

Análisis y Experimentación Científicos

A.K.A. Cómo no hacer un informe del TP2

Organización del Computador II
Javier Pimás

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Slides: J. Pimás, D. González M.

17-04-2018

Agenda

1. Motivación

- 1.1 Objetivos
- 1.2 Requerimientos
- 1.3 Hipótesis

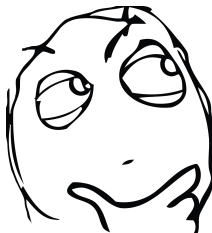
2. Armado de experimentos

- 2.1 Cómo observar performance
- 2.2 Cómo reflexionar sobre los resultados

3. Presentación de los resultados

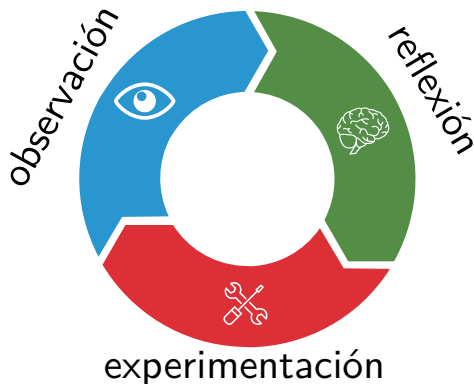
- 3.1 ¿Qué muestran los gráficos?
- 3.2 Ejemplos

¿Ciencias? de la Computación

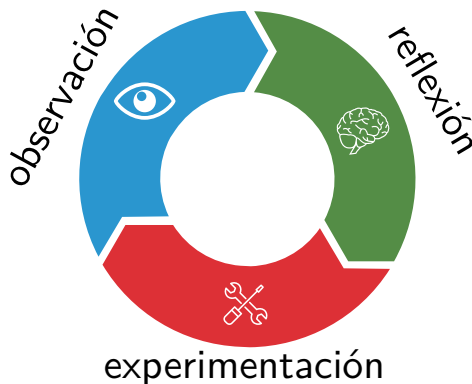


¿Que queremos decir con ciencia?

El ciclo de la investigación



El ciclo de la investigación



De sus informes esperamos conocer sus recorridos alrededor de varias vueltas a este círculo

Un estudio científico debe ser...

- 1 Objetivo
- 2 Reproducible
- 3 Coherente
- 4 Entendible por pares
- 5 Riguroso

Un estudio científico debe ser...

- ① Objetivo
- ② Reproducible
- ③ Coherente
- ④ Entendible por pares
- ⑤ Riguroso
- ⑥ etc
- ⑦ etc
- ⑧ etc



¿Cuáles son nuestras hipótesis?

- Una hipótesis es una **afirmación falseable**, que suponemos cierta.
- Como científicos nos dedicamos a formular hipótesis y validarlas
- No ejemplos:
 - SIMD es mejor que C ...

¿Cuáles son nuestras hipótesis?

- Una hipótesis es una **afirmación falseable**, que suponemos cierta.
- Como científicos nos dedicamos a formular hipótesis y validarlas
- No ejemplos:
 - SIMD es mejor que C ... **es una opinión, no una hipótesis**

¿Cuáles son nuestras hipótesis?

- Una hipótesis es una **afirmación falseable**, que suponemos cierta.
- Como científicos nos dedicamos a formular hipótesis y validarlas
- No ejemplos:
 - SIMD es mejor que C ... **es una opinión, no una hipótesis**
 - SIMD es más rápido que C ...

¿Cuáles son nuestras hipótesis?

- Una hipótesis es una **afirmación falseable**, que suponemos cierta.
- Como científicos nos dedicamos a formular hipótesis y validarlas
- ~~No~~ ejemplos:
 - SIMD es mejor que C ... **es una opinión, no una hipótesis**
 - SIMD es más rápido que C ... **un poco ambigua, pero OK por ahora (es una hipótesis, aun no sabemos si es cierta)**

¿Cuáles son nuestras hipótesis?

Una hipótesis **útil** tiene implicancias interesantes:

¿Cuáles son nuestras hipótesis?

Una hipótesis **útil** tiene implicancias interesantes:

- si SIMD es más rápido que C, **entonces** mi programa con SIMD correrá más rápido que mi programa en C.

¿Cuáles son nuestras hipótesis?

