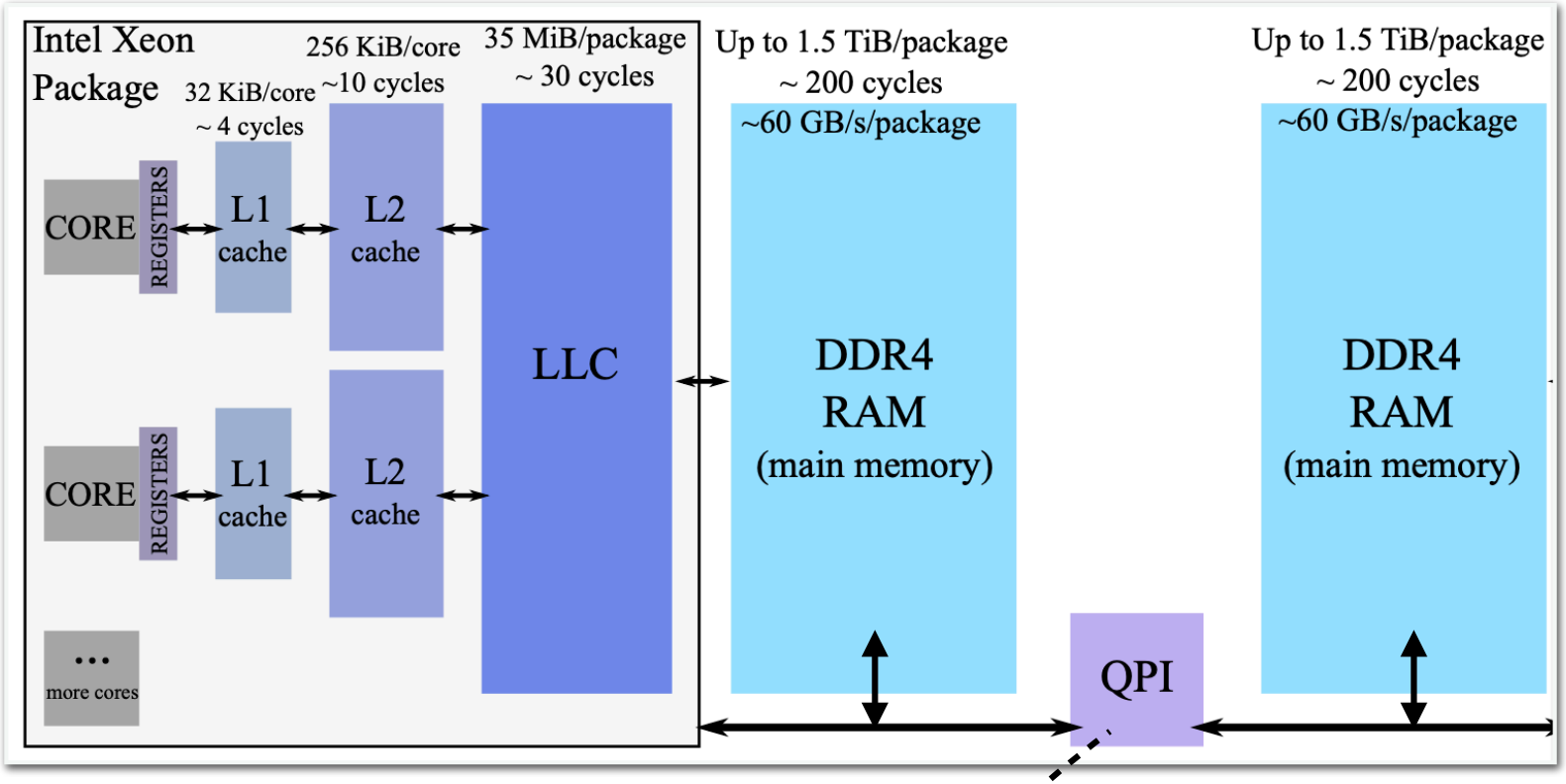
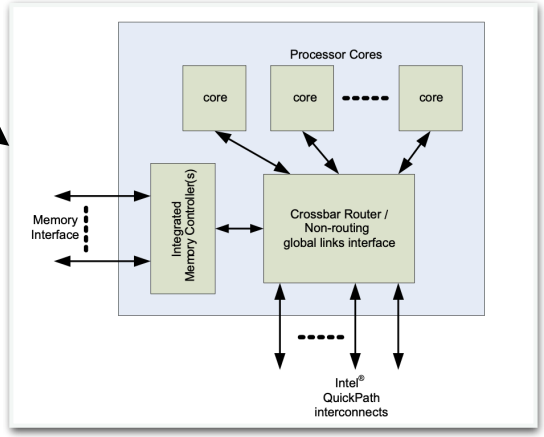
Programación Paralela

