



Московский Государственный Университет им. М.В. Ломоносова
факультет Вычислительной математики и кибернетики
кафедра Суперкомпьютеров и Квантовой информатики



Возможности пакета TaiChi для математического моделирования на современных вычислительных архитектурах

Выполнила:
студентка группы 538, Ши Хуэй

Руководитель:
доцент кафедры СКИ, к.т.н., Русол А.В.

X

Nstep

N

$$-2^{31} \leq \text{Index} < 2^{31} - 1$$

X_sum

Nstep

N

		...	
		...	
		...	
		...	
		...	
		...	

X

--	--	--	--	--

```

@ti.kernel
def substep(i:int):
    for j in range(N - 1):
        dx[j] = (x[j + 1, i] - x[j, i]) - l

    for j in range(N):
        left = -C * dx[j - 1] if j > 0 else 0.0
        right = C * dx[j] if j < N - 1 else 0.0

        contact = -Cs * x[j, i] if j == 0 and x[j, i] < 0.0 else 0.0

        F[j] = left + right - B * vx[j, i] + contact - m * g

    for j in range(N):
        a = F[j] / m
        vx[j, i + 1] = vx[j, i] + a * dt
        x[j, i + 1] = x[j, i] + vx[j, i + 1] * dt

```

```

@ti.kernel
def substep():
    for j in range(N - 1):
        dx[j] = x1[j + 1] - x1[j] - l

    for j in range(N):
        left = -C * dx[j - 1] if j > 0 else 0.0
        right = C * dx[j] if j < N - 1 else 0.0

        contact = -Cs * x1[j] if j == 0 and x1[j] < 0.0 else 0.0

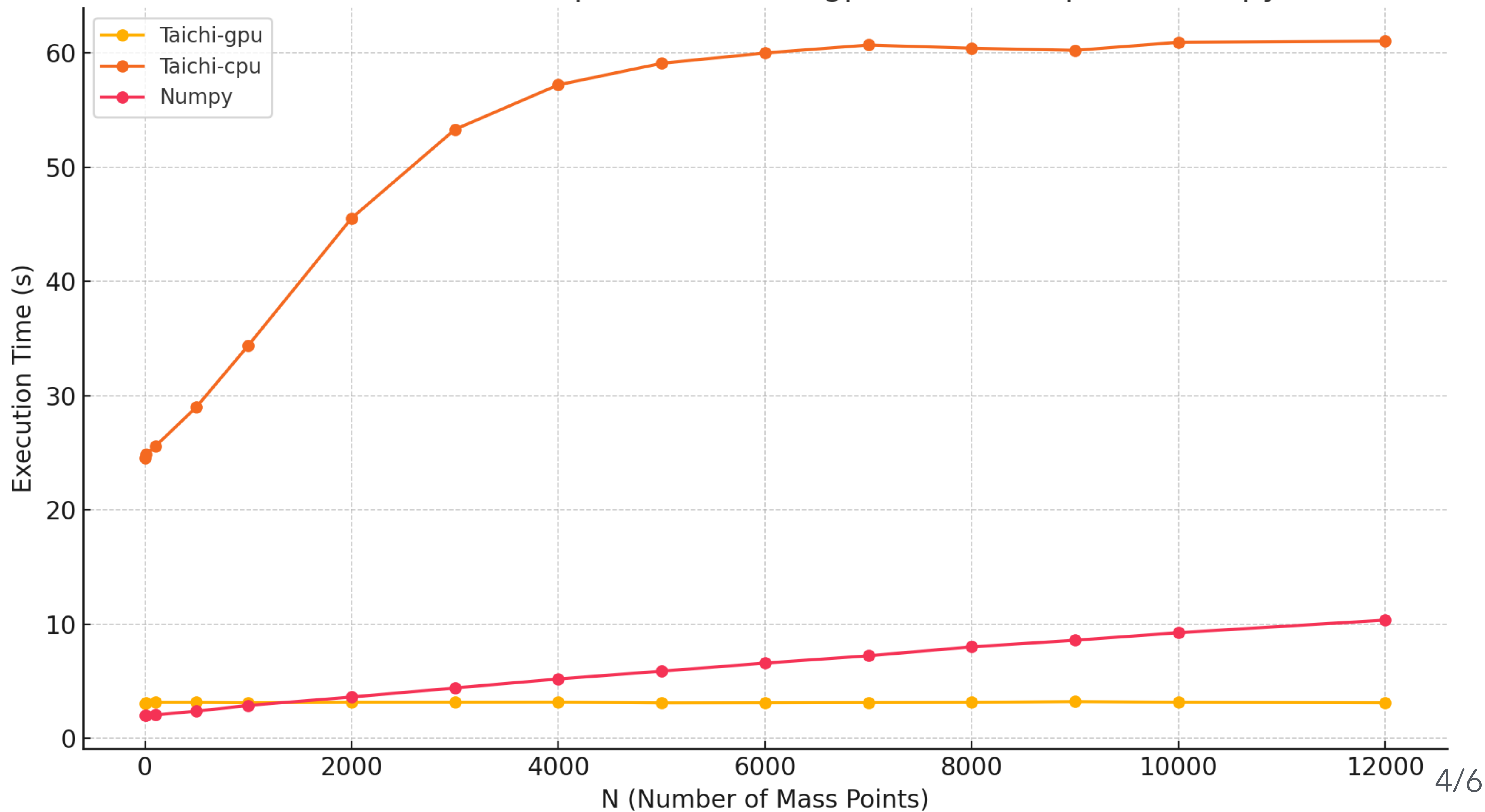
        F[j] = left + right - B * v1[j] + contact - m * g

    for j in range(N):
        a = F[j] / m
        v2[j] = v1[j] + a * dt
        x2[j] = x1[j] + v2[j] * dt

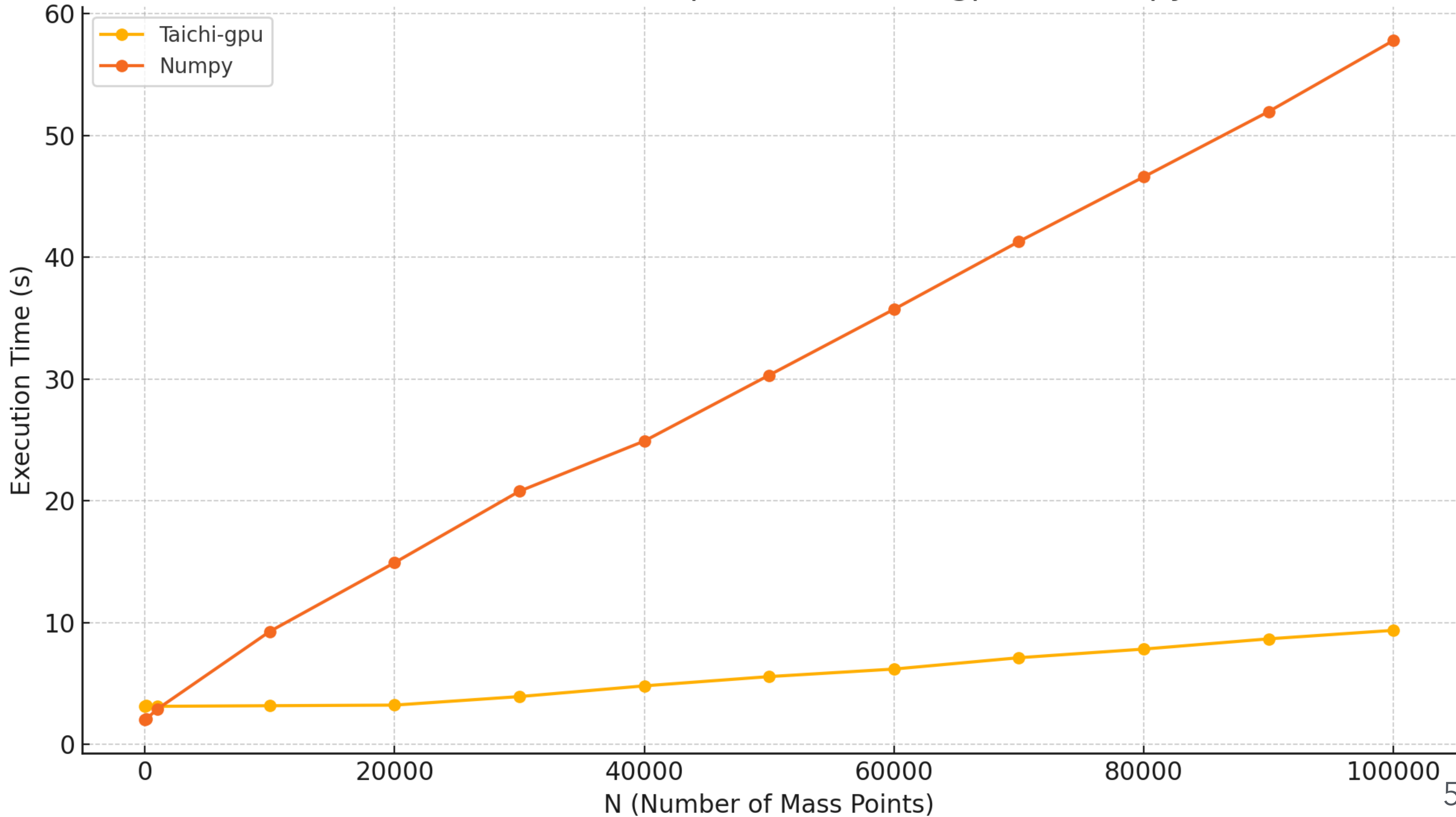
    for j in range(N):
        x1[j] = x2[j]
        v1[j] = v2[j]

```

Execution Time Comparison (Taichi-gpu vs Taichi-cpu vs Numpy)



Execution Time Comparison - Taichi-gpu vs Numpy



Спасибо за внимание