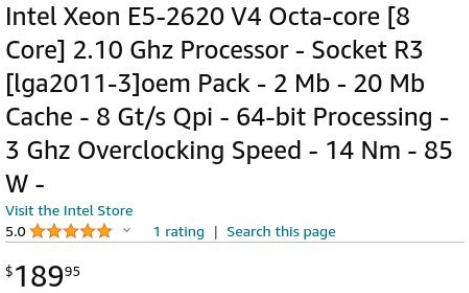
Verificando a influência da CPU em aplicações de álgebra linear densa com aceleradores

CMP270 - Introduction to High Performance Computing

Lucas Barros de Assis

Contexto

- O PCAD da UFRGS conta com 4 máquinas denominadas *tupi,* cada uma delas equipada com uma *NVIDIA RTX 4090*
- No entanto, enquanto as máquinas *tupi1* e *tupi2* têm um *Intel Xeon E5-2620 v4,* as máquinas *tupi3* e *tupi4* contam com um *Intel Core i9-14900KF*
- A partir dessa situação, buscaremos avaliar o impacto do processador utilizado na execução de rotinas de álgebra linear densa auxiliadas pelo acelerador disponível nessas máquinas



Hipótese

Dado que o processador *Intel Core i9-14900KF* é cerca de 2,8 vezes mais caro que o *Intel Xeon E5-2620 v4,* definimos a seguinte hipótese:

> O *speedup* obtido ao utilizar o processador de valor mais elevado é inferior à diferença de preço dos dois processadores disponíveis

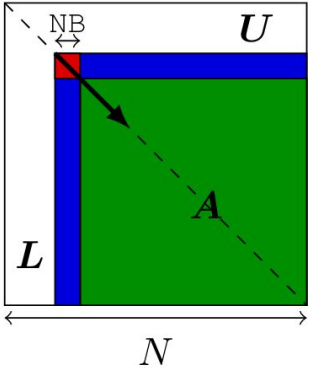
Experimentos

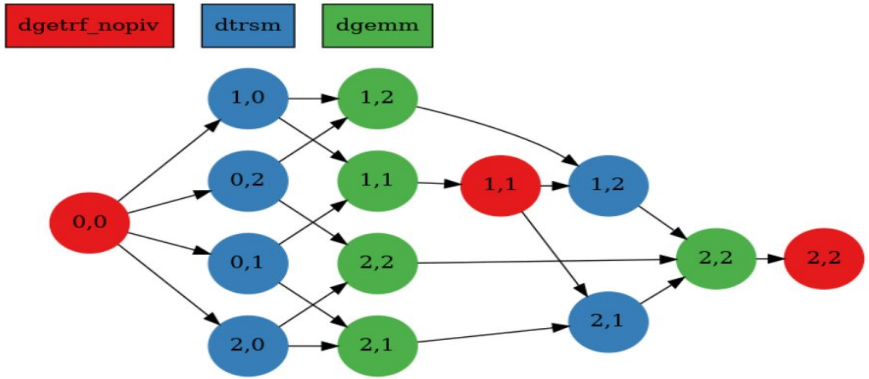
- Os experimentos para medição de performance serão realizados efetuando a operação de fatoração LU implementada pela biblioteca *Chameleon,* capaz de explorar o paralelismo através do suporte de execução *StarPU*
- O tempo medido equivale ao tempo de execução da rotina de fatoração, sem considerar a inicialização do programa

Matrizes ladrilhadas

A1,1	A1,2
A2,1	A2,2

```
for (k = 0; k < N; k++)
  DGTRF-NOPIV(RW, A[k][k]);
  for (m = k+1; m < N; m++)
    DTRSM(RW, A[m][k], R, A[k][k]);
    DTRSM(RW, A[k][m], R, A[k][k]);
  for (n = k+1; n < N; n++) // Update
    for (m = k+1; m < N; m++)
      DGEMM(RW, A[m][n], R, A[m][k],
                        R, A[k][n]);
```





Fonte: Nesi, Lucas Leandro and Mello Schnorr, Lucas and Legrand, Arnaud

Casos de teste previstos

As comparações serão feitas através de testes considerando as seguintes variáveis:

- Utilização da GPU
- Frequência do processador
- Tamanho do bloco utilizado
- Número de *threads* disponível

