



Programación de SM de memoria compartida

**Arquitectura y Tecnología de Computadores
Universidad de Oviedo**

Noelia Díaz Gracia UOI88772

Lucas Díaz Sanzo UOI89670

Aida Fernández Méndez UOI6559

Contenidos

- Introducción
- Compiladores paralelos
 - Open MP
- Hilos y memoria compartida
 - Sincronización
- UPC (Unified Parallel C)
- Conclusiones

Introducción

- Memoria compartida: Hardware/Software
- Retos:
 - Acceso a memoria independiente
 - Coherencia
 - Sincronización

Introducción

- Nuevos lenguajes de programación
- Modificación de lenguajes existentes
- Subrutinas y librerías
- Compilación paralela
- Procesos Unix(ICP)
- Hilos

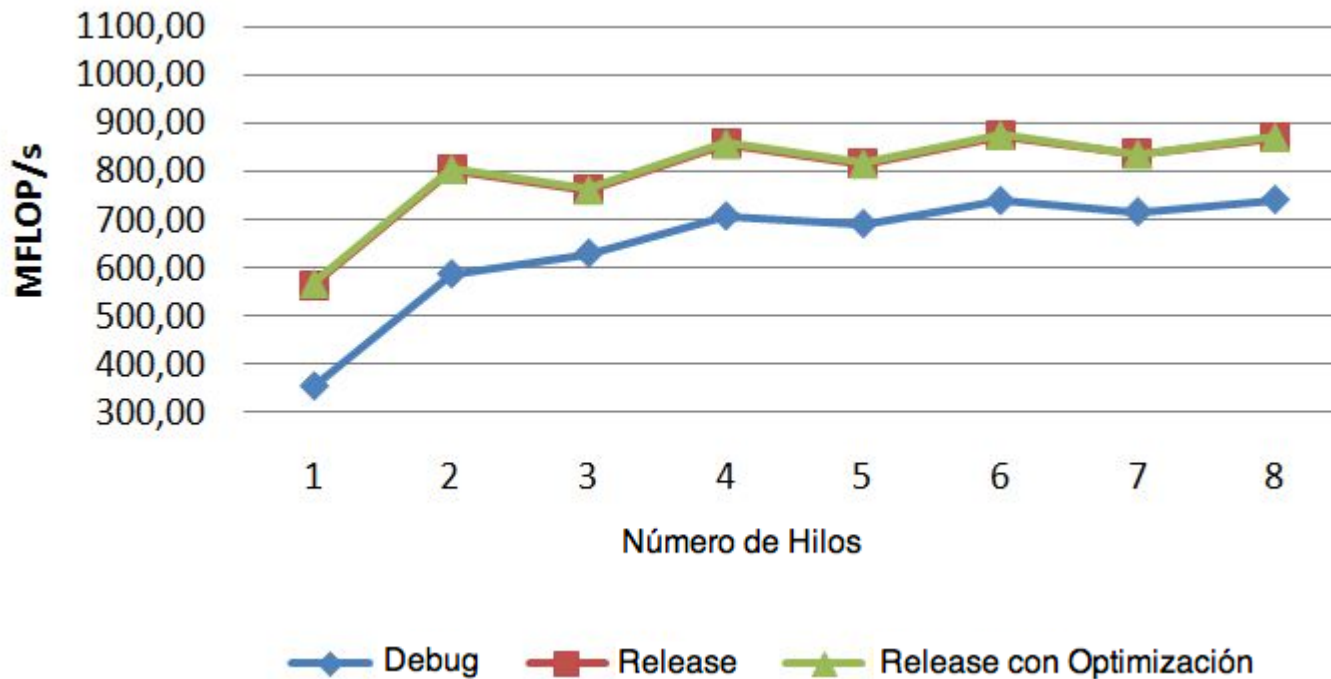
Compiladores paralelos

- Directivas y variables de entorno
 - Integridad de datos transparente
- División ➡ Análisis de dependencias
- Mayor aplicación: bucles
- Uso en aplicaciones distribuidas
- Código compatible con compiladores no paralelos
- Ejemplo: OpenMP
 - C/C++, Fortran
 - `# pragma omp <directiva> [cláusula [, ...] ...]`

Open MP: Programa de prueba

```
#pragma omp parallel for private(sum, rowend, rowbeg, nz)
//multiplicación de matrices
for (i = 0; i < num_rows; i++)
{
    sum = 0.0;
    rowend = Arow[i+1];
    rowbeg = Arow[i];
    for (nz=rowbeg; nz<rowend; ++nz)
    {
        sum += Aval[nz] * x[Acol[nz]];
    }
    y[i] = sum;
}
```

Open MP: Rendimiento



Gmax = 45%

Hilos y memoria compartida

- Estándar: IEEE 1003.1.c POSIX THREADS
 - Gestión de hilos
 - Sincronización
- Aplicaciones prácticas
 - Multiplexación I/O
 - Procesamiento asíncrono
 - Handler de señal
- Modelos de programa
 - Jefe-subordinado
 - Trabajo en equipo
 - Tubería

