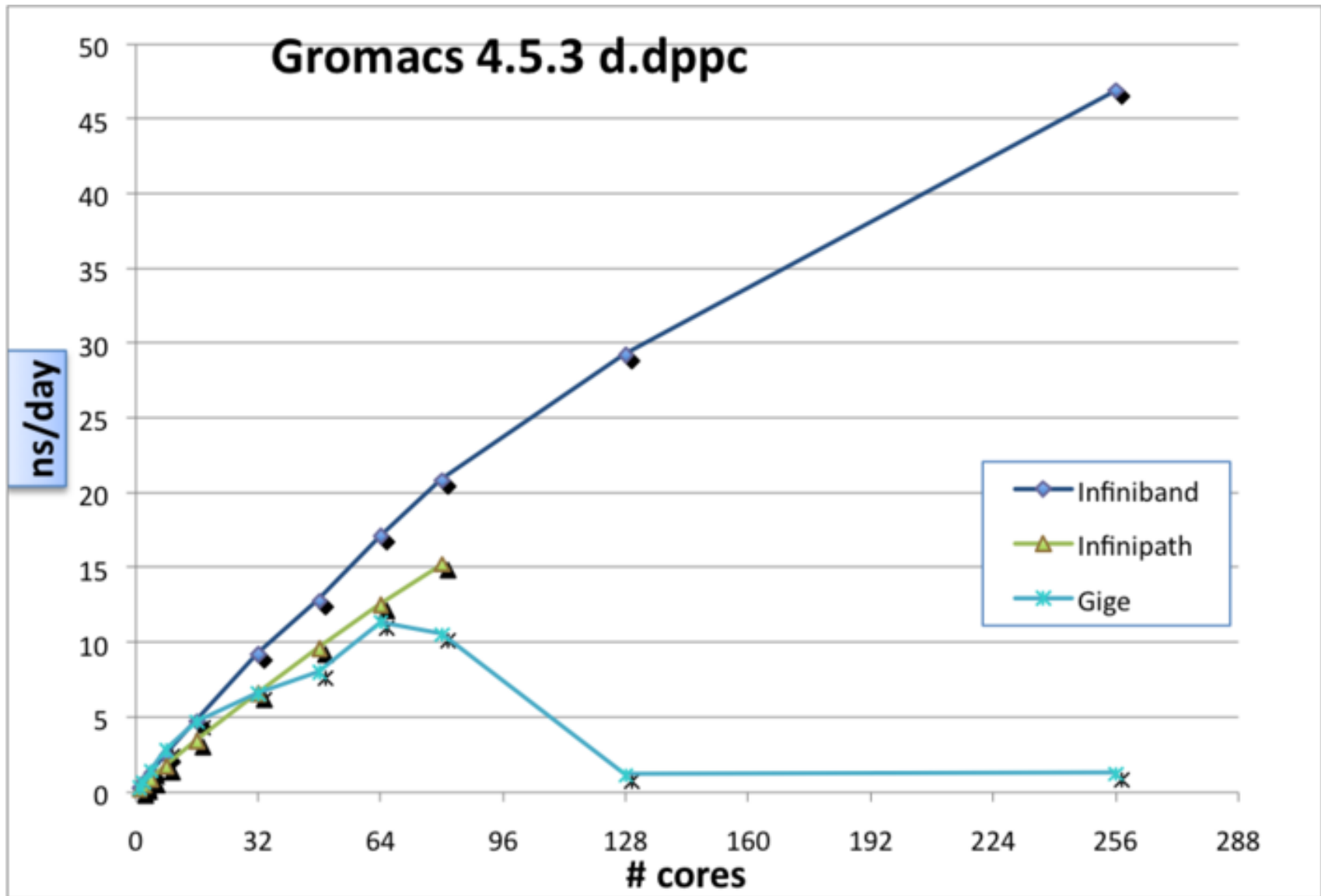
CCAD

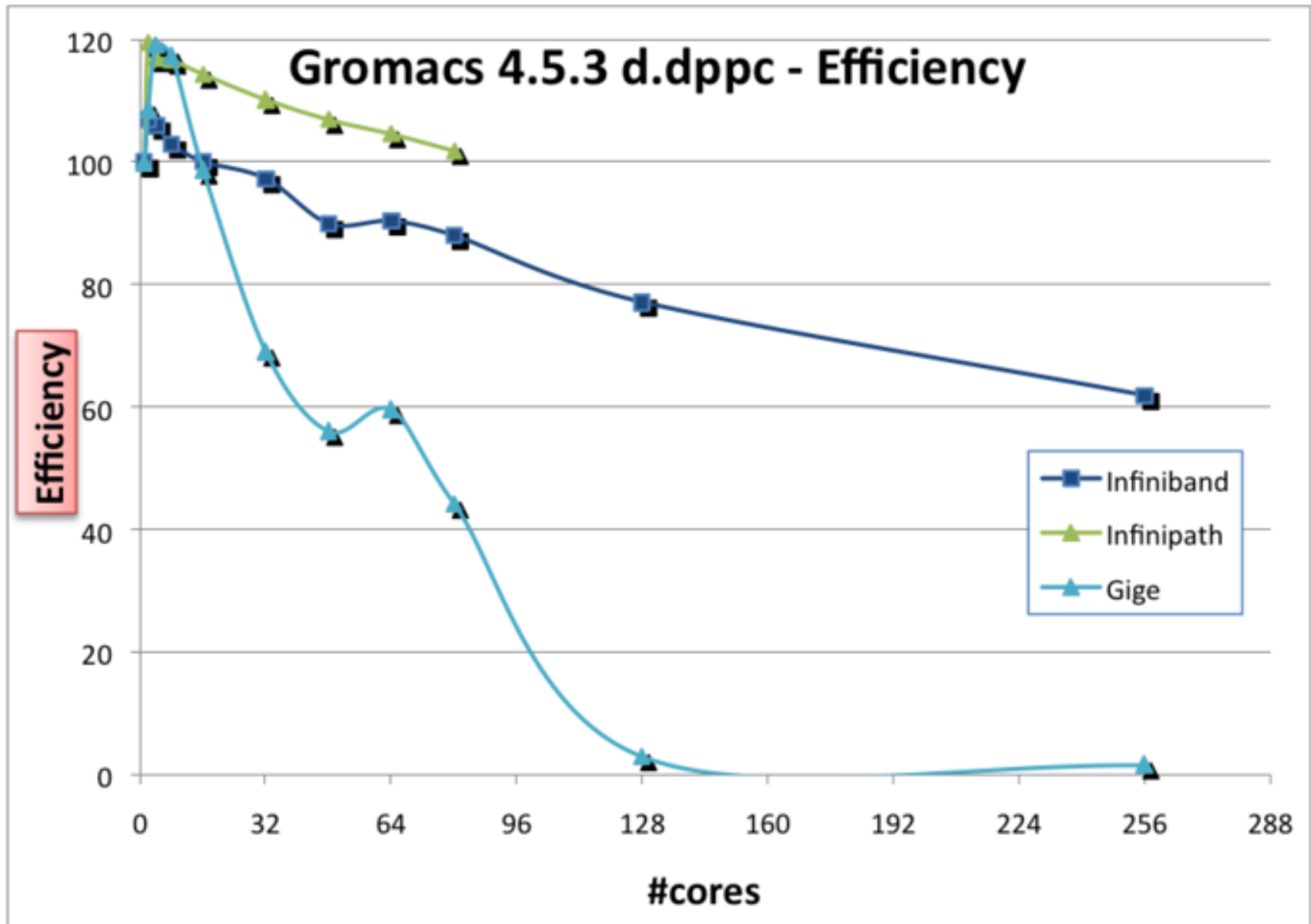
Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba





Ley de Amdahl

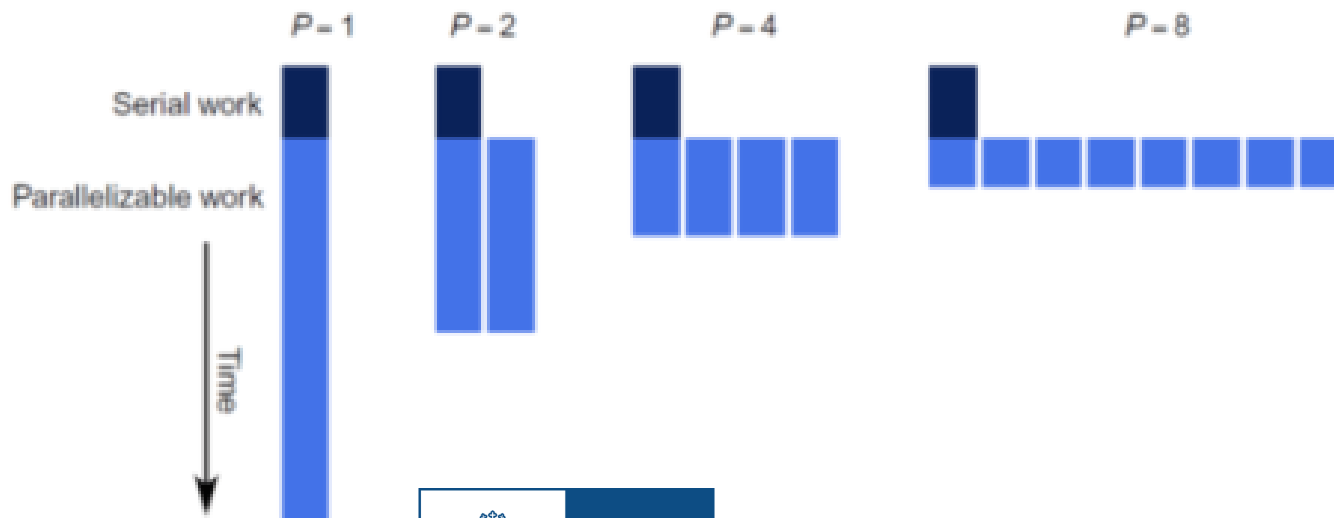
...the effort expended on achieving high parallel processing rates is wasted unless it is accompanied by achievements in sequential processing rates of very nearly the same magnitude.

– Gene Amdahl

El speedup máximo obtenible agregando procesadores está acotado por la porción secuencial del código.

$$T_N = T_1 \left(s + \frac{p}{N} \right)$$

$$S_N = \frac{1}{s + \frac{p}{N}}$$



CCAD

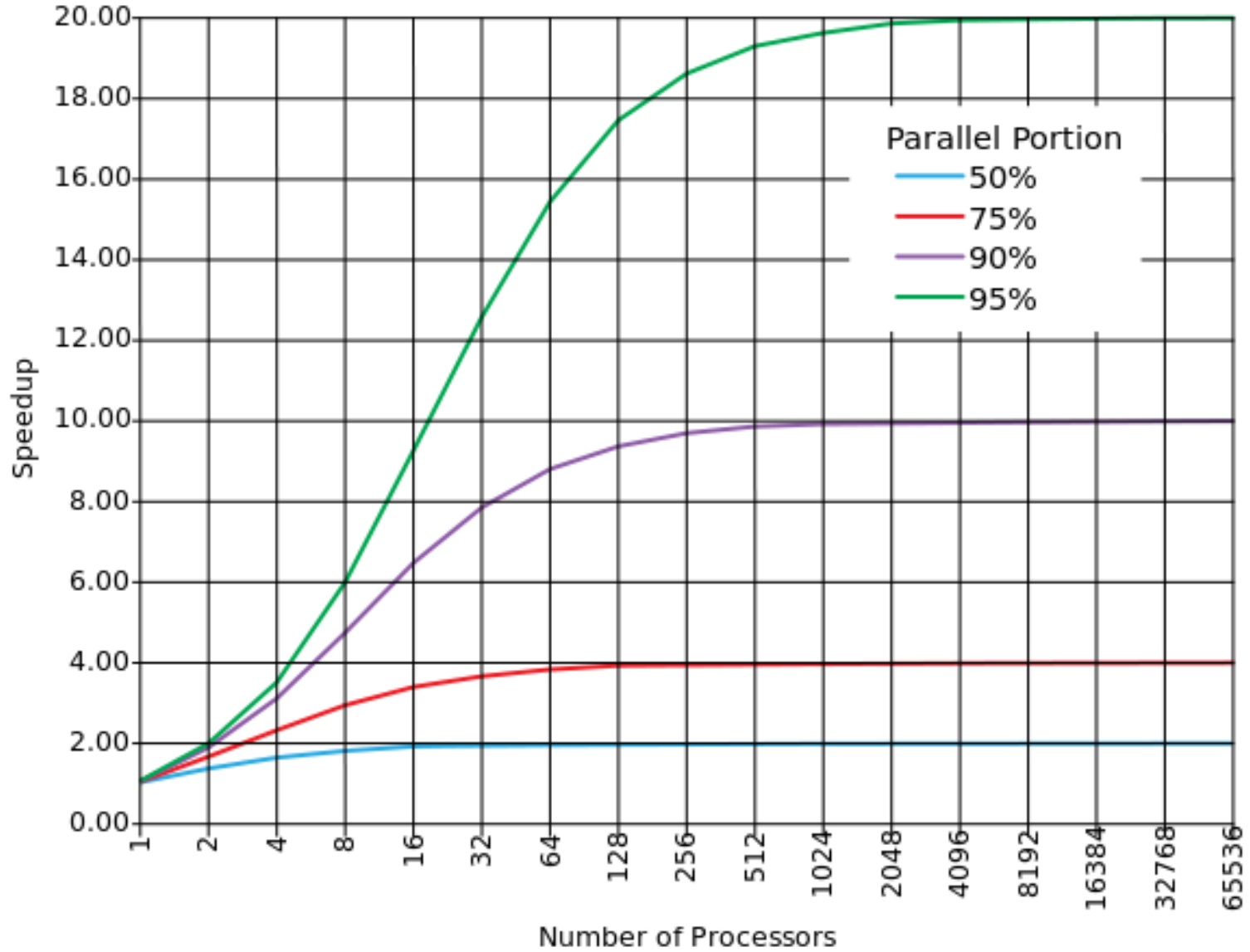
Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Amdahl's Law



