


Ejemplo de optimización secuencial



Luis M^a Costero Valero
Jesús Javier Doménech Arellano
Hristo Ivanov Ivanov



CHOLSKY

```
11 double* cholesky_blocked(const int n, double* A)
12 {
13     double *L = (double*)calloc(n * n, sizeof(double));
14     if (L == NULL)
15         exit(EXIT_FAILURE);
16
17     for (int i = 0; i < n; i++)
18         for (int j = 0; j < (i+1); j++) {
19             double s = 0;
20             for (int k = 0; k < j; k++)
21                 s += L[i * n + k] * L[j * n + k];
22             L[i * n + j] = (i == j) ?
23                 sqrt(A[i * n + i] - s) :
24                 (1.0 / L[j * n + j] * (A[i * n + j] - s));
25         }
26
27     return L;
28 }
```

- ¿Qué hace esto?

Siendo A y C matrices (N x N) y C una matriz triangular:

$$A = C * C^T$$

- ¿Para qué sirve?

Resolución de sistemas: la mayoría de los problemas de computación se reducen a resolver sistemas.

- ¿Cuál es su complejidad teórica?

$O(N^3/3)$

Características de las pruebas

❖ Máquina:

- Intel Xeon E5530 – 2.4 GHZ
- 48 GB RAM

❖ Software:

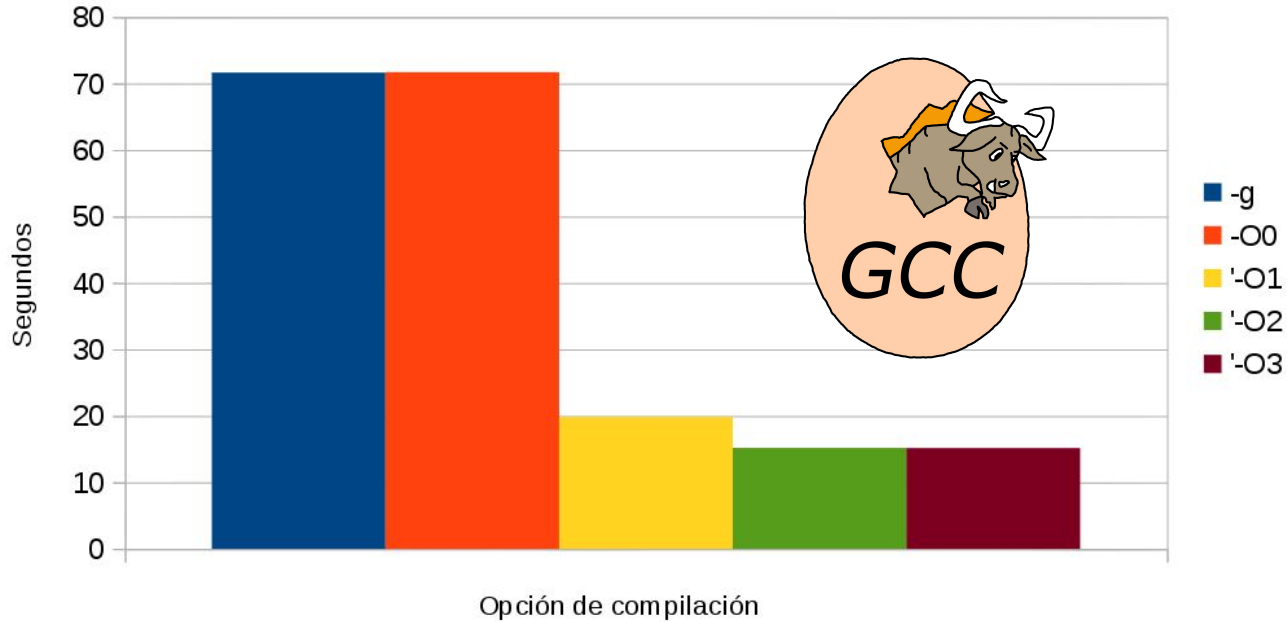
- Debian Jessie, Kernel 3.14-1-amd64
- gcc v4.8.3
- icc v14.0.3 / MKL v2014

❖ Pruebas:

- Mediciones de tiempo desde código (llamadas a funciones del kernel).
- Mediciones sólo del algoritmo. No se tiene en cuenta inicialización / finalización.
- Repetición del algoritmo. Se considera la media de tiempos.
- Matriz de 4096x4096 elementos en DOUBLE.

