# Infraestructura para el Análisis de Rendimiento

Lic. Andrés More
Director: Dr Fernando G. Tinetti

Magíster en Cómputo de Altas Prestaciones
Universidad Nacional de La Plata
Facultad de Informática

Septiembre de 2016

# Contenido



# Resumen
## Resumen

- Las aplicaciones son construidas por especialistas en el dominio del problema, no expertos en rendimiento.
- Se desarrolló infraestructura que simplifica el análisis de rendimiento, permitiendo más tiempo de experimentación y análisis.
- El soporte consiste en información sobre programa, sistema, comportamiento, escalamiento, perfil de ejecución, cuellos de botella, uso de recursos.

# Introducción
## Motivación y Objetivos

- En HPC los desarrolladores son los especialistas del dominio, no son expertos en optimización.
- Menor tiempo de experimentación y análisis ya que el código optimizado puede ejecutarse órdenes de magnitud mejor que una implementación directa teórica.
- La metodología consiste en analizar el estado del arte, formular e implementar una solución sistemática, aplicar lo implementado a casos de estudio y documentar la experiencia.

# Introducción
## Contribuciones Realizadas

- **Reportes técnicos**: Estudio de Multiplicación de Matrices, Comparación de Implementaciones de una Operación BLAS.
- **Artículos**: Optimizing Latency in Beowulf Clusters, Lessons Learned from Contrasting BLAS Kernel Implementations, Hotspot: a Framework to Support Performance Optimization on Multiprocessors.
- **Libro**: Sección Intel Cluster Ready e Intel Cluster Checker en Programming Intel Xeon Phi. Reseña para JCS & T.

# Trabajo Relacionado
## Teoría Básica

- Rendimiento, Paralelismo y Métricas
- Leyes de Escalamiento (Amdahl y Gustafson)

| Número de Procesadores | Porcentaje de Paralelismo | | | | | |
|---|---|---|---|---|---|---|
| | 0.1 | 0.3 | 0.5 | 0.8 | 0.9 | 0.95 |
| 1 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 2 | 1.05 | 1.14 | 1.33 | 1.60 | 1.82 | 1.90 |
| 4 | 1.08 | 1.23 | 1.60 | 2.29 | 3.08 | 3.48 |
| 8 | 1.10 | 1.28 | 1.78 | 2.91 | 4.71 | 5.93 |
| 16 | 1.10 | 1.31 | 1.88 | 3.37 | 6.40 | 9.14 |
| 32 | 1.11 | 1.32 | 1.94 | 3.66 | 7.80 | 12.55 |
| 64 | 1.11 | 1.33 | 1.97 | 3.82 | 8.77 | 15.42 |
| 128 | 1.11 | 1.33 | 1.98 | 3.91 | 9.34 | 17.41 |
| 256 | 1.11 | 1.33 | 1.99 | 3.95 | 9.66 | 18.62 |
| 512 | 1.11 | 1.33 | 2.00 | 3.98 | 9.83 | 19.28 |
| 1024 | 1.11 | 1.33 | 2.00 | 3.99 | 9.91 | 19.64 |
| 2048 | 1.11 | 1.33 | 2.00 | 3.99 | 9.96 | 19.82 |
| 4096 | 1.11 | 1.33 | 2.00 | 4.00 | 9.98 | 19.91 |
| 8192 | 1.11 | 1.33 | 2.00 | 4.00 | 9.99 | 19.95 |
| 16384 | 1.11 | 1.33 | 2.00 | 4.00 | 9.99 | 19.98 |
| 32768 | 1.11 | 1.33 | 2.00 | 4.00 | 10.00 | 19.99 |
| 65536 | 1.11 | 1.33 | 2.00 | 4.00 | 10.00 | 19.99 |