Proceso multihilo

Proceso

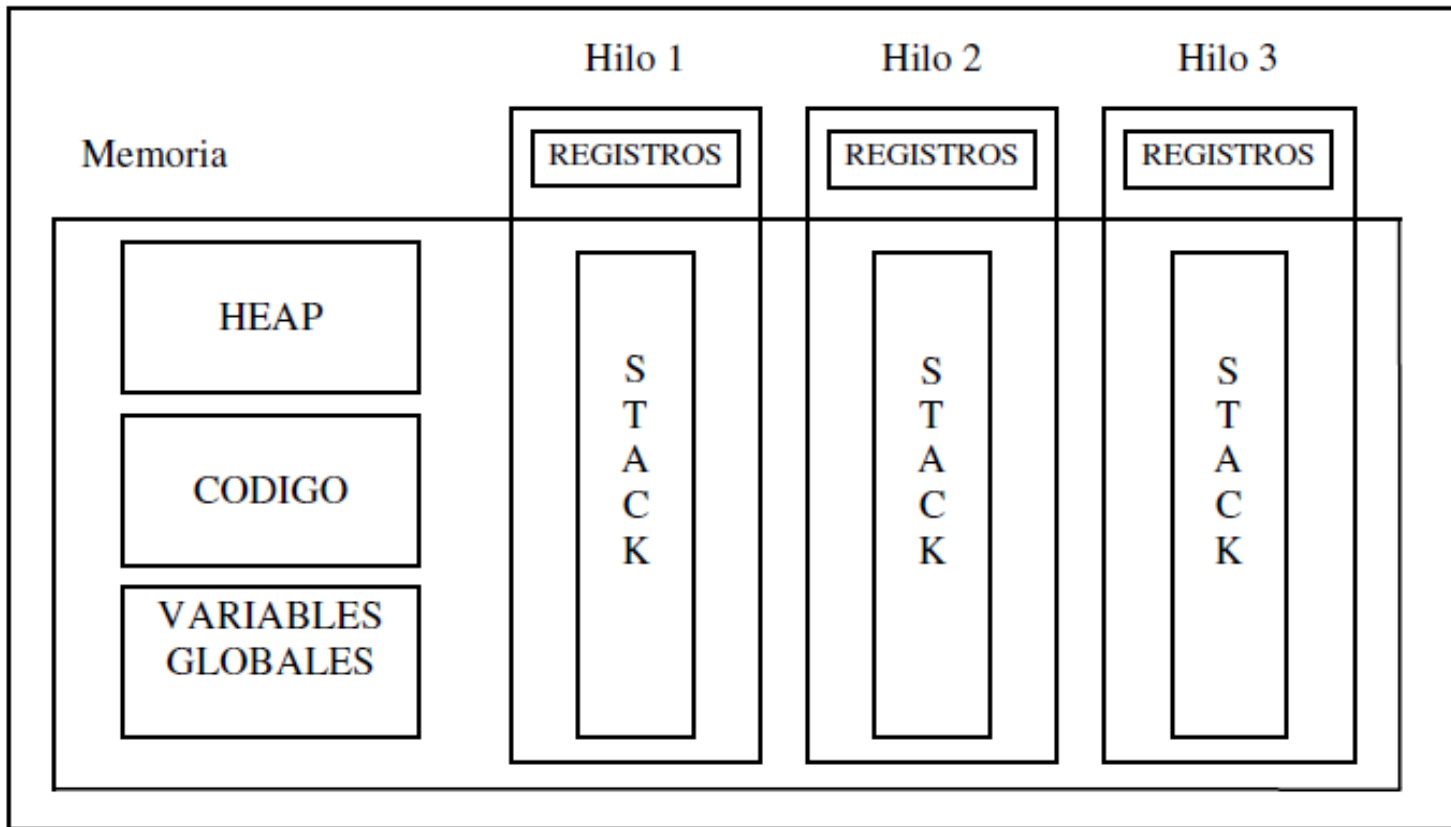


Figura 3. Proceso Unix con 3 hilos.

Sincronización de hilos

- **Objetivos**
 - Sección crítica
 - Exclusión mutua
 - Progreso y espera limitada
- **Mecanismos**
 - Granularidad del bloqueo
 - Variables de sincronización
- **Problemas**
 - Interbloqueo

Variables de sincronización

- **Mútex**
 - Pthread_mutex_lock y pthread_mutex_unlock
- **Variables condicionales**
 - Pthread_cond_signal
 - Pthread_cond_broadcast
 - Pthread_cond_timedwait
- **Semáforos**
 - Sem_post
 - Sem_wait

UPC: Control de flujo

- THREADS, MYTHREADS

```
for (i = 0; i < N; i++){  
    if (MYTHREAD == i%THREADS){  
        //operar  
    }  
}
```

- ¿Mejorable?

