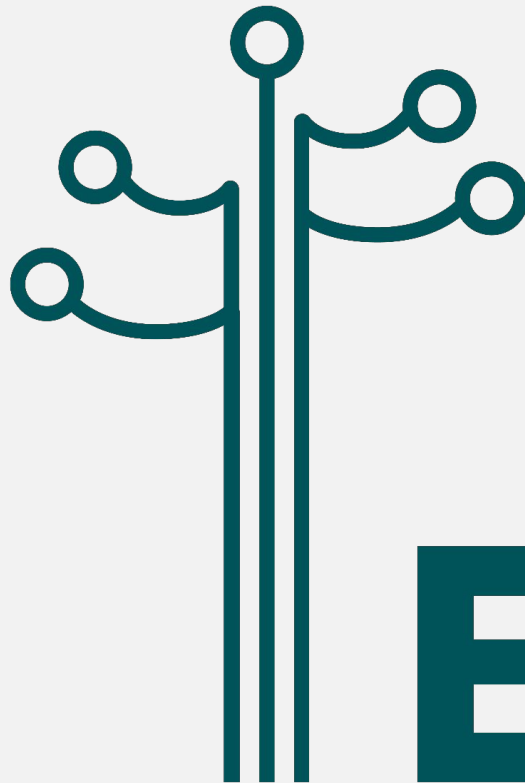


OPRP-MiniCFD

Eliton Machado da Silva
08/12/2022

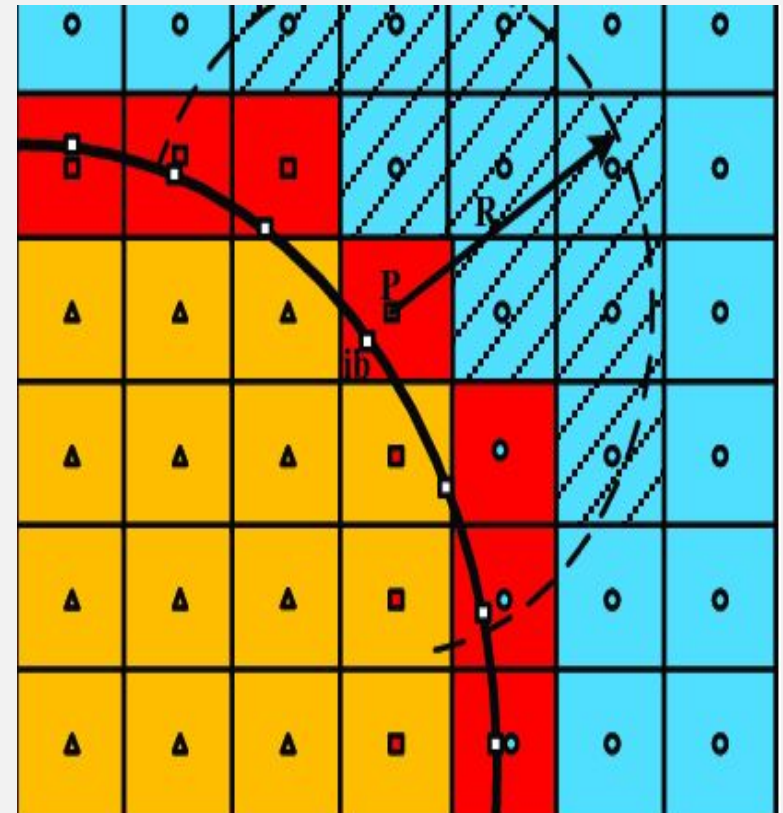
Problema da Maratona de Programação Paralela da ERAD/RS 2021