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Ley de Gustafson

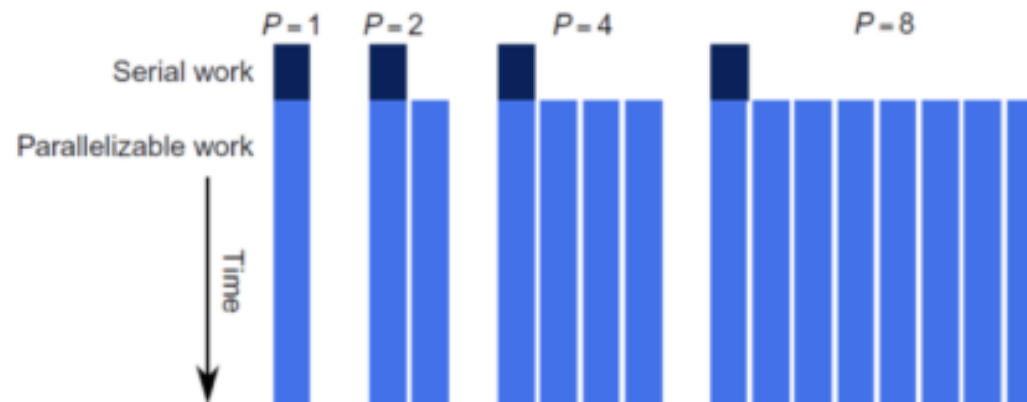
Observación: Generalmente la parte secuencial es independiente del tamaño del problema.

...speedup should be measured by scaling the problem to the number of processors, not by fixing the problem size.

– John Gustafson

Workaround: Si no escala, agrandar el problema.

$$S_N = N - s(N - 1)$$



Scaling

Strong scaling: Escalabilidad de un problema de tamaño fijo respecto a la cantidad de procesadores.

Weak scaling: Escalabilidad de un problema de **tamaño fijo por procesador** respecto a la cantidad de procesadores.



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Profiling en general

Técnicas para obtener información sobre la ejecución de un programa

- Generalmente métricas con el objetivo de optimizar cuellos de botella
- Dos tipos generales según cómo obtienen los datos
 - Los que instrumentan el código para hacer las mediciones necesarias (e.g. gcc -pg, luego gprof o gcov)
 - Los que observan el código en ejecución sin requerir modificaciones (e.g. perf, VTune...)



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Sampling

Un procesador tiene un conjunto de contadores configurables para llevar rastro de una variedad de eventos

- cpuid los lista

```
number of fixed counters    = 0x3 (3)
number of counters per logical processor = 0x8 (8)
```

Se configura un contador que al hacer overflow genera una interrupción y en cada interrupción el kernel anota dónde ocurrió el evento

- Pueden ser ciclos (tiempo) o eventos del procesador
- Default 1 kHz, máximo `kernel.perf_event_max_sample_rate` (100 kHz)

Limitaciones:

- Teorema de muestreo
- A muy alta frecuencia no queda tiempo para trabajo útil



Counting

Se configuran los mismos contadores que para sampling, pero en lugar de parar cada tanto se reportan las cantidades al final

Pros:

- No se pierden datos
 - ...hasta que pedimos más métricas que los contadores disponibles

Cons:

- Difícil de acertarle a la etapa que queremos en programas complejos



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Tracing

A veces la muestra no alcanza, necesitamos contexto para ver de dónde venimos.