| Número de Procesadores | Porcentaje de Paralelismo | | | | | |
|---|---|---|---|---|---|---|
| | 0.1 | 0.25 | 0.5 | 0.75 | 0.9 | 0.95 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 2 | 2 | 2 | 2 |
| 4 | 1 | 2 | 3 | 3 | 4 | 4 |
| 8 | 2 | 3 | 5 | 6 | 7 | 8 |
| 16 | 3 | 5 | 9 | 12 | 15 | 15 |
| 32 | 4 | 9 | 17 | 24 | 29 | 30 |
| 64 | 7 | 17 | 33 | 48 | 58 | 61 |
| 128 | 14 | 33 | 65 | 96 | 115 | 122 |
| 256 | 27 | 65 | 129 | 192 | 231 | 243 |
| 512 | 52 | 129 | 257 | 384 | 461 | 486 |
| 1024 | 103 | 257 | 513 | 768 | 922 | 973 |
| 2048 | 206 | 513 | 1025 | 1536 | 1843 | 1946 |
| 4096 | 411 | 1025 | 2049 | 3072 | 3687 | 3891 |
| 8192 | 820 | 2049 | 4097 | 6144 | 7373 | 7782 |
| 16384 | 1639 | 4097 | 8193 | 12288 | 14746 | 15565 |
| 32768 | 3278 | 8193 | 16385 | 24576 | 29491 | 31130 |
| 65536 | 6555 | 16385 | 32769 | 49152 | 58983 | 62259 |

- Técnicas de Análisis

# Trabajo Relacionado
## Herramientas

- Pruebas de Rendimiento (STREAM/HPL/Intel MPI/HPCC)
- Utilización de las Herramientas
- Tiempo de Ejecución
- Perfil de Ejecución Funcional
- Asistido por *Hardware*
- Reporte de Vectorización

# Descripción del Problema
## Análisis de Rendimiento

- **Problemas**: Interacción Humana, Manejo de Herramientas, Recopilación y Representación, Optimización Temprana, Implementación Teórica.
- **Optimización**: Código, Ejecución, Memoria, Precarga, Punto Flotante.
- **Infraestructura**: Reusabilidad, Configurabilidad, Portabilidad, Extensibilidad, Simplicidad.

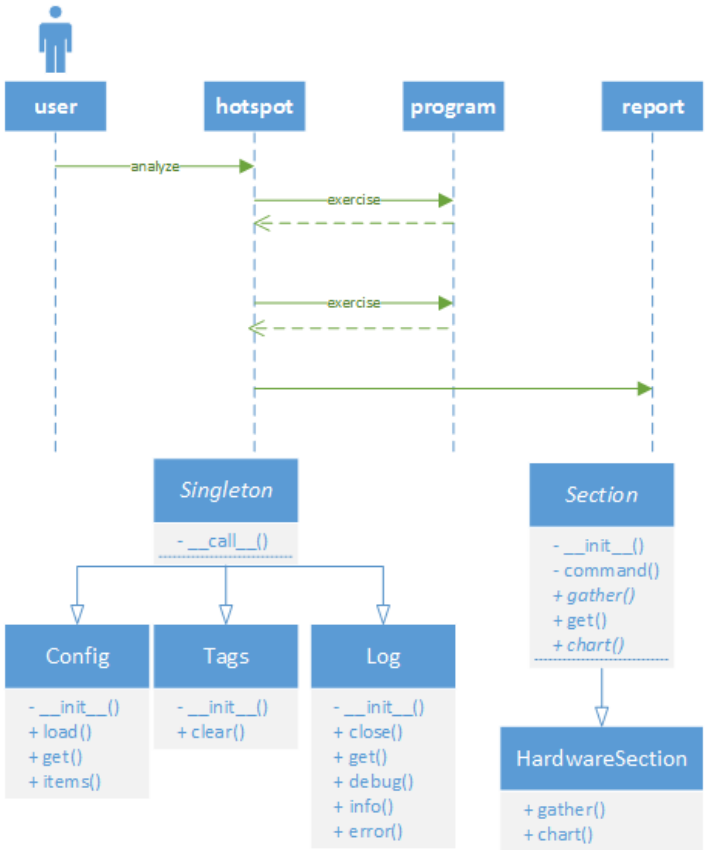
# Propuesta de Solución
## Procedimiento

- Procedimiento iterativo.