ERAD | **RS**
20
22

A simulação consiste na inserção de fluxo no centro
de um stencil neutro

- Computa Massa
- Computa Energia gasta



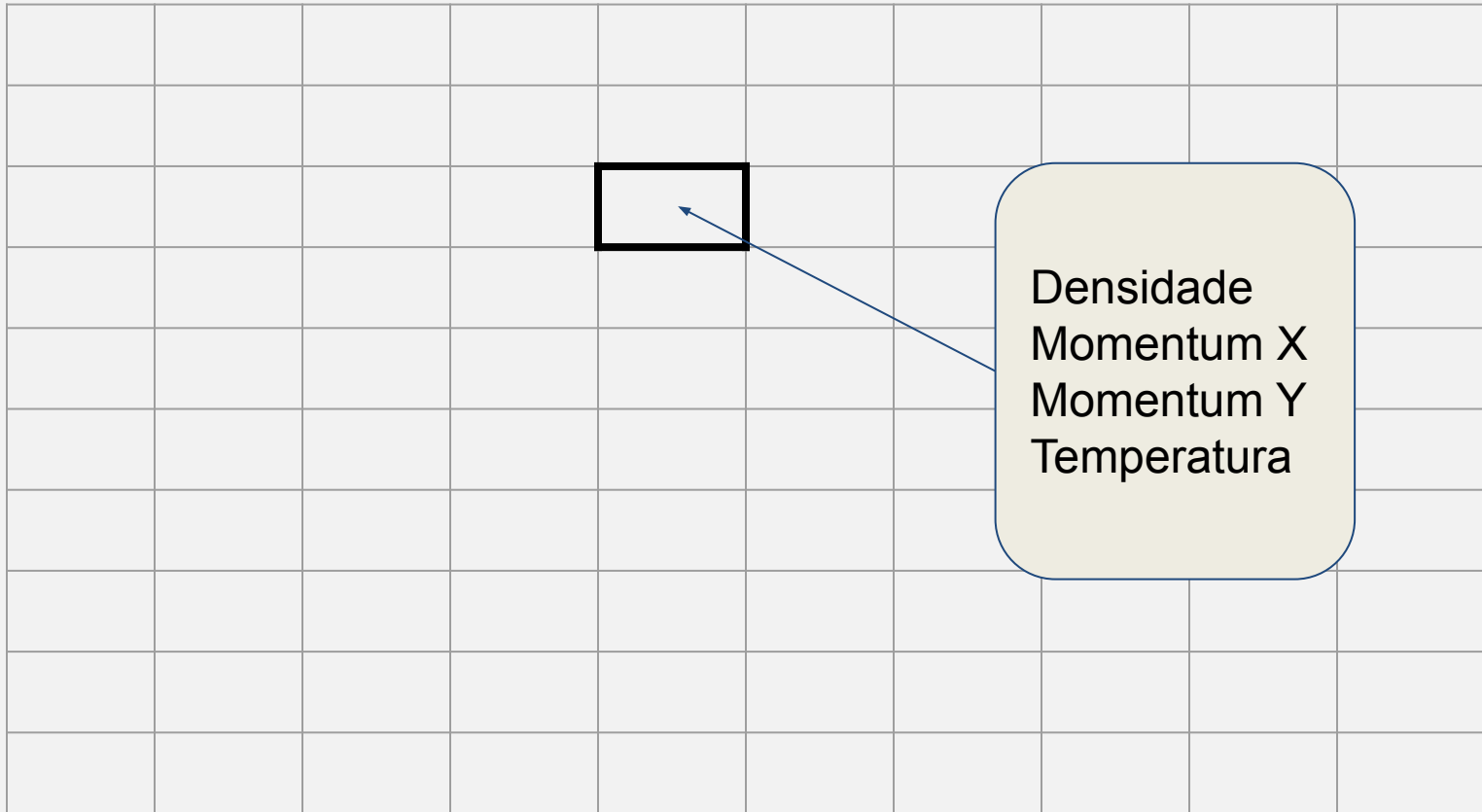
Implementação do problema

- Computar Massa e Energia Gasta
- Simular o comportamento do fluido após 100 unidades de tempo
- Passos de 0.08 u.t.
- Computar o estado da simulação ao longo do tempo
- O estado **S+1** depende do estado **S**