Soluciones:

- Capturar más información al tomar la muestra
- Guardar toda la información siempre

Cosas interesantes que trazar:

- Call stack
- Last Branch Record (acelerado en hardware)

Limitaciones:

- Tamaño
- Overhead

Sanity check: htop

Antes de hacer nada, veamos que todo esté en orden.

htop es un top alternativo, features similares

- Utilización desglosada de cada core
- Utilización de memoria
- Lista de procesos en ejecución

Cosas para revisar:

- Un proceso o hilo por core o hilo del procesador
- No se pelea por tiempo de procesador con otros programas.



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

perf

Herramienta de profiling asociada a la system call `perf_event_open` del kernel de Linux

- No requiere instrumentación
- Bajo overhead
- Hace sampling, counting, tracing

Permite observar:

- La ejecución entera de un programa (cuando se pasa el comando)
- Un proceso en ejecución pasando el PID
- **Todo** lo que ocurre en todo el sistema (-a)



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Primeras mediciones: perf stat

Counting profiler. Muestra un resumen de métricas de ejecución de **todo un programa**.

- Métricas por defecto
 - Tiempo en espacio de usuario y en kernel (como `time` comando)
 - Utilización de CPUs
 - Ciclos, instrucciones, IPC
 - Branches, predicciones de branches erradas
 - (-ddd) Loads de distintos niveles de cache y sus misses
- Se puede configurar a mano cualquier conjunto de eventos (-e) y métricas basadas en eventos (-M) obtenidos con `perf list`
- Top-down profiling con `--topdown`



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

perf list

Lista los eventos y métricas que conoce perf

- Hay muchos más y con mucho más detalle de lo que generalmente necesitamos
- Y aún así no son todos
 - Herramienta con lista extendida para procesadores Intel: `ocperf`
- "Metric groups" (al final) provee algunas métricas útiles ya procesadas



