

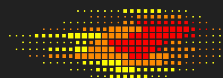
WSCAD 2023

XXIV Simpósio em Sistemas Computacionais de Alto Desempenho

17 a 20 de outubro, 2023 — Porto Alegre, Brasil

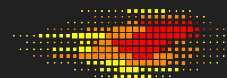
Ensino de Software Pipelining e Escalonamento em GPUs com Python no Google Colab

Ricardo Ferreira, José A. Nacif
Universidade Federal de Viçosa



Sumário

- Python, Jupyter Notebooks e Google Colab
- Software Pipelining
- Escalonamento em GPU
- Considerações Finais



Python

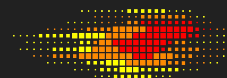
- ipython = Interatividade
- Interesse dos estudantes de Computação
- Muitos exemplos, facilidades de prototipação, Chatgpt, etc....



Google Colab

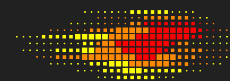
- Documentação (texto+gráfica), Código, Interatividade
- [Acesso e Configuração fácil, Gratuito e Colaborativo](#)
- Rápido ciclo de desenvolvimento e teste





Software Pipelining

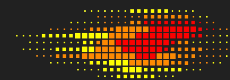
- Ferramentas: Unrolling e Escalonamento Dinâmico...porém **não** existem ferramentas para ensino de **Software Pipelining**
- Complexidade para validação e inversão do código
- Desafios de balanceamento



Exemplo Loop para Vetor e o código Assembly

$$v[i] = (v[i] + v[i])^2$$

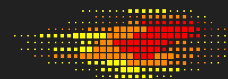
Codigo	F	D	E	W	F	D	E	W
Ld f3,0(r1)	1	2	34	5	15	16	1718	19
add f3,f3,f3	2	34	567	8	16	1718	19-21	22
mult f3,f3,f3	34	567	8-12	13				
sd f3,0(r1)	567	8-12	1314	-				
Addi r1,r1,8	8-12	13	14	15				
Beq r1,r2,LOOP	13	14						



Exemplo Loop para Vetor e o código Assembly

$$v[i] = (v[i] + v[i])^2$$

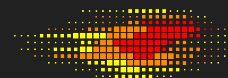
Codigo	F	D	E	W	F	D	E	W
Ld f3,0(r1)	1	2	34	5	15	16	1718	19
add f3,f3,f3	2	34	567	8	16	1718	19-21	22
mult f3,f3,f3	34	567	8-12	13				
sd f3,0(r1)	567	8-12	1314	-				
Addi r1,r1,8	8-12	13	14	15				
Beq r1,r2,LOOP	13	14						



Exemplo Loop para Vetor e o código Assembly

$$v[i] = (v[i] + v[i])^2$$

Codigo	F	D	E	W	F	D	E	W
Ld f3,0(r1)	1	2	34	5	15	16	1718	19
add f3,f3,f3	2	34	567	8	16	1718	19-21	22
mult f3,f3,f3	34	567	8-12	13				
sd f3,0(r1)	567	8-12	1314	-				
Addi r1,r1,8	8-12	13	14	15				
Beq r1,r2,LOOP	13	14						

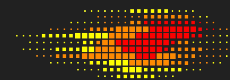


Exemplo Loop para Vetor e o código Assembly

$$v[i] = (v[i] + v[i])^2$$

Codigo	F	D	E	W	F	D	E	W
Ld f3,0(r1)	1	2	34	5	15	16	1718	19
add f3,f3,f3	2	34	567	8	16	1718	19-21	22
mult f3,f3,f3	34	567	8-12	13				
sd f3,0(r1)	567	8-12	1314	-				
Addi r1,r1,8	8-12	13	14	15				
Beq r1,r2,LOOP	13	14						

2 ciclos para Load
3 ciclos para Add
5 ciclos para Mult



Loop Original

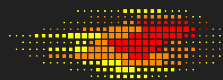
Codigo	F	D	E	W	F	D	E	W
Ld f3,0(r1)	1	2	34	5	15	16	1718	19
add f3,f3,f3	2	34	567	8	16	1718	19-21	22
mult f3,f3,f3	34	567	8-12	13				
sd f3,0(r1)	567	8-12	1314	-				
Addi r1,r1,8	8-12	13	14	15				
Beq r1,r2,LOOP	13	14						