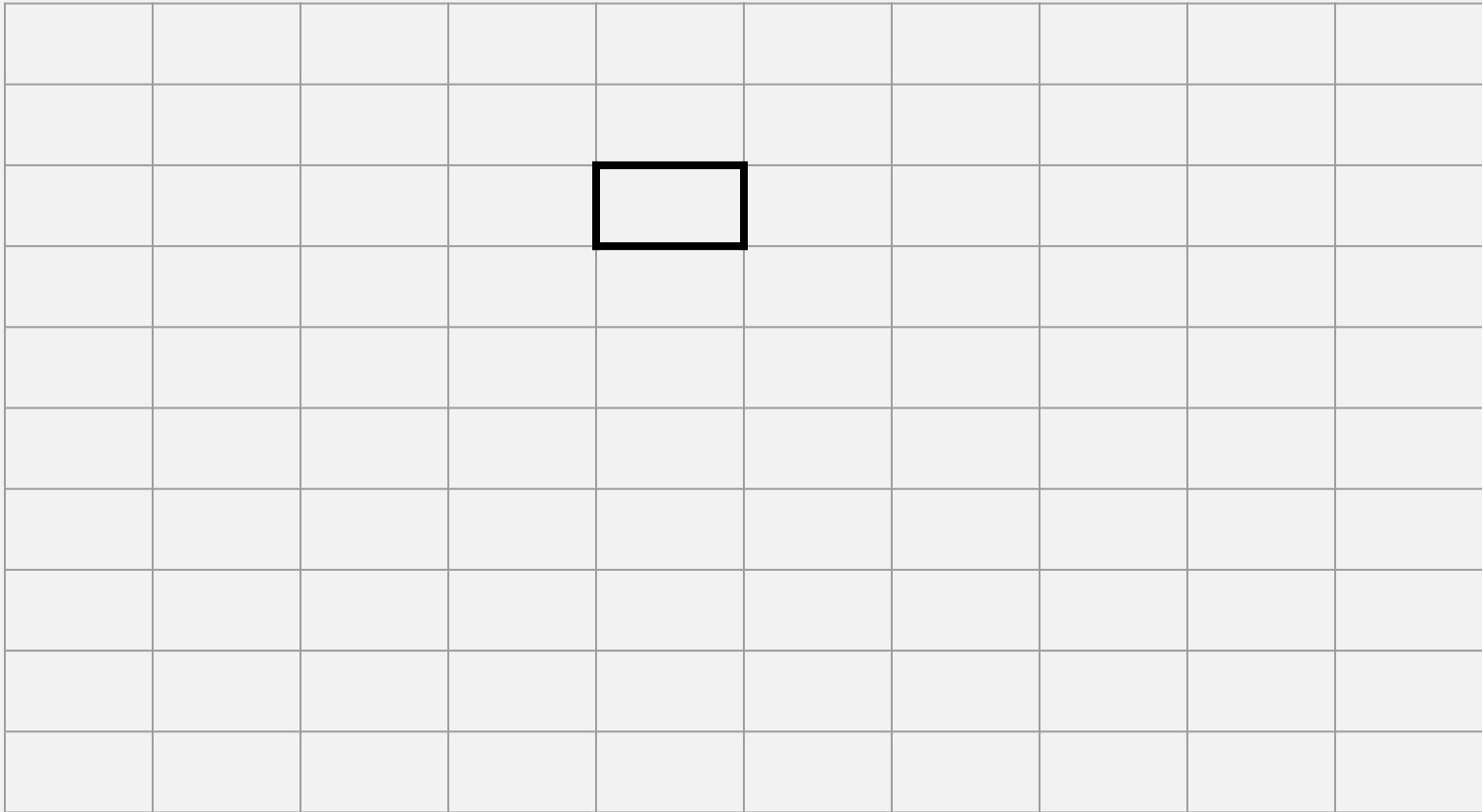
Estado da simulação

[illegible]