Una hipótesis **útil** tiene implicancias interesantes:

- si SIMD es más rápido que C, **entonces** mi programa con SIMD correrá más rápido que mi programa en C. *Si $p \rightarrow q$.*

¿Cuáles son nuestras hipótesis?

Una hipótesis **útil** tiene implicancias interesantes:

- si SIMD es más rápido que C, **entonces** mi programa con SIMD correrá más rápido que mi programa en C. Si $p \rightarrow q$.
- Luego, si $\neg q \rightarrow \neg p$.

¿Cuáles son nuestras hipótesis?

Una hipótesis **útil** tiene implicancias interesantes:

- si SIMD es más rápido que C, **entonces** mi programa con SIMD correrá más rápido que mi programa en C. Si $p \rightarrow q$.
- Luego, si $\neg q \rightarrow \neg p$.
- Si mi programa con SIMD no es más rápido que mi programa en C, entonces "SIMD es más rápido que C" no es cierto.

¿Cuáles son nuestras hipótesis?

Una hipótesis **útil** tiene implicancias interesantes:

- si SIMD es más rápido que C, **entonces** mi programa con SIMD correrá más rápido que mi programa en C. Si $p \rightarrow q$.
- Luego, si $\neg q \rightarrow \neg p$.
- Si mi programa con SIMD no es más rápido que mi programa en C, entonces "SIMD es más rápido que C" no es cierto.
- Por definición una hipótesis es algo no demostrado, y tan sólo podemos asumirla cierta **momentáneamente** para verificar que sus implicancias también lo son.

Refinando las hipótesis

“SIMD es más rápido que C”

Refinando las hipótesis

“SIMD es más rápido que C”

- **Problema:** demasiado ambiguo, ¿se cumple **siempre**?

Refinando las hipótesis

“SIMD es más rápido que C”

- **Problema:** demasiado ambiguo, ¿se cumple **siempre**?
- Si fuera cierto entonces, con cualquier programa, en cualquier plataforma, SIMD es más rápido que C

Refinando las hipótesis

“SIMD es más rápido que C”

- **Problema:** demasiado ambiguo, ¿se cumple **siempre**?
- Si fuera cierto entonces, con cualquier programa, en cualquier plataforma, SIMD es más rápido que C
- Entonces ... **es FALSO** (no me crean, piensen contraejemplos)

Refinando las hipótesis

“Nuestra implementación con SIMD es más rápida que la de C”

Refinando las hipótesis

“Nuestra implementación con SIMD es más rápida que la de C”

- **Problema:** demasiado genérico, ¿se puede **demostrar**?

Refinando las hipótesis

“Nuestra implementación con SIMD es más rápida que la de C”

- **Problema:** demasiado genérico, ¿se puede **demostrar**?
- Si fuera cierto entonces, con todo compilador, para todo nivel de optimización, en todas las plataformas ... **impracticable**.

Refinando las hipótesis

“Nuestra implementación con SIMD es más rápida que la de C”

- **Problema:** demasiado genérico, ¿se puede **demostrar**?
- Si fuera cierto entonces, con todo compilador, para todo nivel de optimización, en todas las plataformas ... **impracticable**.
- Moraleja... **el investigador es esclavo de sus hipótesis**

Refinando las hipótesis

“Nuestra implementación con SIMD puede ser hasta 80 % más rápida que la de C compilado con GCC v5.1.1 -O3”

Refinando las hipótesis

“Nuestra implementación con SIMD puede ser hasta 80 % más rápida que la de C compilado con GCC v5.1.1 -O3”

- ¿Se cumple **siempre**?

Refinando las hipótesis

“Nuestra implementación con SIMD puede ser hasta 80 % más rápida que la de C compilado con GCC v5.1.1 -O3”

- ¿Se cumple **siempre**? Alcanza con mostrar 1 caso.

Refinando las hipótesis

“Nuestra implementación con SIMD puede ser hasta 80 % más rápida que la de C compilado con GCC v5.1.1 -O3”

- ¿Se cumple **siempre**? Alcanza con mostrar 1 caso. Quizás existan corridas que no sean 80 % más rápidas, pero nuestra hipótesis no afirma que todas lo sean.