14 ciclos

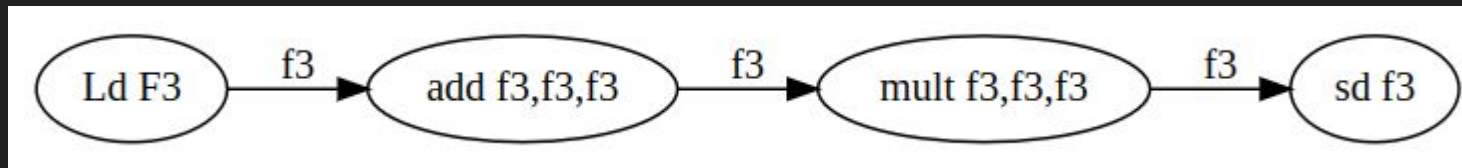
Software Pipelining

Codigo	F	D	E	W	F	D	E	W
sd f3,0(r1)	1	2	34	-	9	10	1112	-
mult f3,f4,f4	2	3	4-8	9	10	11	12-16	17
add f4,f5,f5	3	4	567	8	11	12	13-15	16
Addi r1,r1,8	4	5	6	7	12	13	14	15
Ld f5,0(r1)	5	67	89	10	13	1415	1617	18
Beq r1,r2,LOOP	67	8						

7 ciclos

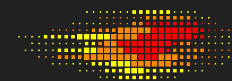


Software Pipelining

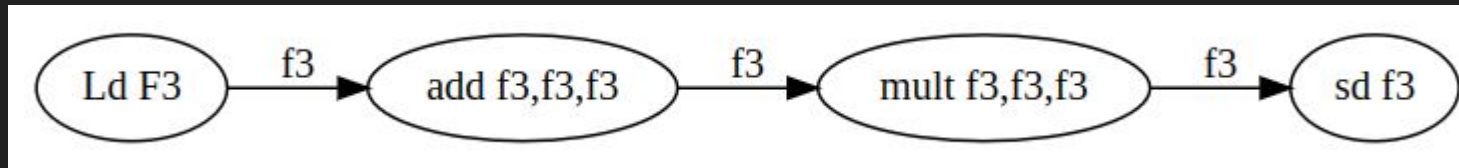


1. Executar em Python -> Resultado
2. Grafo de dependência
3. Renomear Registradores
4. Preâmbulo e Epílogo
5. Balancear Caminhos
6. Validar a execução em Python

Execução do loop original em Python

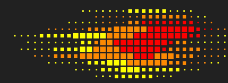


WSCAD 2023

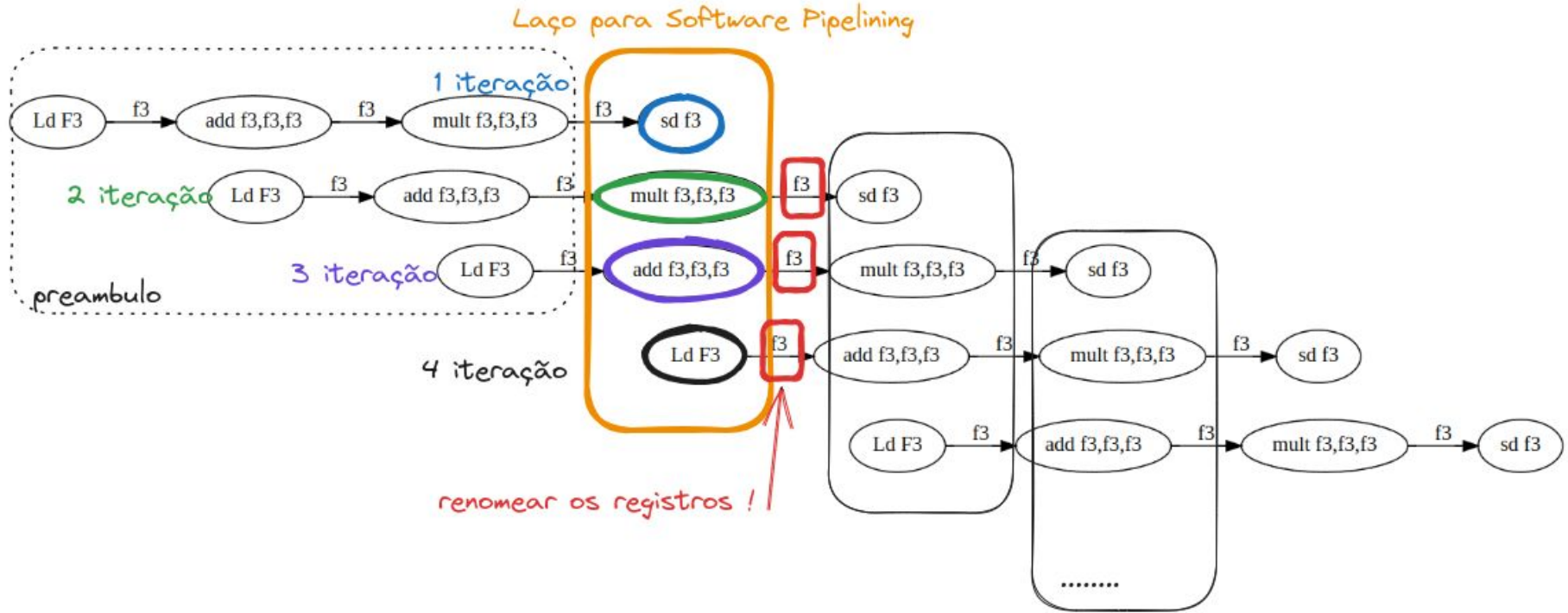


```
10 memory = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99]
11
12 for i in range(20):
13     f3 = memory[i]
14     f3 = f3+f3
15     f3 = f3*f3
16     memory[i] = f3
17
18 print(memory)
```

```
[0, 4, 16, 36, 64, 100, 144, 196, 256, 324, 400, 484, 576,
```