1. Se establece una línea base.
2. Se instrumenta el binario.
3. Se comprueba estabilidad.
4. Se comprueba uso de recursos.
5. Se realiza un perfil.
6. Se realiza una optimización.
7. Se comprueba y se itera.

# Propuesta de Solución
## Infraestructura

- hotspot
- Combina `gcc`, `make`, `prof`, `gprof`, `pidstat`, `latex`.
- Arquitectura y diseño.



# Propuesta de Solución
## Implementación

- GNU/Linux, Ubuntu, Python, matplotlib, numpy.
- Configuración
- Reporte: Formato portable, hipervínculo, secciones y gráficos, tendencia, referencias, inglés.
- Reporte: Resumen, Contenido, Programa, Capacidad del Sistema, Carga de Trabajo, Escalabilidad, Perfil de Ejecución, Bajo Nivel, Referencias.

# Casos de Aplicación
## Sistema de Prueba

- Hardware y Software

```
memory      7984MiB System memory
processor   Intel(R) Core(TM) i5-3320M CPU @ 2.60GHz
bridge      440FX - 82441FX PMC [Natoma]
bridge      82371SB PIIX3 ISA [Natoma/Triton II]
storage     82371AB/EB/MB PIIX4 IDE
network     82540EM Gigabit Ethernet Controller
bridge      82371AB/EB/MB PIIX4 ACPI
storage     82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode]
```

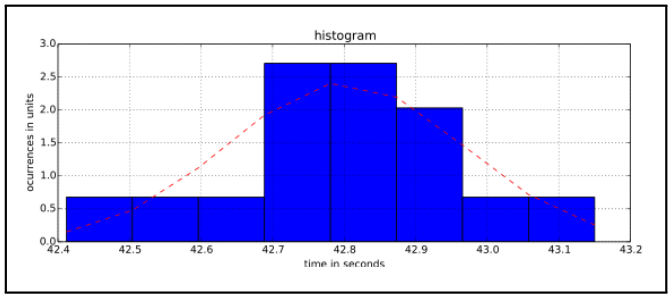
- Pruebas de Rendimiento

1. Host: `ubuntu`
2. Distribution: `Ubuntu, 14.04, trusty`.
   This codename provides LSB (Linux Standard Base) and distribution-specific information.
3. Compiler: `gcc (Ubuntu 4.8.2-19ubuntu1) 4.8.2`.
   Version number of the compiler program.
4. C Library: `GNU C Library (Ubuntu EGLIBC 2.19-0ubuntu6.6) stable release version 2.19`.
   Version number of the C library.

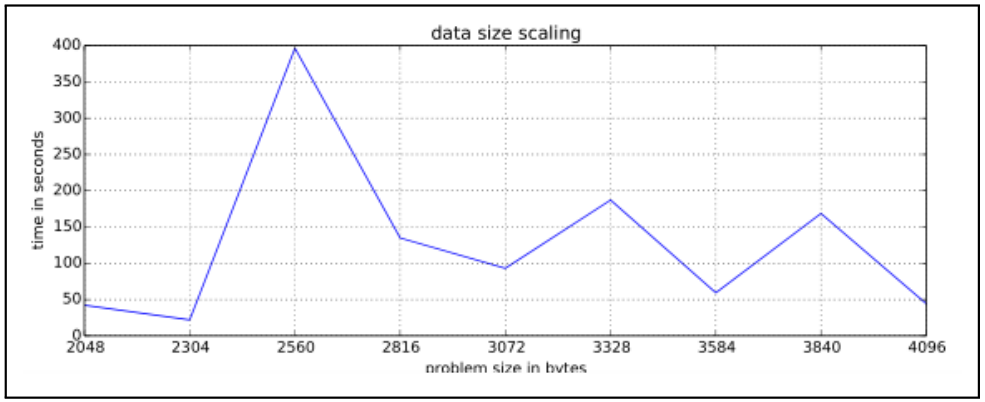
# Casos de Aplicación
## Multiplicación de Matrices