Gabriel Astudillo Muñoz

Utilización de la memoria cache

Organización de memoria en Intel Xeon

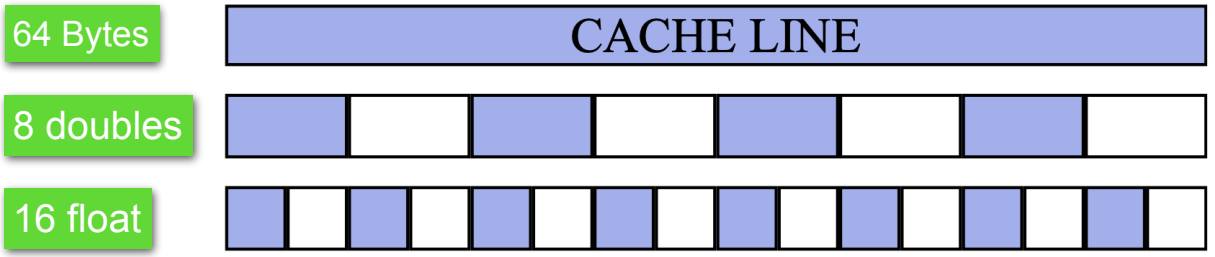




https://software.intel.com/sites/products/collateral/hpc/vtune/performance_analysis_guide.pdf

Organización de la memoria cache

La transferencia de datos entre memoria principal y memoria cache es a través de bloques de datos



Ejemplo Core i5-9400

hwloc-ls

```
Machine (7834MB total)
  Package L#0
    NUMANode L#0 (P#0 7834MB)
    L3 L#0 (9216KB)
      L2 L#0 (256KB) + L1d L#0 (32KB) + L1i L#0 (32KB) + Core L#0 + PU L#0 (P#0)
      L2 L#1 (256KB) + L1d L#1 (32KB) + L1i L#1 (32KB) + Core L#1 + PU L#1 (P#1)
      L2 L#2 (256KB) + L1d L#2 (32KB) + L1i L#2 (32KB) + Core L#2 + PU L#2 (P#2)
      L2 L#3 (256KB) + L1d L#3 (32KB) + L1i L#3 (32KB) + Core L#3 + PU L#3 (P#3)
      L2 L#4 (256KB) + L1d L#4 (32KB) + L1i L#4 (32KB) + Core L#4 + PU L#4 (P#4)
      L2 L#5 (256KB) + L1d L#5 (32KB) + L1i L#5 (32KB) + Core L#5 + PU L#5 (P#5)
```