WSCAD 2023



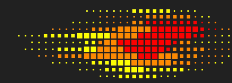
Original



```
10 memory = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99]
11
12 for i in range(20):
13     f3 = memory[i]
14     f3 = f3+f3
15     f3 = f3*f3
16     memory[i] = f3
17
18 print(memory)
```

[0, 4, 16, 36, 64, 100, 144, 196, 256, 324, 400, 484, 576, 676, 784, 896, 1020, 1156, 1304, 1464, 1636]

Software Pipelining



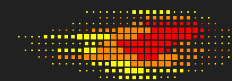
WSCAD 2023



```
12 f5= memory[0]
13 f4 = f5+f5
14 f3= f4*f4
15 f5= memory[1]
16 f4 = f5+f5
17 f5= memory[2]
18 for i in range(15):
19     memory[i] = f3 # sd f3
20     f3 = f4*f4 # mul f3,f4,f4
21     f4 = f5+f5 # add f4,f5,f5
22     f5 = memory[i+3] # ld f5,"12"(r1)
23
24 print(memory)
```

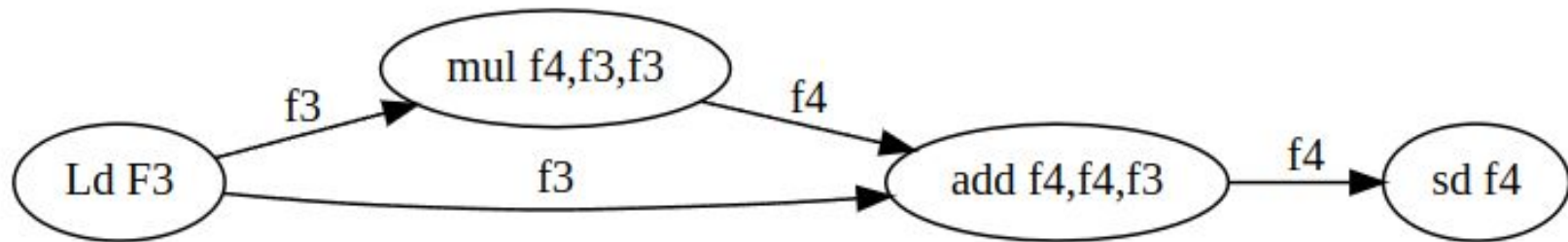
[0, 4, 16, 36, 64, 100, 144, 196, 256, 324, 400, 484, 576, 676, 784, 896, 1020, 1156, 1304, 1464, 1636, 1824, 2024, 2236, 2456, 2684, 2924, 3176, 3440, 3716, 4004, 4304, 4616, 4940, 5276, 5624, 5984, 6356, 6740, 7136, 7544, 7964, 8396, 8840, 9296, 9764, 10244, 10736, 11240, 11756, 12284, 12824, 13376, 13940, 14516, 15104, 15704, 16316, 16940, 17576, 18224, 18884, 19556, 20240, 20936, 21644, 22364, 23096, 23840, 24596, 25364, 26144, 26936, 27740, 28556, 29384, 30224, 31076, 31940, 32816, 33704, 34604, 35516, 36440, 37376, 38324, 39284, 40256, 41240, 42236, 43244, 44264, 45296, 46340, 47396, 48464, 49544, 50636, 51740, 52856, 53984, 55124, 56276, 57440, 58616, 59804, 60996, 62192, 63392, 64596, 65804, 67024, 68256, 69492, 70732, 71976, 73224, 74476, 75732, 76992, 78256, 79524, 80796, 82072, 83352, 84636, 85924, 87216, 88512, 89812, 91116, 92424, 93736, 95052, 96372, 97696, 99024, 100356, 101692, 103032, 104376, 105724, 107076, 108432, 109792, 111156, 112524, 113896, 115272, 116652, 118036, 119424, 120816, 122212, 123612, 125016, 126424, 127836, 129252, 130672, 132096, 133524, 134956, 136392, 137832, 139276, 140724, 142176, 143632, 145092, 146556, 148024, 149496, 150972, 152452, 153936, 155424, 156916, 158412, 159912, 161416, 162924, 164436, 165952, 167472, 168996, 170524, 172056, 173592, 175132, 176676, 178224, 179776, 181332, 182892, 184456, 186024, 187596, 189172, 190752, 192336, 193924, 195516, 197112, 198712, 200316, 201924, 203536, 205152, 206772, 208396, 210024, 211656, 213292, 214932, 216576, 218224, 219876, 221532, 223192, 224856, 226524, 228196, 229872, 231552, 233236, 234924, 236616, 238312, 240012, 241716, 243424, 245136, 246852, 248572, 250296, 252024, 253756, 255492, 257232, 258976, 260724, 262476, 264232, 265992, 267756, 269524, 271296, 273072, 