CCAD

Centro de
Computación
de Alto
Desempeño

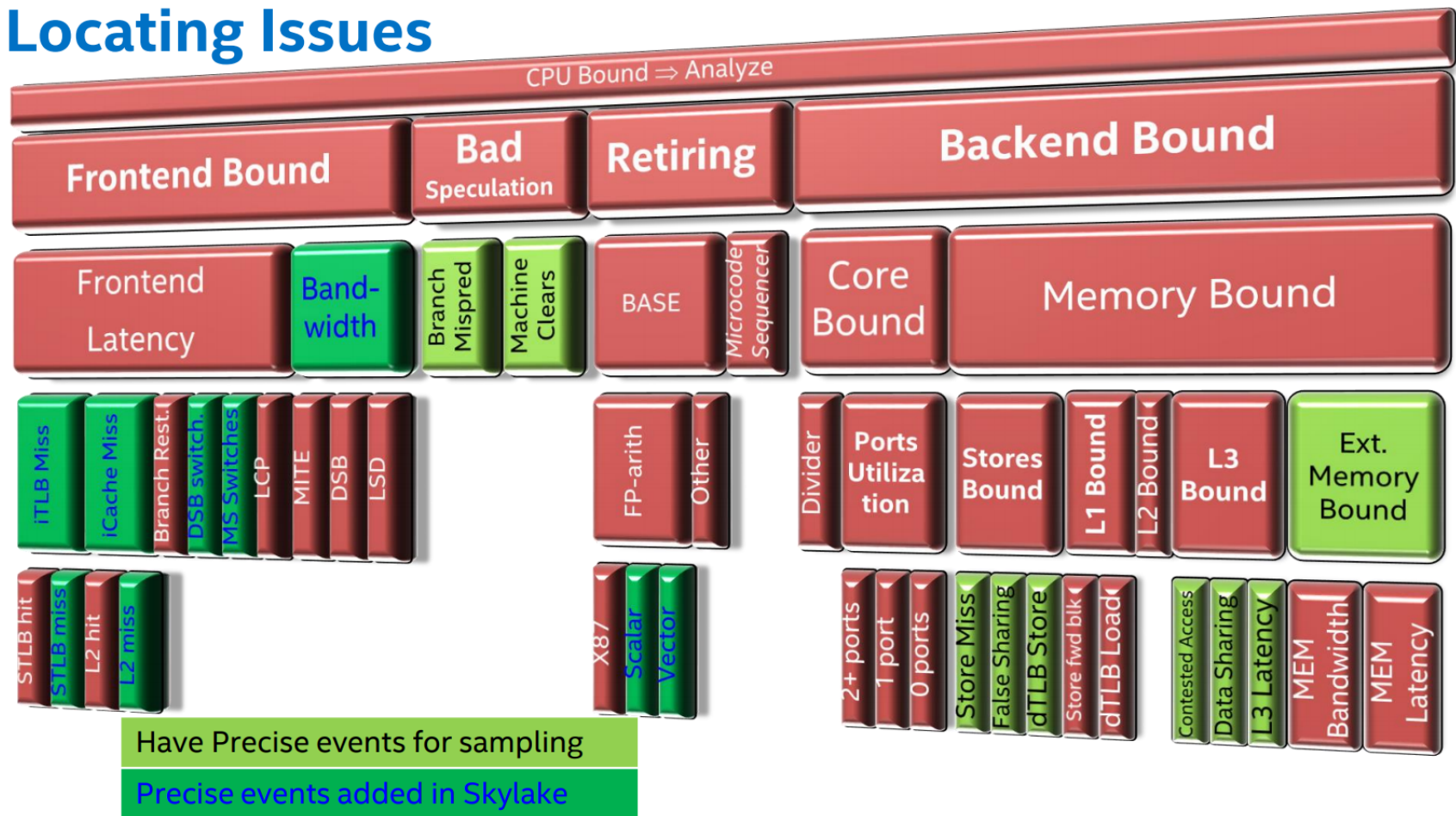


UNC

Universidad
Nacional
de Córdoba

Top-down profiling

Locating Issues



Ahmad Yasin, Top-down Microarchitecture Analysis through Linux perf and toplev tools



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Explorando: perf top

Sampling de tiempo *todo el sistema* con información en tiempo real

- Interfaz de texto
- Reporte de utilización de CPU por función
- Opción para hacer zoom en una función y ver utilización por instrucción
 - Si hay información de debugging, se muestra el código fuente asociado a cada instrucción también
- Incluye utilización del kernel, que puede ayudar para problemas más complicados con system calls



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Apartado: Symbol table

El sampling de perf crea un histograma de ocurrencias de eventos por instrucción. Para sacar algo útil de esta información, necesitamos tener un mapeo entre el binario en ejecución y el código fuente. (la tabla de símbolos)

- Viene con la información de debugging (compilar con `-g`)

A tener en cuenta:

- Algunos eventos demoran algunas instrucciones en notificarse y se le cuentan a la instrucción equivocada
- Las optimizaciones más agresivas vistas ayer (inlining, LTO, transformaciones de loops) destruyen la relación 1:1 entre binario y código
 - Aflojarle un poco a las optimizaciones (`-O2`) para ubicarse



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Dump de métricas: perf record

Sampling profiler. Se configuran las métricas a observar y se inicia el profiling. perf vuelca los datos a perf.data para posterior análisis.

Análisis offline: perf report

Interfaz similar a perf top, procesa y muestra lo guardado en perf.data.



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

perf c2c

Sampling profiler de propósito específico. Sólo cuenta líneas de cache que han sido encontradas modificadas en un lugar remoto (HITM)

- Para detectar contención sobre líneas de cache, en particular false sharing (problema común por el que OpenMP no paraleliza aunque parezca trivial)



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Intel Advisor

Profiler para asistir con vectorización y paralelización de código.

Features interesantes:

- Roofline analysis por función
- Análisis muy específico para generar sugerencias de optimizaciones

Para utilizar:

1. `source (...)/intel/advisor/advixe-vars.sh`
2. Correr el profiler
 - `advixe-gui` para la interfaz gráfica
 - `advixe-cl` para el programa de línea de comandos (para poner en jobs)



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Intel VTune Amplifier

Profiler de propósito general con interfaz gráfica amigable

- Hotspots: Sampling profiler similar a `perf record + perf report`
- Microarchitecture exploration: *top-down metrics por función*

Para utilizar:

1. `source (...) /intel/vtune_amplifier/amplxe-vars.sh`
2. Correr el profiler
 - `amplxe-gui` para la interfaz gráfica
 - `amplxe-cl` para el programa de línea de comandos (para poner en jobs)



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

Profiling en Python:

- cProfile
 - Profiler de funciones
 - `python -m cProfile script.py`
- line_profiler
 - Decorar funciones a medir con `@profile`
 - Correr el script con `kernprof`



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba

¿Y ahora?

Paralelizar!

- Usuarios Fortran/C/C++: Aprender OpenMP *moderno*
 - `omp parallel`, `task`, `simd...`
- Usuarios CUDA, C/C++ también: Aprender SYCL
- Todes: Aprender MPI



CCAD

Centro de
Computación
de Alto
Desempeño



UNC

Universidad
Nacional
de Córdoba