`lscpu | grep cache`

- L1d cache: 192KB
- L1i cache: 192KB
- L2 Cache: 1.5MB
- L3 Cache: 9MB

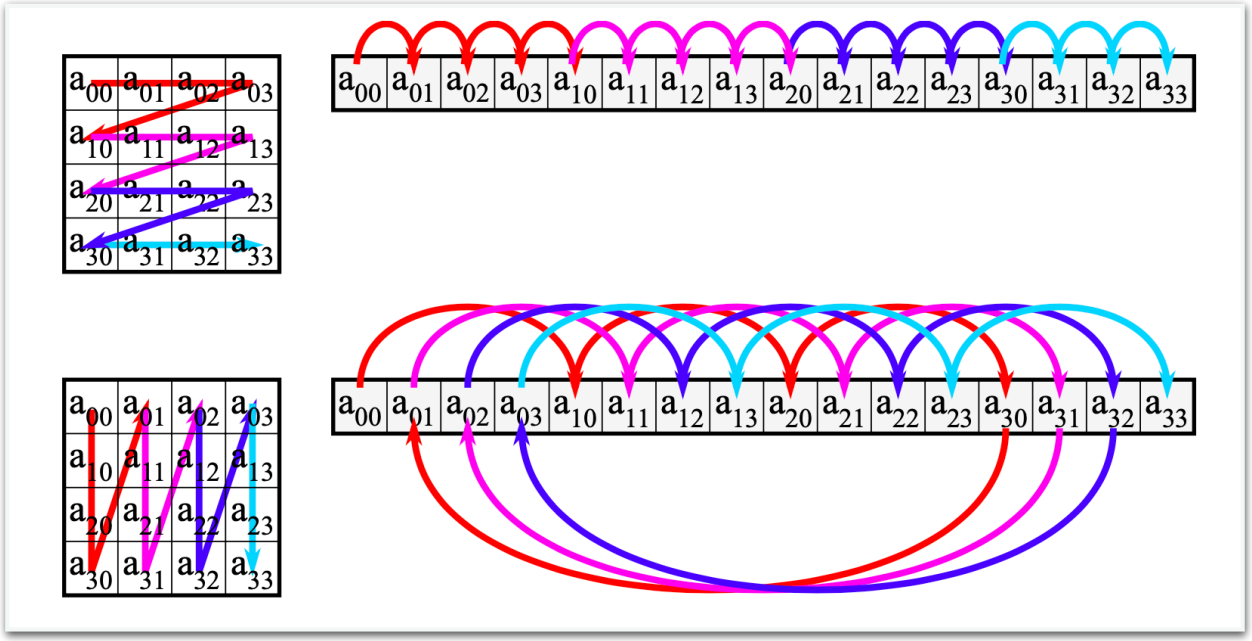
`getconf -a | grep CACHE`

```
LEVEL1_DCACHE_SIZE                 32768
LEVEL1_DCACHE_ASSOC                8
LEVEL1_DCACHE_LINESIZE             64

LEVEL2_CACHE_SIZE                  262144
LEVEL2_CACHE_ASSOC                 4
LEVEL2_CACHE_LINESIZE              64

LEVEL3_CACHE_SIZE                  9437184
LEVEL3_CACHE_ASSOC                 12
LEVEL3_CACHE_LINESIZE              64
```

Condiciones de borde para el análisis

- La línea de la cache es de 64B (V)
- Cache no puede almacenar múltiples líneas (F)

Consecuencias



Los datos siempre deben estar en zonas contiguas de memoria

Caso matrices 2d:
Pedir una zona lineal de memoria

Creación

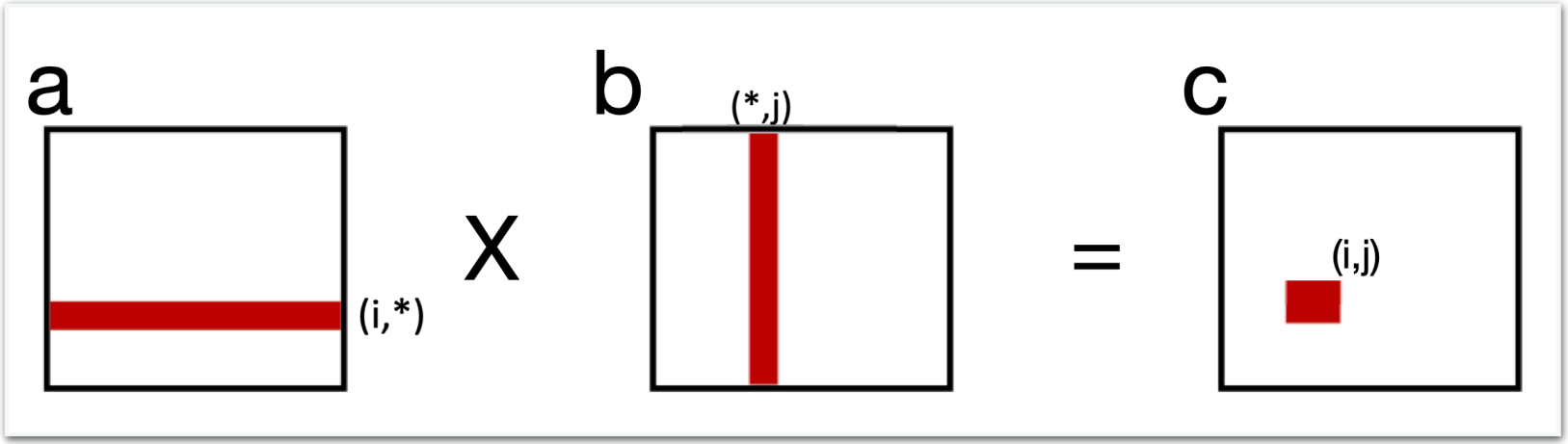
```
double* M = new double[nfil * ncol]
```