274852, 276636, 278424, 280216, 282012, 283812, 285616, 287424, 289236, 291052, 292872, 294696, 296524, 298356, 300192, 302032, 303876, 305724, 307576, 309432, 311292, 313156, 315024, 316896, 318772, 320652, 322536, 324424, 326316, 328212, 330112, 332016, 333924, 335836, 337752, 339672, 341596, 343524, 345456, 347392, 349332, 351276, 353224, 355176, 357132, 359092, 361056, 363024, 364996, 366972, 368952, 370936, 372924, 374916, 376912, 378912, 380916, 382924, 384936, 386952, 388972, 390996, 393024, 395056, 397092, 399132, 401176, 403224, 405276, 407332, 409392, 411456, 413524, 415596, 417672, 419752, 421836, 423924, 426016, 428112, 430212, 432316, 434424, 436536, 438652, 440772, 442896, 445024, 447156, 449292, 451432, 453576, 455724, 457876, 460032, 462192, 464356, 466524, 468696, 470872, 473052, 475236, 477424, 479616, 481812, 484012, 486216, 488424, 490636, 492852, 495072, 497296, 499524, 501756, 503992, 506232, 508476, 510724, 512976, 515232, 517492, 519756, 522024, 524296, 526572, 528852, 531136, 533424, 535716, 538012, 540312, 542616, 544924, 547236, 549552, 551872, 554196, 556524, 558856, 561192, 563532, 565876, 568224, 570576, 572932, 575292, 577656, 580024, 582396, 584772, 587152, 589536, 591924, 594316, 596712, 599112, 601516, 603924, 606336, 608752, 611172, 613596, 616024, 618456, 620892, 623332, 625776, 628224, 630676, 633132, 635592, 638056, 640524, 642996, 645472, 647952, 650436, 652924, 655416, 657912, 660412, 662916, 665424, 667936, 670452, 672972, 675496, 678024, 680556, 683092, 685632, 688176, 690724, 693276, 695832, 698392, 700956, 703524, 706096, 708672, 711252, 713836, 716424, 719016, 721612, 724212, 726816, 729424, 732036, 734652, 737272, 739896, 742524, 745156, 747792, 750432, 753076, 755724, 758376, 761032, 763692, 766356, 769024, 771696, 774372, 777052, 779736, 782424, 785116, 787812, 790512, 793216, 795924, 798636, 801352, 804072, 806796, 809524, 812256, 814992, 817732, 820476, 823224, 825976, 828732, 831492, 834256, 837024, 839796, 842572, 845352, 848136, 850924, 853716, 856512, 859312, 862116, 864924, 867736, 870552, 873372, 876196, 879024, 881856, 884692, 887532, 890376, 893224, 896076, 898932, 901792, 904656, 907524, 910396, 913272, 916152, 919036, 921924, 924816, 927712, 930612, 933516, 936424, 939336, 942252, 945172, 948096, 951024, 953956, 956892, 959832, 962776, 965724, 968676, 971632, 974592, 977556, 980524, 983496, 986472, 989452, 992436, 995424, 998416, 1001412, 1004412, 1007416, 1010424, 1013436, 1016452, 1019472, 1022496, 1025524, 1028556, 1031592, 1034632, 1037676, 1040724, 1043776, 