

Covarianza paralela

La covarianza es un valor que indica el grado de variación conjunta de dos variables aleatorias, es utilizada para ver si existe dependencia entre ambas variables.

1) Descripción del algoritmo

Expresión matemática

$$\sigma_{xy} = \frac{\sum f_i x_i y_i}{N} - \bar{x} \bar{y}$$

Código secuencial

```
1  template <class T>
2  double covarianza(vector<T> a, vector<T> b) {
3      T sum_a = 0, sum_b = 0, sum_ab = 0;
4      double tam = min(a.size(), b.size());
5
6      for (int i = 0; i < (int)tam; i++){
7          sum_ab += a[i] * b[i];
8          sum_b += b[i];
9          sum_a += a[i];
10     }
11     return (sum_ab / tam) - (sum_a / tam) * (sum_b / tam);
12 }
```

2) Paralelización

Código paralelo:

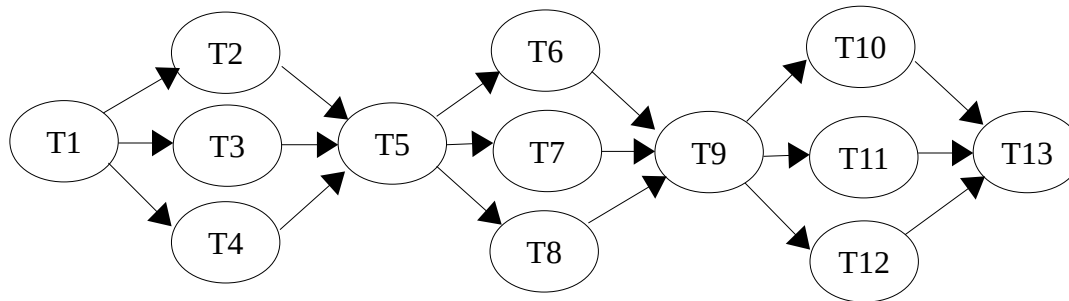
```
1  template <class T>
2  double covarianza(vector<T> a, vector<T> b) {
3      T sum_a = 0, sum_b = 0, sum_ab = 0;
4      double tam = min(a.size(), b.size());
5
6      #pragma omp parallel for reduction(+ : sum_ab)
7      for (int i = 0; i < (int)tam; i++)
8          sum_ab += a[i] * b[i];
9
10     #pragma omp parallel for reduction(+ : sum_b)
11     for (int i = 0; i < (int)tam; i++)
12         sum_b += b[i];
13
14     #pragma omp parallel for reduction(+ : sum_a)
15     for (int i = 0; i < (int)tam; i++)
16         sum_a += a[i];
17 }
```

```

18  return (sum_ab / tam) - (sum_a / tam) * (sum_b / tam);
19 }

```

Grafo de tareas para un tamaño de problema pequeño con asignación a flujo de instrucciones



Problema de ejemplo:

vector1 = {2,3,4}

vector2 = {5,6,7}

T1: Buscar el vector de menor tamaño (Linea 4)