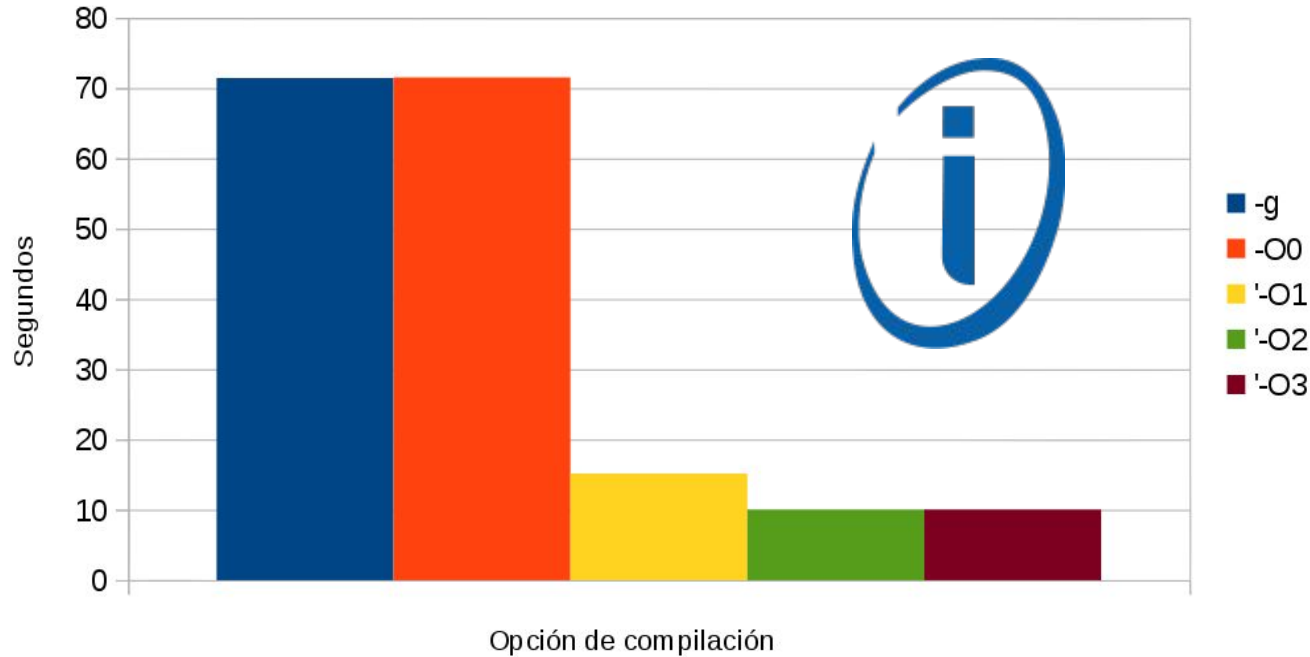
gcc (GNU Compiler Collection)

Tiempo de ejecución según el flag de compilación (GCC)