- Estabilidad

Execution time:
(a) problem size range: 2048 - 4096
(b) geomean: 42.80017 seconds
(c) average: 42.80048 seconds
(d) stddev: 0.16483
(e) min: 42.41161 seconds
(f) max: 43.15019 seconds
(g) repetitions: 16 times

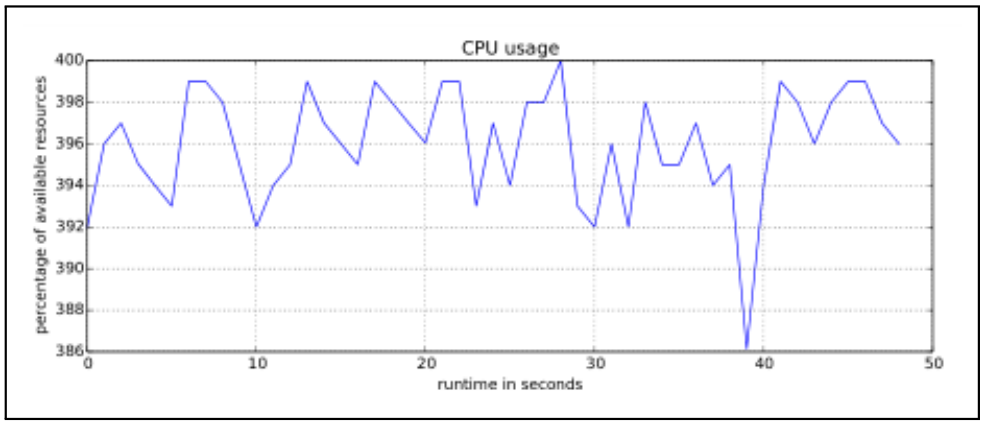



- Escalabilidad



# Casos de Aplicación
## Transmisión de Calor en 2D

- Utilización de recursos




- Instrucciones

```
        :   /* set boundary values */
        :   for (i = 0; i < CRESN; i++)
        :     {
        :       if (i < 256 || i > 768)
        :         solution[cur_gen][i][0] = solution[cur_gen][i][1];
   6.77 :   400e3b:       movsd  0x8(rdx),xmm0
  36.41 :   400e40:       movsd  xmm0,(rdx)
   2.92 :   400e44:       jmpq   400db7 <compute_one_iteration+0x37>
```

# Casos de Aplicación
## Conjunto de Mandelbrot

- Estructuras

```
struct complextype {
float                          real;                /*     0     4 */
float                          imag;                /*     4     4 */

/* size: 8, cachelines: 1, members: 2 */
/* last cacheline: 8 bytes */
};
```

```
      :          do
      :            {
      :              temp = z.real * z.real - z.imag * z.imag + c.real;
      :              z.imag = 2.0 * z.real * z.imag + c.imag;
 0.57 :   400999:       addsd  xmm6,xmm0
 6.56 :   40099d:       unpcklpd xmm0,xmm0
      :              z.real = temp;
      :              lensq = z.real * z.real + z.imag * z.imag;
 5.17 :   4009a5:       movaps xmm1,xmm0
 2.26 :   4009a8:       mulss  xmm1,xmm0
 3.94 :   4009ac:       movaps xmm2,xmm3
 0.13 :   4009af:       mulss  xmm2,xmm3
19.29 :   4009b3:       addss  xmm3,xmm0
```

# Conclusiones y Trabajo Futuro

## Conclusiones

- La optimización requiere un análisis disciplinado para ser efectiva.
- Se provee una infraestructura automática que soporta el analísis.

# Conclusiones y Trabajo Futuro
## Trabajo Futuro

- Extensión.
- Aplicación.
- Soporte MPI.