T2, T3, T4: Paralelización de bucle de la linea 7

T2: Multiplicar 2 por 5 y sumarlo a sum_ab

T3: Multiplicar 3 por 6 y sumarlo a sum_ab

T4: Multiplicar 4 por 7 y sumarlo a sum_ab

T5: Barrera implícita de #pragma omp parallel for

T6, T7, T8: Paralelización de bucle de la linea 11

T6: Sumar 2 a sum_b

T7: Sumar 3 a sum_b

T8: Sumar 4 a sum_b

T9: Barrera implícita de #pragma omp parallel for

T10, T11, T12: Paralelización de bucle de la linea 15

T10: Sumar 5 a sum_a

T11: Sumar 6 a sum_a

T12: Sumar 7 a sum_a

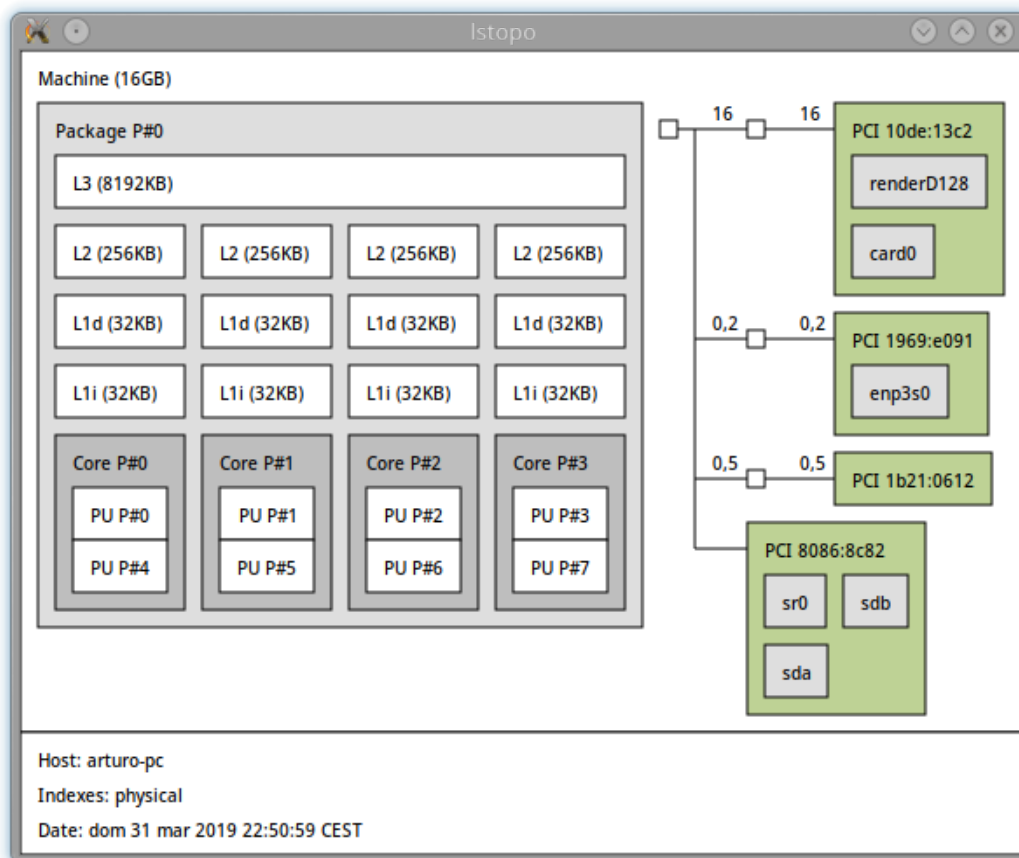
T13: Barrera implícita de #pragma omp parallel for, calculos finales y devolución de la covarianza