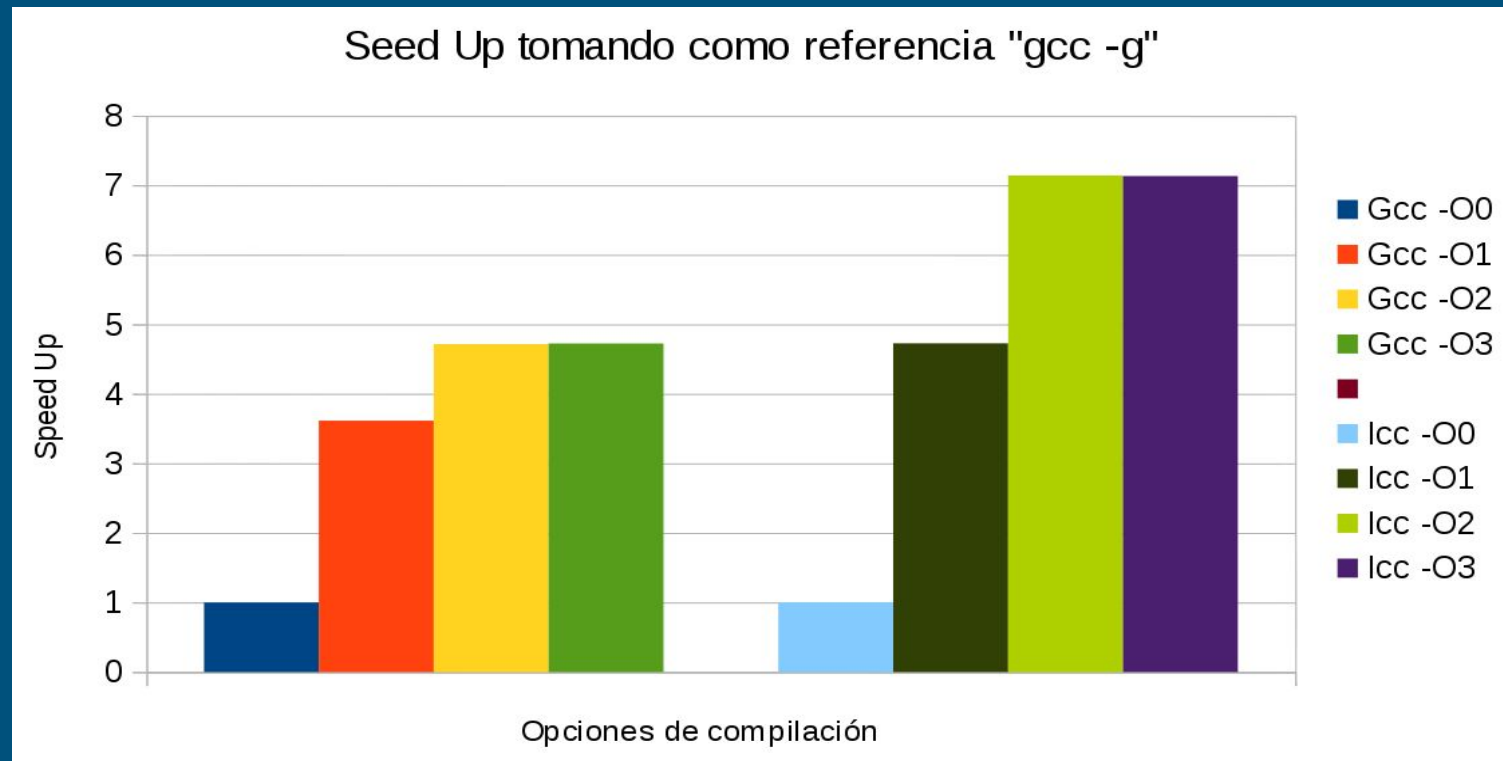


icc (Intel C Compiler)

Tiempo de ejecución según el flag de compilación (ICC)



SpeedUp - Comparativa



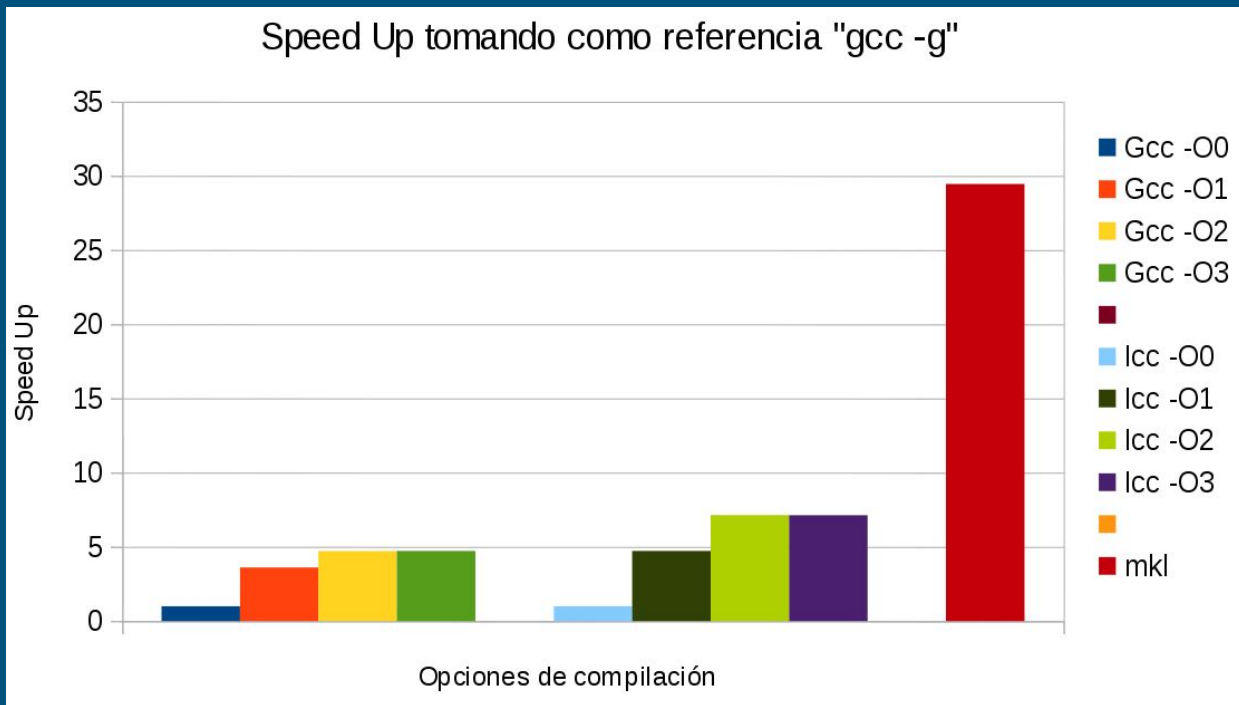
Librería matemática - MKL

- Llamadas a la librería proporcionada por Intel.
- Distintas opciones de ejecución

```
18
19 double *cholesky_blocked(const int n, double* A)
20 {
21     static int INFO;
22     static const char L = 'L';
23     dpotrf_(&L, &n, A, &n, &INFO);
24
25     return A;
26 }
27
```

```
11
12 MKL_LIB_DIR=/opt/intel/mkl/lib/intel64/
13 MKL_INC_DIR=/opt/intel/mkl/include/
14
15 LIBS = -L$(MKL_LIB_DIR) -lmkl_sequential -lmkl_core -lmkl_rt
16 INCS = -I$(MKL_INC_DIR)
17
```

MKL - SpeedUp



Bibliografía

- ❖ <http://pm.bsc.es/>
- ❖ <https://software.intel.com/en-us/c-compilers>
- ❖ <https://gcc.gnu.org/>
- ❖ https://es.wikipedia.org/wiki/Factorizaci%C3%B3n_de_Cholesky