Refinando las hipótesis

“Nuestra implementación con SIMD puede ser hasta 80 % más rápida que la de C compilado con GCC v5.1.1 -O3”

- ¿Se cumple **siempre**? Alcanza con mostrar 1 caso. Quizás existan corridas que no sean 80 % más rápidas, pero nuestra hipótesis no afirma que todas lo sean.
- ¿Es un indicio de algo esperable y repetible?

Validación de las hipótesis

“Nuestra implementación con SIMD puede ser hasta 80 % más rápida que la de C compilado con GCC v5.1.1 -O3”

Validación de las hipótesis

“Nuestra implementación con SIMD puede ser hasta 80 % más rápida que la de C compilado con GCC v5.1.1 -O3”

- Con afirmaciones empíricas, las verdades demostradas suelen ser “débiles” .

Validación de las hipótesis

“Nuestra implementación con SIMD puede ser hasta 80 % más rápida que la de C compilado con GCC v5.1.1 -O3”

- Con afirmaciones empíricas, las verdades demostradas suelen ser “débiles” .
- La fortaleza de la hipótesis suele estar más asociada a qué tan reproducible es su afirmación.

Validación de las hipótesis

“Nuestra implementación con SIMD puede ser hasta 80 % más rápida que la de C compilado con GCC v5.1.1 -O3”

- Con afirmaciones empíricas, las verdades demostradas suelen ser “débiles” .
- La fortaleza de la hipótesis suele estar más asociada a qué tan reproducible es su afirmación.
- ¿Es la hipótesis un indicio de algo esperable y repetible?

Validación de las hipótesis

“Nuestra implementación con SIMD puede ser hasta 80 % más rápida que la de C compilado con GCC v5.1.1 -O3”

- Con afirmaciones empíricas, las verdades demostradas suelen ser “débiles” .
- La fortaleza de la hipótesis suele estar más asociada a qué tan reproducible es su afirmación.
- ¿Es la hipótesis un indicio de algo esperable y repetible?
- Para saberlo, repetiremos nuestros experimentos muchas veces.

Resultados, hipótesis y “demostraciones”

- A diferencia de una hipótesis, un **resultado** es sólo una muestra. **No es verdadero ni falso, tan sólo un hecho observado.**

Resultados, hipótesis y “demostraciones”

- A diferencia de una hipótesis, un **resultado** es sólo una muestra. **No es verdadero ni falso, tan sólo un hecho observado.**
- Nuestro trabajo más difícil consistirá en encontrar hipótesis que puedan ser validadas mediante los resultados de las mediciones.

Resultados, hipótesis y “demostraciones”

- A diferencia de una hipótesis, un **resultado** es sólo una muestra. **No es verdadero ni falso, tan sólo un hecho observado.**
- Nuestro trabajo más difícil consistirá en encontrar hipótesis que puedan ser validadas mediante los resultados de las mediciones.
- Además, buscaremos hipótesis **útiles**, cuyos resultados sean **estadísticamente significativos**.

Resultados, hipótesis y “demostraciones”

- A diferencia de una hipótesis, un **resultado** es sólo una muestra. **No es verdadero ni falso, tan sólo un hecho observado.**
- Nuestro trabajo más difícil consistirá en encontrar hipótesis que puedan ser validadas mediante los resultados de las mediciones.
- Además, buscaremos hipótesis **útiles**, cuyos resultados sean **estadísticamente significativos**.
- Finalmente, nuestros experimentos deben ser **sistemáticos**, para evitar que se introduzcan errores.

Objetivos prácticos



¿Como medimos “performance”?

¿Qué medir?

- **Tiempo** → (numerito seguido de su unidad)

Podemos medir en unidades de tiempo, como microsegundos.

El problema es que para eventos que suceden muy rápidamente la precisión de los relojes es insuficiente.

¿Qué medir?

- **Tiempo** → (numerito seguido de su unidad)

Podemos medir en unidades de tiempo, como microsegundos.

El problema es que para eventos que suceden muy rápidamente la precisión de los relojes es insuficiente.

- **Ticks de Reloj** → (numerito sin unidad)

Para obtener el contador de ticks en Intel utilizamos la instrucción `rdtsc`.

¿Qué medir?

- **Tiempo** → (numerito seguido de su unidad)

Podemos medir en unidades de tiempo, como microsegundos.

El problema es que para eventos que suceden muy rápidamente la precisión de los relojes es insuficiente.