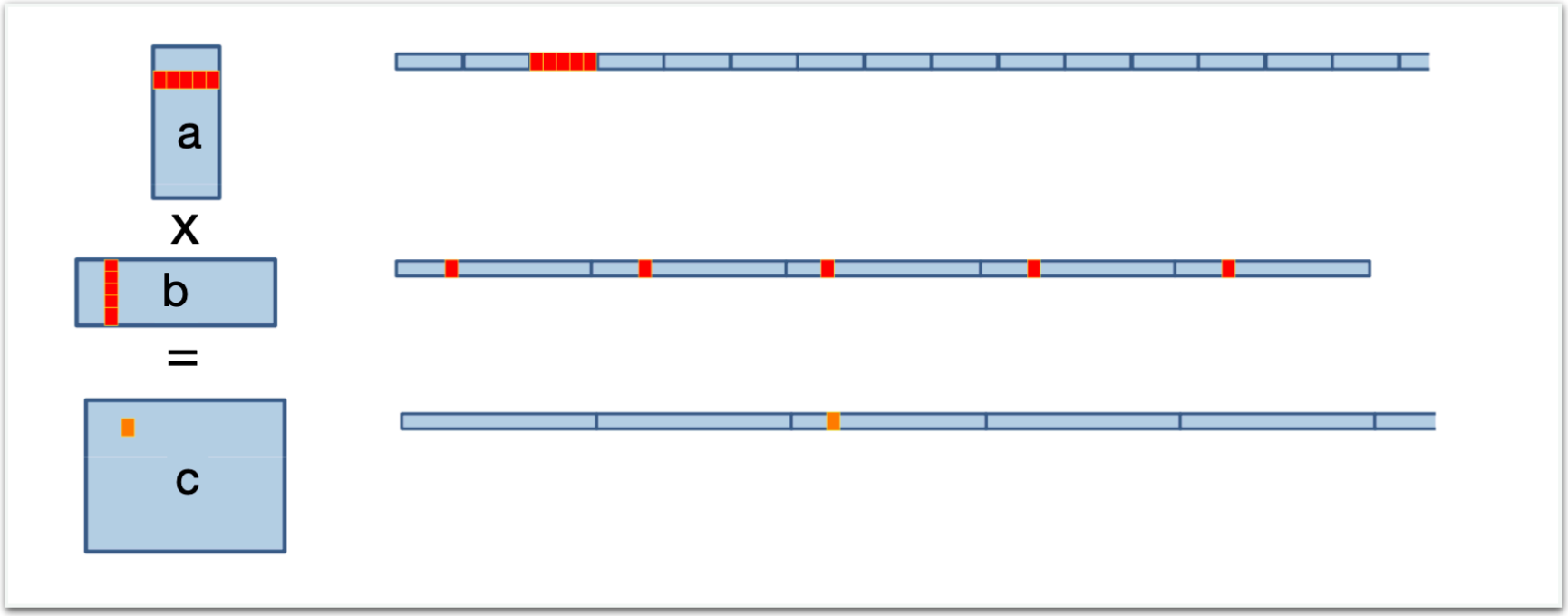
Acceso al componente $M_{i,j}$:

```
M[i*ncol+j] //con  0≤i<nfil, 0≤j<ncol
```