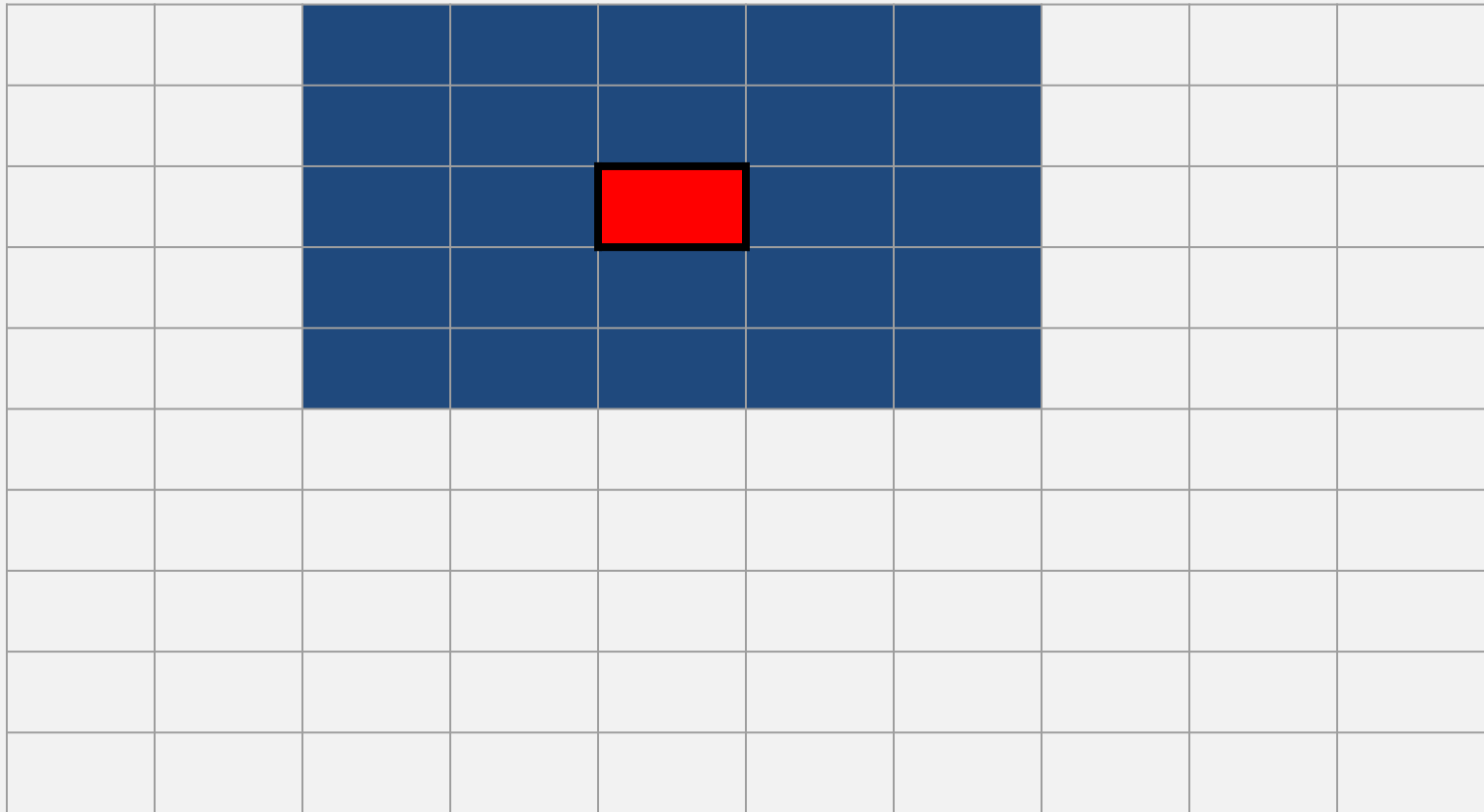
Estado da simulação



Calculando o estado $S+1$



Calculando o estado S+1



Paralelização

- Openmp
- Mpi
- Mpi + Openmp
- OpenAcc
- Mpi + OpenAcc

Otimização OPENMP

```
//Apply the tendencies to the fluid state
#pragma omp parallel for
for (int ll=0; ll<NUM_VARS; ll++) {
    for (int k=0; k<nnz; k++) {
        for (int i=0; i<nnx; i++) {
            int inds = (k+hs)*(nnx+2*hs) + ll*(nnz+2*hs)*(nnx+2*hs) + i+hs;
            int indt = ll*nnz*nnx + k*nnx + i;
            state_out[inds] = state_init[inds] + dt * tend[indt];
        }
    }
}
```