- **Ticks de Reloj** → (numerito sin unidad)

Para obtener el contador de ticks en Intel utilizamos la instrucción `rdtsc`.

¿Qué puede afectar la medición?

- Ruido del sistema

Nuestra aplicación **no corre sola** en el sistema, la interacción con otras aplicaciones genera ruido.

¿Qué puede afectar la medición?

- Ruido del sistema

Nuestra aplicación **no corre sola** en el sistema, la interacción con otras aplicaciones genera ruido.

- Datos de entrada

Puede existir alguna **característica especial** de nuestros datos de entrada que generen una medición no esperada.

¿Qué puede afectar la medición?

- Ruido del sistema

Nuestra aplicación **no corre sola** en el sistema, la interacción con otras aplicaciones genera ruido.

- Datos de entrada

Puede existir alguna **característica especial** de nuestros datos de entrada que generen una medición no esperada.

- Sistema Operativo

El sistema operativo y su accionar **afecta** nuestra medición. Debemos controlarlo.

¿Qué puede afectar la medición?

- Ruido del sistema

Nuestra aplicación **no corre sola** en el sistema, la interacción con otras aplicaciones genera ruido.

- Datos de entrada

Puede existir alguna **característica especial** de nuestros datos de entrada que generen una medición no esperada.

- Sistema Operativo

El sistema operativo y su accionar **afecta** nuestra medición. Debemos controlarlo.

- El propio procesador

Los **mecanismos de optimización** del rendimiento de un procesador afectan la medición, aumentando y disminuyendo la frecuencia de trabajo de las distintas partes.

¿Cómo evitar outliers?

Sean las siguientes muestras de datos,

> a = c(32, 55, 32, 54, 65, 32, 33, 54, 78, 2093486723)

> b = c(32, 55, 32, 54, 65, 32, 33, 54, 78)

¿Cómo evitar outliers?

Sean las siguientes muestras de datos,

```
> a = c(32, 55, 32, 54, 65, 32, 33, 54, 78, 2093486723)
```

```
> b = c(32, 55, 32, 54, 65, 32, 33, 54, 78)
```

```
> mean(a)
```

209348716

```
> mean(b)
```

48.33333

```
> median(a)
```

54

```
> sd(a)
```

662018614

```
> sd(b)
```

16.9632

¿Cómo evitar outliers?

Sean las siguientes muestras de datos,

> a = c(32, 55, 32, 54, 65, 32, 33, 54, 78, 2093486723)

> b = c(32, 55, 32, 54, 65, 32, 33, 54, 78)

> mean(a)

209348716

> mean(b)

48.33333

> median(a)

54

> sd(a)

662018614

> sd(b)

16.9632

Se debe poder evitar y controlar la aparición de outliers. En el caso de tener outliers, se debe poder clasificarlos y cuantificarlos en la muestra.

Debemos armar un protocolo para realizar nuestras mediciones

¿Qué valores se espera obtener?

- **Tiempo** Un número que represente un tiempo, si es muy grande entonces serán segundos, horas, días. Si es muy chico serán microsegundos, nanosegundos.
¿Es razonable demorar 3 segundos para ejecutar 1000 instrucciones?

¿Qué valores se espera obtener?

- **Tiempo** Un número que represente un tiempo, si es muy grande entonces serán segundos, horas, días. Si es muy chico serán microsegundos, nanosegundos.
¿Es razonable demorar 3 segundos para ejecutar 1000 instrucciones?
- **Rendimiento** Denota la diferencia de rendimiento entre dos implementaciones. Se expresa como porcentajes.
120 %, puede significar que A es 20 % más eficiente en tiempo que B.
120 %, puede significar que A demora un 20 % más que B.
10 %, puede significar que A demora el 10 % de B.

¿Qué valores se espera obtener?

- **Tiempo** Un número que represente un tiempo, si es muy grande entonces serán segundos, horas, días. Si es muy chico serán microsegundos, nanosegundos.
¿Es razonable demorar 3 segundos para ejecutar 1000 instrucciones?
- **Rendimiento** Denota la diferencia de rendimiento entre dos implementaciones. Se expresa como porcentajes.
120 %, puede significar que A es 20 % más eficiente en tiempo que B.
120 %, puede significar que A demora un 20 % más que B.
10 %, puede significar que A demora el 10 % de B.
- **Ticks de Reloj** Son valores enteros muy grandes. Nos va a interesar la relación entre estos.
Dos implementaciones que demoran 190231359147543 y 192125445767335 ticks, tienen similar rendimiento.