Implementação

```c
#include <chameleon.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>

int main(int argc, char* argv[])
{
    size_t N; // matrix order
    size_t NB; // number of rows/columns in a block
    int NCPU; // number of cores to use
    int NGPU; // number of gpus (cuda devices) to use
    int seedA = 42;
    struct timeval  tv1, tv2;
    double seconds_elapsed;

    /* descriptors necessary for calling CHAMELEON tile interface  */
    CHAM_desc_t *descA = NULL;

    int major, minor, patch;
    CHAMELEON_Version(&major, &minor, &patch);

    /* Initialize the number of CPUs to be used with threads */
    NCPU = atoi(argv[1]);
    NGPU = atoi(argv[2]);

    /* Linear system parameters */
    N = atoi(argv[3]);
    NB= atoi(argv[4]);

    /* Initialize CHAMELEON with main parameters */
    CHAMELEON_Init( NCPU, NGPU );

    CHAMELEON_Desc_Create(&descA,  NULL, ChamRealFloat,
                          NB, NB,  NB*NB, N, N, 0, 0, N, N, 1, 1);

    /* Generate A matrix with random values */
    CHAMELEON_splrnt_Tile( descA, seedA );

    /* Initiate time measure */
    gettimeofday(&tv1, NULL);

    /* Compute the LU factorization of the matrix A */
    CHAMELEON_sgetrf_nopiv_Tile(descA);

    /* Finish time measure */
    gettimeofday(&tv2, NULL);

    /* Deallocate data */
    CHAMELEON_Desc_Destroy( &descA );

    /* Finalize CHAMELEON */
    CHAMELEON_Finalize();

    seconds_elapsed = (double) (tv2.tv_usec - tv1.tv_usec) / 1000000 + (double) (tv2.tv_sec - tv1.tv_sec);
    printf("%d %d %d %d %f\n", NCPU, NGPU, N, NB, seconds_elapsed);
    return EXIT_SUCCESS;
}
```

Resultados iniciais - script de teste

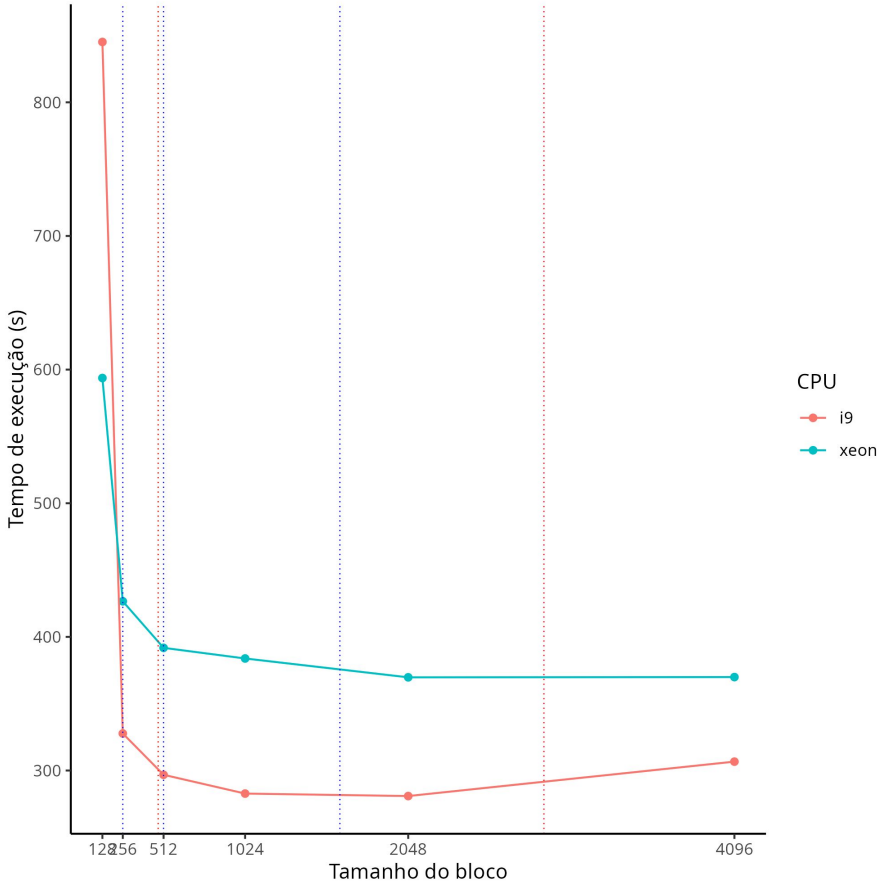
```
#!/usr/bin/bash
HOSTNAME=$(hostname)
OUTPUT_FILE=$(date +%d-%m_%H%M)
OUTPUT_FILE="experiments/${OUTPUT_FILE}_${HOSTNAME}.csv"
NCPU = $1
NGPU = $2
SIZE = $3

echo "NCPU,NGPU,N,NB,t" > ${OUTPUT_FILE}

for i in 128 256 512 1024 2048 4096
do
    for j in 1 2 3 4 5
    do
        echo "Test ${j}/5 with block size ${i}"
        ./main ${NCPU} ${NGPU} ${SIZE} ${i} >> ${OUTPUT_FILE}
    done
done
```