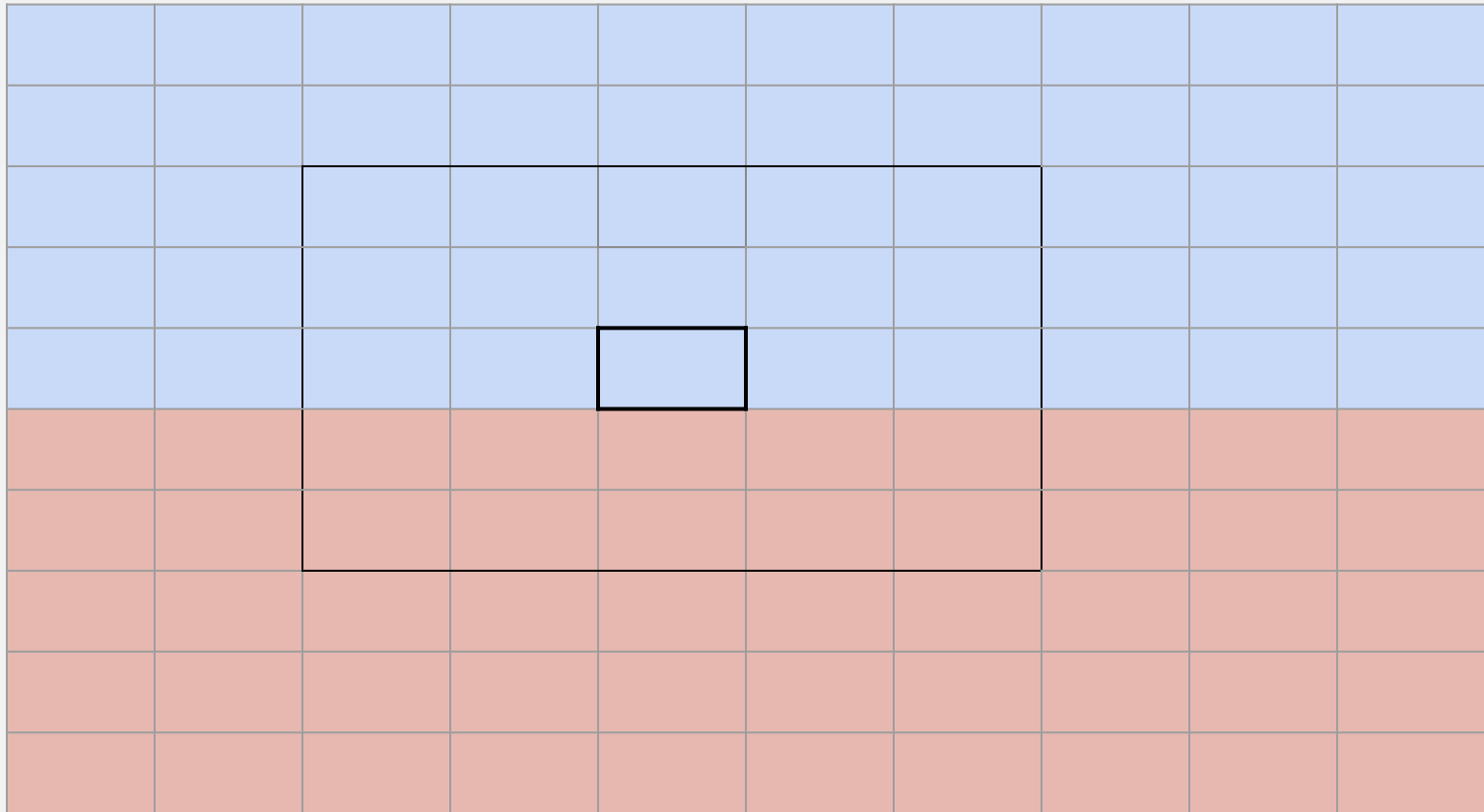
- 10 trechos triviais

Otimização OPENMP

```
//Compute fluxes in the x-direction for each cell  
double d_vals[NUM_VARS], vals[NUM_VARS];  
#pragma omp parallel for private(vals, d_vals)  
for (int k=0; k<nnz; k++) {  
    for (int i=0; i<nnx+1; i++) {  
        ...  
    }  
}
```

Otimização MPI

Otimização MPI



Otimização MPI

```
int sz = hs * NUM_VARS * nnz;
for (int i = 0; i < hs; i++) {
    for (int ll=0; ll<NUM_VARS; ll++) {
        for (int k=0; k<nnz; k++) {
            int idx = i*NUM_VARS*nnz + ll*nnz + k;
            int pos = ll*(nnz+2*hs)*(nnx+2*hs) + (k+hs)*(nnx+2*hs) + nnx + i;
            mpistate[idx] = state[pos];
        }
    }
}

MPI_Send(mpistate, sz, MPI_DOUBLE, ngbr, 0, MPI_COMM_WORLD);
MPI_Recv(mpistate, sz, MPI_DOUBLE, ngbl, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);

for (int i = 0; i < hs; i++) {
    for (int ll=0; ll<NUM_VARS; ll++) {
        for (int k=0; k<nnz; k++) {
            int idx = i*NUM_VARS*nnz + ll*nnz + k;
            int pos = ll*(nnz+2*hs)*(nnx+2*hs) + (k+hs)*(nnx+2*hs) + i;
            state[pos] = mpistate[idx];
        }
    }
}
```

Otimização OpenAcc

- O código parece se beneficiar muito da arquitetura SIMD
- Performance inicial pior que sequencial

Otimização OpenAcc

- O código parece se beneficiar muito da arquitetura SIMD
- Performance inicial pior que sequencial

Otimização OpenAcc

```
#pragma acc data copyin(state_init, tend), copy(state_out)
#pragma acc parallel loop
for (int ll=0; ll<NUM_VARS; ll++) {
    for (int k=0; k<nnz; k++) {
        for (int i=0; i<nnx; i++) {
            int inds = (k+hs)*(nnx+2*hs) + ll*(nnz+2*hs)*(nnx+2*hs) + i+hs;
            int indt = ll*nnz*nnx + k*nnx + i;
            state_out[inds] = state_init[inds] + dt * tend[indt];
        }
    }
}
```

- Muita cópia de memória

Otimização OpenAcc

```
#pragma acc enter data copyin(state, state_tmp, cfd_dens_cell,  
cfd_dens_theta_cell, cfd_dens_int,  
cfd_dens_theta_int,cfd_pressure_int), create(flux, tend)  
while (etime < sim_time) {  
    ...  
    //Perform a single time step  
    do_timestep(state,state_tmp,flux,tend,dt);  
    ...  
}  
#pragma acc exit data copyout(state)
```