¿Qué analizar? (1)

Debemos entender qué vamos a analizar,

- **Cuánto demora una implementación:**

Obtener una medida de tiempo para relacionarla en un contexto

¿Qué analizar? (1)

Debemos entender qué vamos a analizar,

- **Cuánto demora una implementación:**

Obtener una medida de tiempo para relacionarla en un contexto

- **Comparar dos implementaciones:**

Obtener la diferencia de tiempos o de porcentaje relativo entre dos implementaciones

¿Qué analizar? (1)

Debemos entender qué vamos a analizar,

- **Cuánto demora una implementación:**

Obtener una medida de tiempo para relacionarla en un contexto

- **Comparar dos implementaciones:**

Obtener la diferencia de tiempos o de porcentaje relativo entre dos implementaciones

- **Comparar el uso de distintas instrucciones:**

Obtener una medida de mejora con respecto a utilizar un determinado conjunto de instrucciones con respecto a otro

¿Qué analizar? (2)

Debemos entender qué vamos a analizar,

- **Medir el rendimiento de una implementación:**

Obtener el porcentaje de mejora de una implementación con respecto a una implementación patrón

¿Qué analizar? (2)

Debemos entender qué vamos a analizar,

- **Medir el rendimiento de una implementación:**

Obtener el porcentaje de mejora de una implementación con respecto a una implementación patrón

- **Factor limitante en una implementación:**

Obtener una medida relativa de rendimiento con respecto a forzar un factor (ej. memoria, saltos)

¿Qué analizar? (2)

Debemos entender qué vamos a analizar,

- **Medir el rendimiento de una implementación:**
Obtener el porcentaje de mejora de una implementación con respecto a una implementación patrón
- **Factor limitante en una implementación:**
Obtener una medida relativa de rendimiento con respecto a forzar un factor (ej. memoria, saltos)
- **Análisis del comportamiento de una implementación:**
Rendimiento bajo distinto conjunto de parámetros de entrada.

¿Qué resultado se espera?

- **A mantiene** su rendimiento cuando se modifican los parámetros de entrada.

¿Qué resultado se espera?

- **A mantiene** su rendimiento cuando se modifican los parámetros de entrada.
- **A varía** su rendimiento cuando se modifican los parámetros de entrada.

¿Qué resultado se espera?

- **A mantiene** su rendimiento cuando se modifican los parámetros de entrada.
- **A varía** su rendimiento cuando se modifican los parámetros de entrada.
 - Variación correlacionada con una función, predecible.
 - No predecible, relacionada con factores externos a los parámetros de entrada.

¿Qué resultado se espera?

- **A mantiene** su rendimiento cuando se modifican los parámetros de entrada.
- **A varía** su rendimiento cuando se modifican los parámetros de entrada.
 - Variación correlacionada con una función, predecible.
 - No predecible, relacionada con factores externos a los parámetros de entrada.
- **A es más rápido** que **B**
 - en tiempo
 - en porcentaje

¿Qué resultado se espera?

- A **mantiene** su rendimiento cuando se modifican los parámetros de entrada.
- A **varía** su rendimiento cuando se modifican los parámetros de entrada.
 - Variación correlacionada con una función, predecible.
 - No predecible, relacionada con factores externos a los parámetros de entrada.
- A es **más rápido** que B
 - en tiempo
 - en porcentaje
- A se **comporta igual** a B
 - diferencia no significativa

¿Qué resultado se espera?

- **A mantiene** su rendimiento cuando se modifican los parámetros de entrada.
- **A varía** su rendimiento cuando se modifican los parámetros de entrada.
 - Variación correlacionada con una función, predecible.
 - No predecible, relacionada con factores externos a los parámetros de entrada.
- **A es más rápido** que **B**
 - en tiempo
 - en porcentaje
- **A se comporta igual** a **B**
 - diferencia no significativa
- **A y B se comportan muy diferente**
 - analizar casos independientemente

¿Cómo armar datos de entrada?

Los datos de entrada se toman de casos reales o se generan artificialmente.

¿Cómo armar datos de entrada?