UPC (Unified Parallel C)

- Lenguaje de programación paralelo
- Extensión de ISO C
- Compilador upc: similar a gcc
 - `upcc -o holamundo -T=4 holamundo.upc`

UPC: Control de flujo

```
for (i = MYTHREAD; i < N; i+=THREADS){  
    //operar  
}
```

UPC: Modelo de memoria

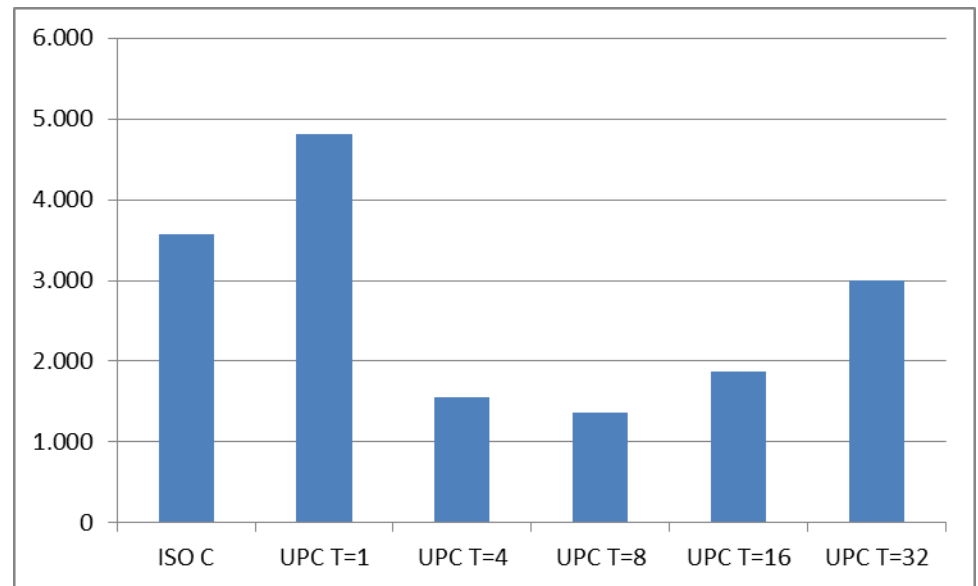
	Hilo 0	Hilo 1	---	Hilo n
espacio compartido	Compartido			
espacio privado	Privado hilo 0	Privado hilo 1	---	Privado hilo n

- Punteros a región compartida
 - `static shared int a;`
- Punteros a región privada

UPC: Rendimiento

- Programa prueba: matriz por vector

Lenguaje	Tiempo de ejecución(s)
ISO C	3.570
UPC T=1	4.805
UPC T=4	1.560
UPC T=8	1.358
UPC T=16	1.872
UPC T=32	2.995



Gmax: 228%

Conclusiones

- Complicación programación, depuración y detección de errores
- Mejora de rendimiento

Bibliografía

- 1. Cory Quammen, “Introducción a la Programación en ambientes con Memoria Compartida y Memoria-Distribuída”,
<http://bc.inter.edu/facultad/jyeckle/cursos/parallel%20computing/papers/2-%20paradigma%20threads%20y%20mpi.pdf>
- 2. John Mellor Crummey, “Programming Shared-memory Platforms with Pthreads”, Enero 2011, <http://www.clear.rice.edu/comp422/lecture-notes/comp422-2011-Lecture6-Pthreads.pdf>
- 3. Manish Parashar, “Shared Memory Programming: Threads”, 1998,
http://www.ece.rutgers.edu/~parashar/Classes/98-99/ece566/slides/lecture_pthreads.PDF
- 4. Domingo Gimenez Cánovas, “Programación paralela: Lenguajes y Modelos”, 2003,
http://www.slidefinder.net/p/programación_paralela_tema_lenguajes_modelos/15238786
- 5. S. Dasgupta, “Computer Architecture: A Modern Synthesis” (Tomo 2), John Wiley & Sons, 1989.
- 6. J.L. Hennessy y D.A. Patterson, “Computer Architecture: A Quantitative Approach”, Morgan Kaufman Publishers, 2002.
- 7. Apuntes de Programación paralela de la Universidad Politécnica de Valencia (Temas 15 y 16),

Bibliografía (II)

- 8. Ruud Van der Pass, “Basic concepts in Paralellization”, 2010, http://www.compunity.org/training/tutorials/2%20Basic_Concepts_Parallelization.pdf
-
- 9. Ruud Van der Pass, “An overview to OpenMP”, 2010.
- 10. B. Chapman, G. Jost, R. van der Pas, “Using OpenMP”, October 2007.
- 11. D. Giménez Cánovas y Javier Cuenca, “Programación en Memoria Compartida”, 2005, <http://dis.um.es/~domingo/doctorado/0506/ComMatParUPCT/Sesion7ProMemCom.pdf>
- 12. Parallel Programming: Techniques and Applications using Networked Workstations and Parallel Computers Barry Wilkinson and Michael Allen □ Prentice Hall, 1999
- 13. T. El-Gazawi, W. Carlson, T. Sterling, K. Yelick, “UPC Distributed Shared Memory Programming”, Wiley-Intersign, 2005
- 14. T. El-Gazawi, “Unified Paralell C – UPC Tutorial”, 2009, The George Washington University.



FIN