Otimização OpenAcc

Otimizações visando a arquitetura simd

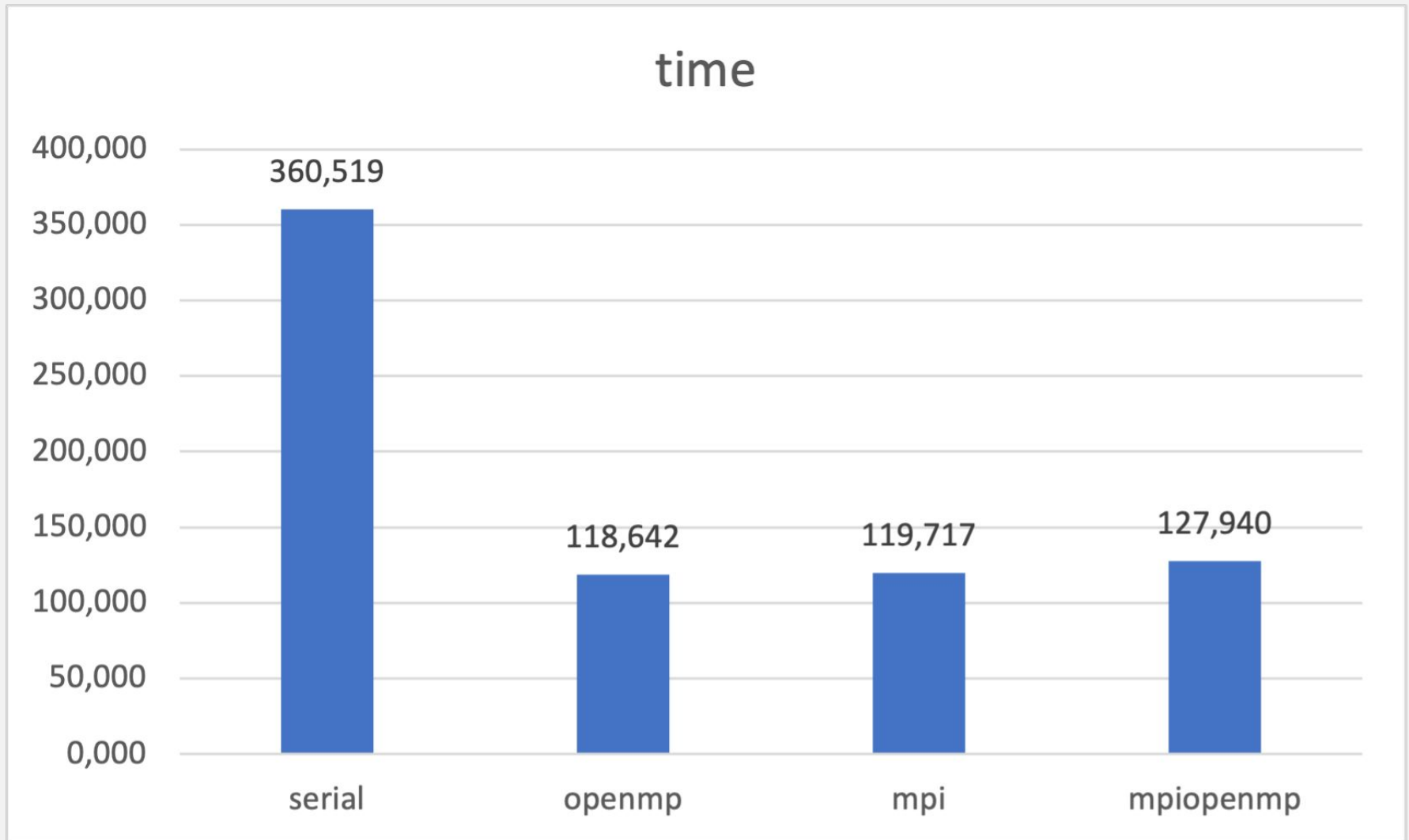
- Loop unroll
- Remoção de ifs

Resultados

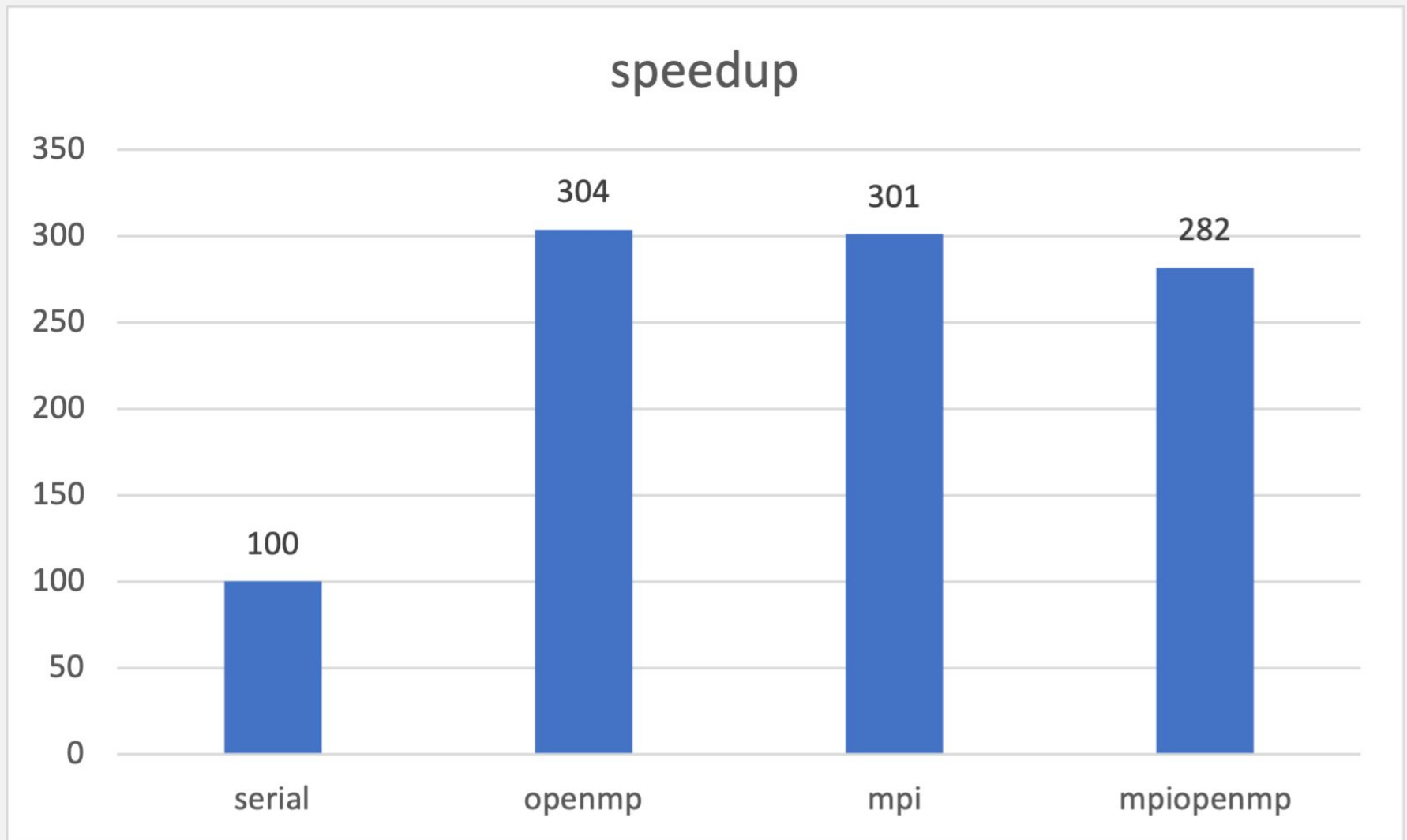