Los datos de entrada se toman de casos reales o se generan artificialmente.

Se crean,

- **Modificando una variable**

Ej. Alterando el valor de una componente de la simulación

- **Modificando un conjunto de variables bajo una regla**

Ej. Alterando el tamaño de la simulación en ancho y alto al mismo tiempo

¿Cómo armar datos de entrada?

Los datos de entrada se toman de casos reales o se generan artificialmente.

Se crean,

- **Modificando una variable**

Ej. Alterando el valor de una componente de la simulación

- **Modificando un conjunto de variables bajo una regla**

Ej. Alterando el tamaño de la simulación en ancho y alto al mismo tiempo

Tener en cuenta,

- Variables seleccionadas
- Cantidad total de entradas
- Entrada común o tipo

¿Cómo analizar resultados?

La etapa de discusión de los resultados, implica cuestionar las hipótesis sobre las que construimos nuestro experimento.

- ¿El resultado es esperado?
- ¿Existe algún factor que no estamos teniendo en consideración?
- ¿Es posible validar nuestro resultado con un experimento control?
- Si el resultado no es el esperado, ¿el experimento es incorrecto?
- Si los resultados son correctos, ¿hacer un nuevo experimento?

¿Qué se quiere mostrar?

Dependiendo de qué se quiera mostrar, se puede utilizar un tipo de gráfico u otro.

- **Rendimiento / Tiempo:** Area, Barras, Líneas
- **Relaciones:** Líneas, Dispersión
- **Porcentajes:** Torta, Area o Barras acumuladas
- **Comportamiento / Variación:** Líneas, Superficie

¿Qué colores? ¿Qué tipo de gráfico? ¿Qué escala?

Colores

- Gusto / estética
- Resaltar un dato sobre otro
- Pensar en el medio de distribución (impreso)

Tipo

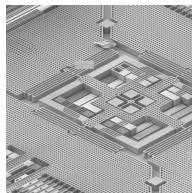
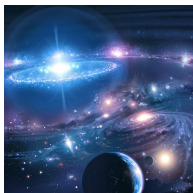
- El que mejor muestre la información (ojo con 3d)
- Que sirva para explicar (sin información de más)

Escala

- Lineal o Logarítmica
- Sobre qué eje va cada escala
- Valores en los ejes

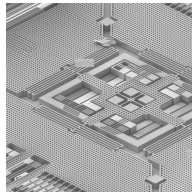
¿Qué vamos a presentar?

Valores muy chicos o muy grandes



¿Qué vamos a presentar?

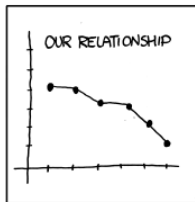
Valores muy chicos o muy grandes



Relación entre valores sin importar la magnitud

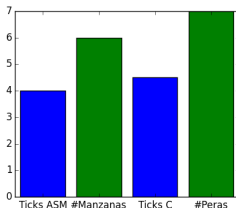


Humor de XKCD



Ejemplo 1: Comparando performance

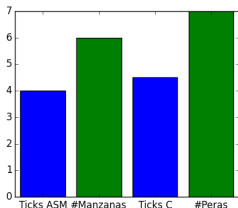
No podemos agrupar en un gráfico cosas que no tiene nada que ver entre ellas.



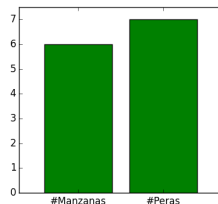
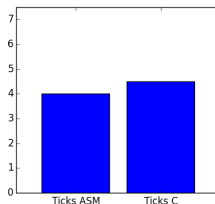
Ejemplo 1: Comparando performance

No podemos agrupar en un gráfico cosas que no tiene nada que ver entre ellas.

¡Malo!



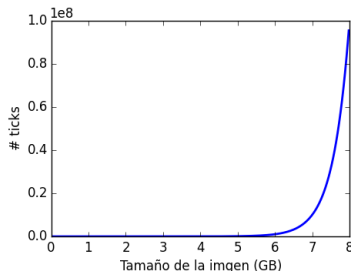
¡Bueno!



Tenemos q usar dos gráficos...