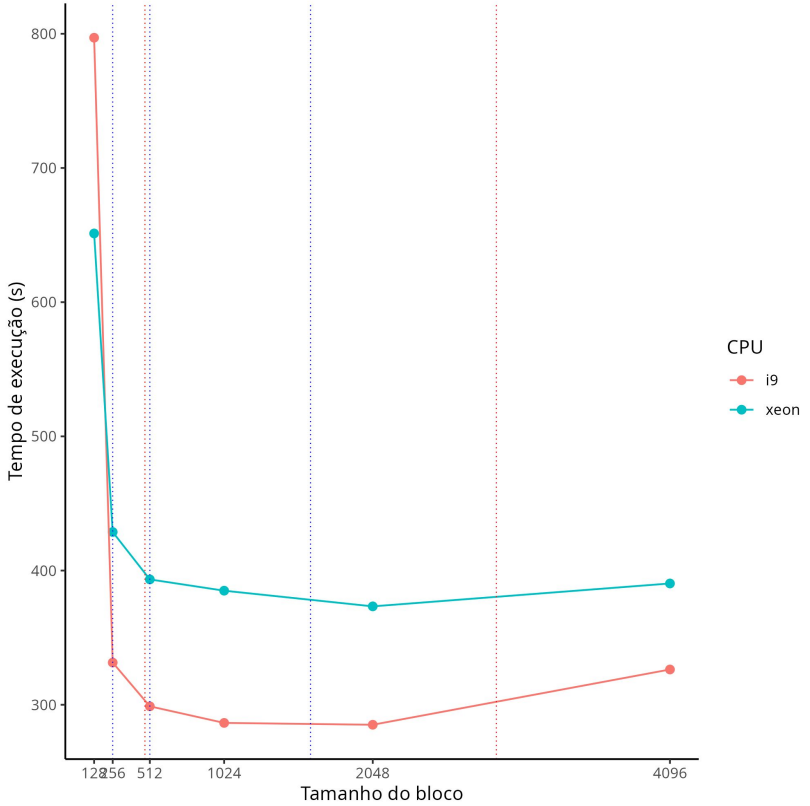
Teste 1 - sem acelerador

- Testando sem a GPU para buscar o tamanho de bloco mais adequado para cada processador, levando em conta que o *Intel Core i9-14900KF* possui 24 núcleos físicos e o *Intel Xeon E5-2620 v4* apenas 8

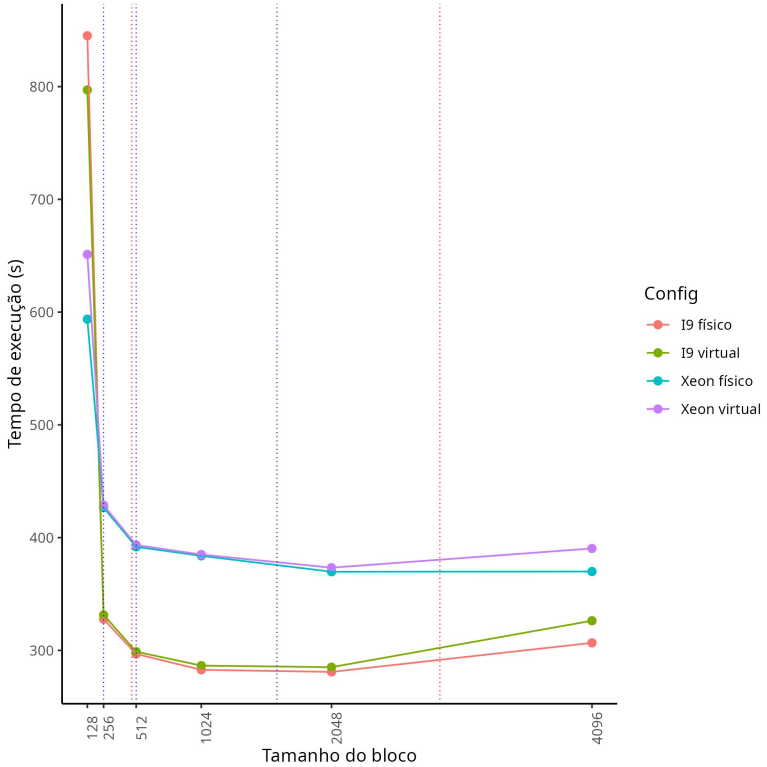


Teste 2 - sem acelerador, usando núcleos lógicos

- Similar ao teste anterior, porém utilizando os 16 (no caso do *Intel Xeon E5-2620 v4*) e 32 (no caso do *Intel Core i9-14900KF*) núcleos disponíveis



Teste 3 - Comparando todos os testes sem acelerador



Análises futuras

- Testes com acelerador usando o máximo disponível de cada processador
- Testes com acelerador e limitado pelo número de processos do processador mais fraco
- Testes com acelerador e limitado pela memória *cache* do processador mais fraco
- Testes com acelerador e usando o máximo disponível em cada processador
- Testes com acelerador e buscando a utilização máxima do mesmo
- Produção dos rastros de execução

Obrigado pela atenção!

Dúvidas?