Testes em duas arquiteturas

- Ens5oprp
- Laboratório F307

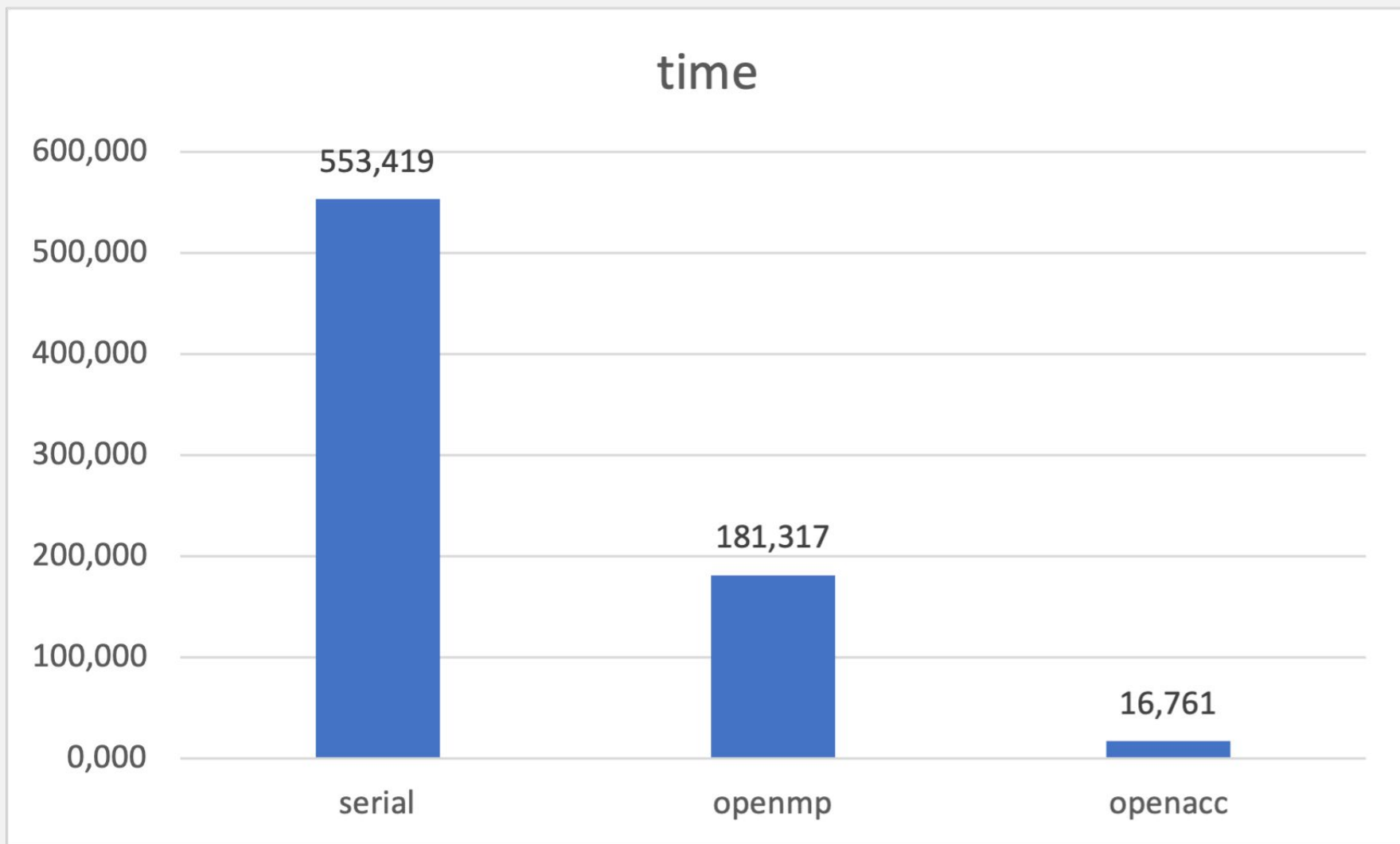
Laboratório F307



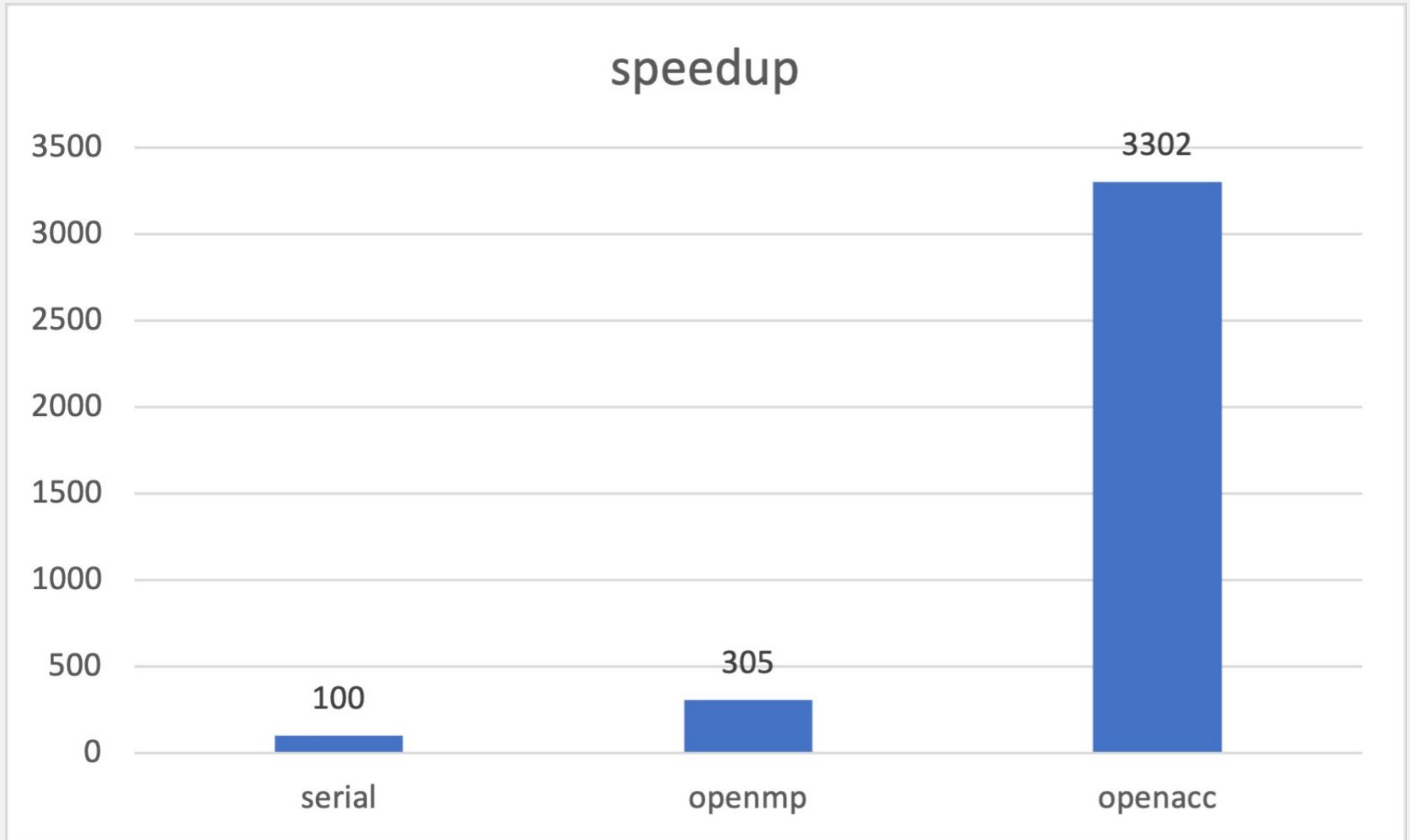
Laboratório F307



Ens5opr



Ens5opr





Obrigado

**UDESC – Universidade do Estado de
Santa Catarina**

eliton.mds@edu.udesc.br

<https://github.com/EMachad0>