Ejemplo 2: Analizando límites/tendencias

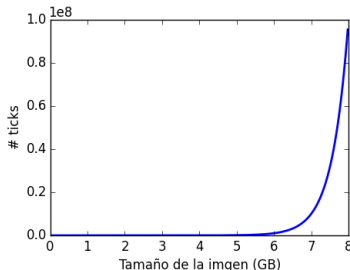
Escalas: Podemos setear la escala de cada eje con distintos criterios (lineal, log, loglog, etc) A veces, los datos lo requieren...



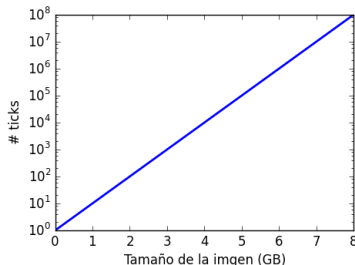
Ejemplo 2: Analizando límites/tendencias

Escalas: Podemos setear la escala de cada eje con distintos criterios (lineal, log, loglog, etc) A veces, los datos lo requieren...

¡Malo!



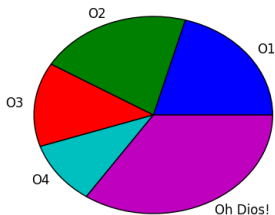
¡Bueno!



Cambiamos el eje x para que tenga escala logarítmica

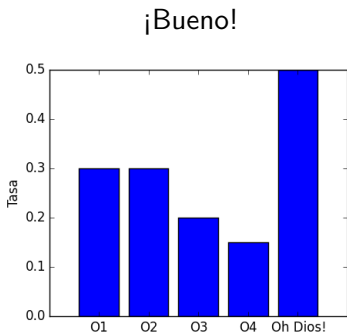
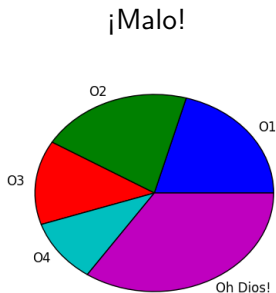
Ejemplo 3: Proporciones

Un grafico de torta sirve para ver proporciones en un TODO, no para agrupar valores que no constituyen un todo...



Ejemplo 3: Proporciones

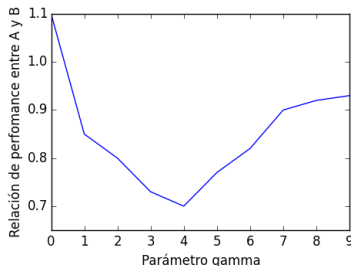
Un grafico de torta sirve para ver proporciones en un TODO, no para agrupar valores que no constituyen un todo...



Barras puede ser una opción

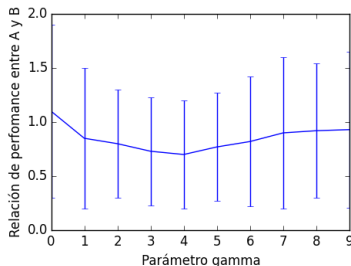
Ejemplo 4: Optimizar un parámetro

Tenemos que encontrar el parámetro γ óptimo.



Ejemplo 4: Optimizar un parámetro

Tenemos que encontrar el parámetro γ óptimo.

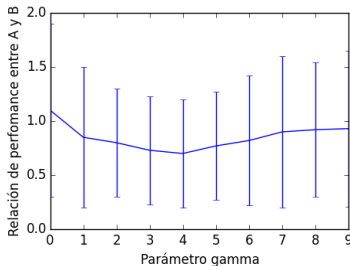


Utilizamos barras de error

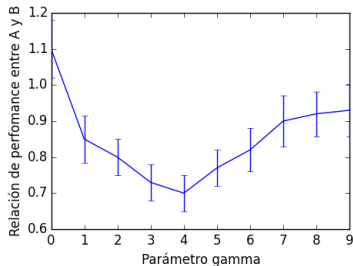
Ejemplo 4: Optimizar un parámetro

Tenemos que encontrar el parámetro γ óptimo.

¡Malo!



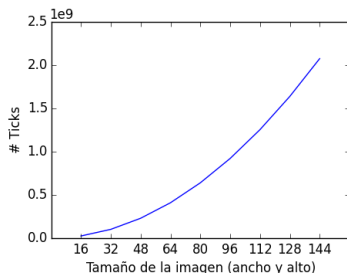
¡Bueno!