3) Evaluación de prestaciones

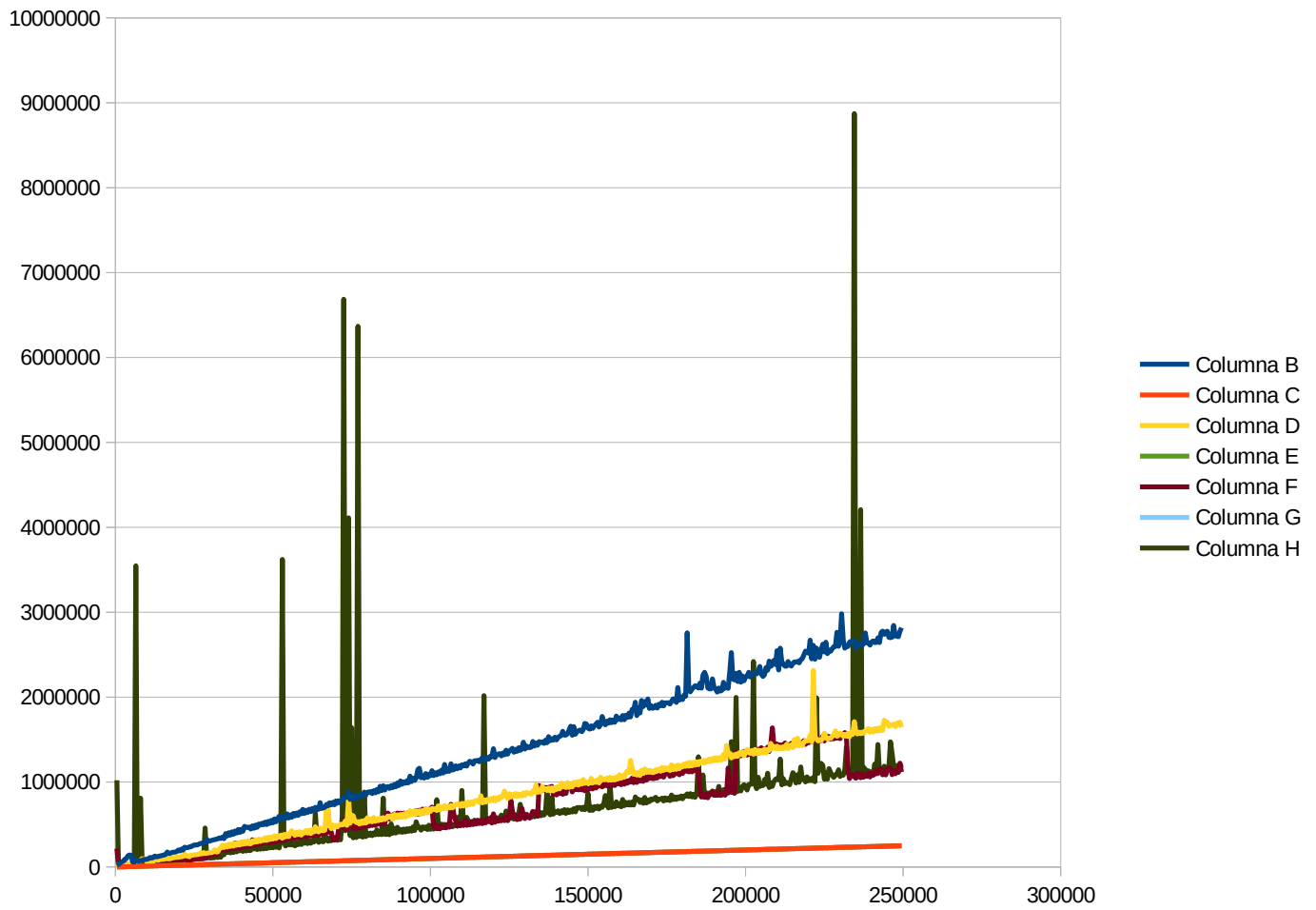
Descripción de la arquitectura

Arquitectura:	x86_64
modo(s) de operación de las CPUs:	32-bit, 64-bit
Orden de los bytes:	Little Endian
Tamaños de las direcciones:	39 bits physical, 48 bits virtual
CPU(s):	8
Lista de la(s) CPU(s) en línea:	0-7
Hilo(s) de procesamiento por núcleo:	2
Núcleo(s) por «socket»:	4
«Socket(s)»	1
Modo(s) NUMA:	1
ID de fabricante:	GenuineIntel
Familia de CPU:	6
Modelo:	60

Nombre del modelo: Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz
 Revisión: 3
 CPU MHz: 1738.690
 CPU MHz máx.: 4400,0000
 CPU MHz mín.: 800,0000
 BogoMIPS: 8003.47
 Virtualización: VT-x
 Caché L1d: 32K
 Caché L1i: 32K
 Caché L2: 256K
 Caché L3: 8192K
 CPU(s) del nodo NUMA 0: 0-7
 Indicadores: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge
 mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx
 pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xt
 opology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vmx est
 tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt
 tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm cpuid_fault invpcid_single
 pti ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsbase
 tsc_adjust bmi1 avx2 smep bmi2 erms invpcid xsaveopt dtherm ida arat pln pts
 flush_l1d



Escalabilidad fuerte



Para mas detalle mirar el archivo covarianza.ods