Ejemplo: Multiplicación de matrices

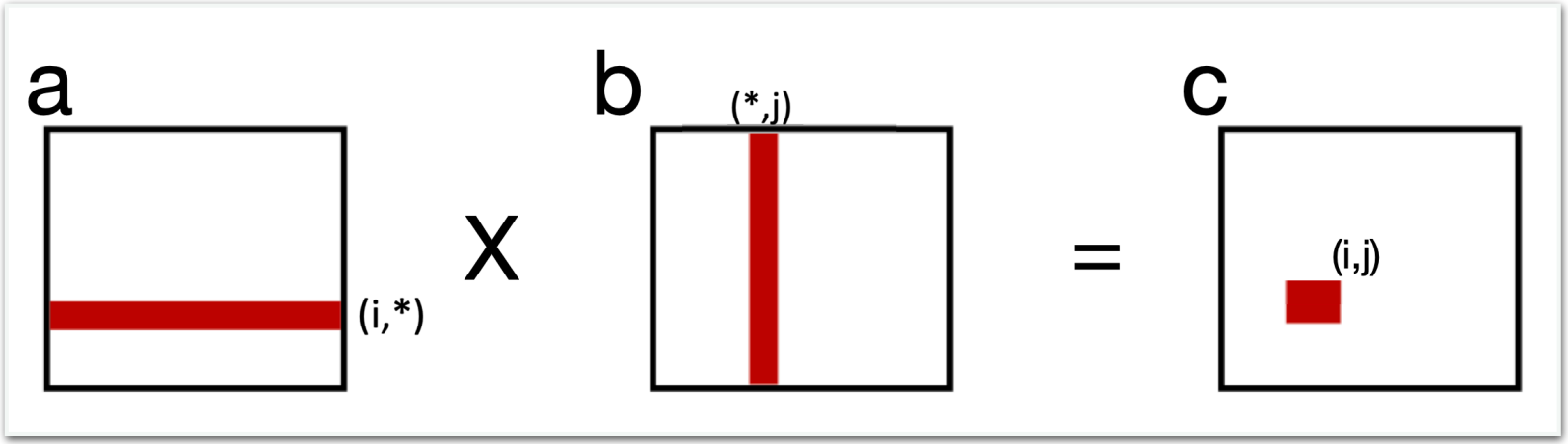
Caso 1)





Ejemplo: Multiplicación de matrices

Caso 1)



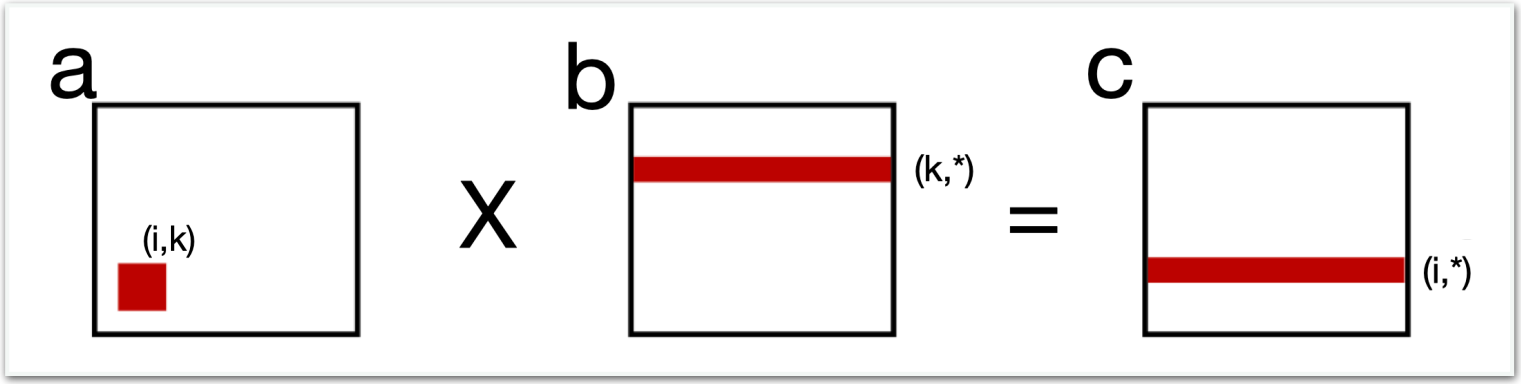

```
for(size_t i=0; i < filas(a); i++){
  for(size_t j=0; j < columnas(b); j++){
    for(size_t k=0; k < columnas(a); k++){
      c[i][j] += a[i][k] * b[k][j];
    }
  }
}
```

Tasa de pérdidas de la caché

a=0.125; b=1; c=0

Ejemplo: Multiplicación de matrices

Caso 2)




```
for(size_t k=0; k < columnas(a); k++){
  for(size_t i=0; i < filas(a); i++){
    float r = a[i][k]
    for(size_t j=0; j < columnas(b); j++){
       c[i][j] += r * b[k][j];
    }
  }
}
```

Tasa de pérdidas de la caché

a=0; b=0.125; c=0.125

Menor tasa de pérdidas

Mejor desempeño

Menor tiempo de ejecución