Utilizamos barras de error

Ejemplo 5: Analizando límites

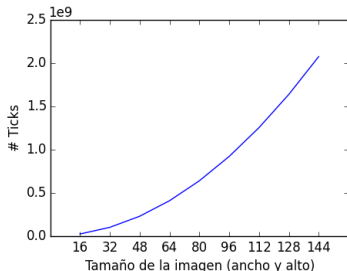
Queremos ver donde la performance varía.



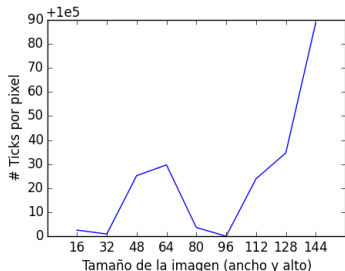
Ejemplo 5: Analizando límites

Queremos ver donde la performance varía.

¡Malo!



¡Bueno!



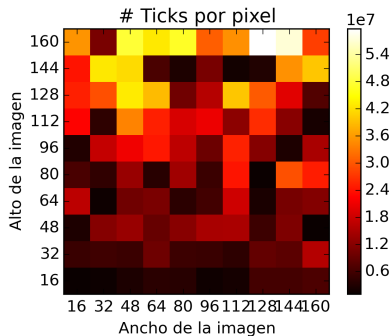
Normalizamos los datos

Ejemplo 5: Analizando límites en 2 variables

Queremos ver donde la performance varía pero separando ancho y alto.

Ejemplo 5: Analizando límites en 2 variables

Queremos ver donde la performance varía pero separando ancho y alto.



Usamos una escala de color para graficar valores

Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**

Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**
- Ser **ordenado en las explicaciones**, explicar tanto el código como los experimentos, datos y resultados

Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**
- Ser **ordenado en las explicaciones**, explicar tanto el código como los experimentos, datos y resultados
- **No adjuntar gráficos de más**, sólo deben ir los que hagan falta

Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**
- Ser **ordenado en las explicaciones**, explicar tanto el código como los experimentos, datos y resultados
- **No adjuntar gráficos de más**, sólo deben ir los que hagan falta
- El promedio de una muestra debe estar **acompañado de su desvío**

Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**
- Ser **ordenado en las explicaciones**, explicar tanto el código como los experimentos, datos y resultados
- **No adjuntar gráficos de más**, sólo deben ir los que hagan falta
- El promedio de una muestra debe estar **acompañado de su desvío**
- No todo se puede explicar, existen comportamientos **inexplicables**

Pequeños pero no menos importantes consejos

- Los resultados deben ser **reproducibles y consistentes**
- Ser **ordenado en las explicaciones**, explicar tanto el código como los experimentos, datos y resultados
- **No adjuntar gráficos de más**, sólo deben ir los que hagan falta
- El promedio de una muestra debe estar **acompañado de su desvío**
- No todo se puede explicar, existen comportamientos **inexplicables**
- El informe debe ser un **trabajo integral**, consistente y bien escrito

Herramientas

- `objdump` – para desensamblar archivos objeto
`$> objdump -D -Mintel archivo.o | grep -A 10 func`
- `run.sh` – script para correr guardando el output y el estado actual (hostname, git commit, power mode)
- `setcpufreq.sh` – script para fijar `cpufreq` y quitar turbo
- **Rstudio / Rodeo** – IDEs para escribir texto, graficar y hacer análisis dinámicamente.

Informe

- ❶ Implementación
 - Explicación general de la solución
 - Detalles de implementación
 - Uso de constantes y memoria
- ❷ Análisis preeliminar
 - Comparación de rendimiento de ASM vs C
 - Comparar para distintos tamaños, relaciones entre implementaciones
- ❸ Hipótesis de trabajo
 - Conjunto de ideas de experimentos
 - Afirmaciones que buscan probar verdaderas
 - Deben ser concisas y claras
- ❹ Diseño experimental
 - Explicación de como y que van a medir
 - Explicación del conjunto de datos de entrada
 - Detalles de la plataforma y la configuración de la misma
- ❺ Resultados y Análisis
 - Resultados obtenidos, gráficos y tablas
 - Explicación e interpretación de los resultados obtenidos
- ❻ Conclusiones
 - Relación entre las hipótesis de trabajo y resultados

¿Preguntas?