1046832, 1049892, 1052956, 1056024, 1059096, 1062172, 1065252, 1068336, 1071424, 1074516, 1077612, 1080712, 1083816, 1086924, 1090036, 1093152, 1096272, 1099396, 1102524, 1105656, 1108792, 1111932, 1115076, 1118224, 1121376, 1124532, 1127692, 1130856, 1134024, 1137196, 1140372, 1143552, 1146736, 1149924, 1153116, 1156312, 1159512, 1162716, 1165924, 1169136, 1172352, 1175572, 1178796, 1182024, 1185256, 1188492, 1191732, 1194976, 1198224, 1201476, 1204732, 1207992, 1211256, 1214524, 1217796, 1221072, 1224352, 1227636, 1230924, 1234216, 1237512, 1240812, 1244116, 1247424, 1250736, 1254052, 1257372, 1260696, 1264024, 1267356, 1270692, 1274032, 1277376, 1280724, 1284076, 1287432, 1290792, 1294156, 1297524, 1300896, 1304272, 1307652, 1311036, 1314424, 1317816, 1321212, 1324612, 1328016, 1331424, 1334836, 1338252, 1341672, 1345096, 1348524, 1351956, 1355392, 1358832, 1362276, 1365724, 1369176, 1372632, 1376092, 1379556, 1383024, 1386496, 1389972, 1393452, 1396936, 1400424, 1403916, 1407412, 1410912, 1414416, 1417924, 1421436, 1424952, 1428472, 1431996, 1435524, 1439056, 1442592, 1446132, 1449676, 1453224, 1456776, 1460332, 1463892, 1467456, 1471024, 1474596, 1478172, 1481752, 1485336, 1488924, 1492516, 1496112, 1499712, 1503316, 1506924, 1510536, 1514152, 1517772, 1521396, 1525024, 1528656, 1532292, 1535932, 1539576, 1543224, 1546876, 1550532, 1554192, 1557856, 1561524, 1565196, 1568872, 1572552, 1576236, 1579924, 1583616, 1587312, 1591012, 1594716, 1598424, 1602136, 1605852, 1609572, 1613296, 1617024, 1620756, 1624492, 1628232, 1631976, 1635724, 1639476, 1643232, 1646992, 1650756, 1654524, 1658296, 1662072, 1665852, 1669636, 1673424, 1677216, 1681012, 1684812, 1688616, 1692424, 1696236, 1700052, 1703872, 1707696, 1711524, 1715356, 1719192, 1723032, 1726876, 1730724, 1734576, 1738432, 1742292, 1746156, 1750024, 1753896, 1757772, 1761652, 1765536, 1769424, 1773316, 1777212, 1781112, 1785016, 1788924, 1792836, 1796752, 1800672, 1804596, 1808524, 1812456, 1816392, 1820332, 1824276, 1828224, 1832176, 1836132, 1840092, 1844056, 1848024, 1851996, 1855972, 1859952, 1863936, 1867924, 1871916, 1875912, 1879912, 1883916, 1887924, 1891936, 1895952, 1899972, 1903996, 1908024, 1912056, 1916092, 1920132, 1924176, 1928224, 1932276, 1936332, 1940392, 1944456, 1948524, 1952596, 1956672, 1960752, 1964836, 1968924, 1973016, 1977112, 1981212, 1985316, 1989424, 1993536, 1997652, 2001772, 2005896, 2009924, 2014056, 2018192, 2022332, 2026476, 2030624, 2034776, 2038932, 2043092, 2047256, 2051424, 2055596, 2059772, 2063952, 2068136, 2072324, 2076516, 2080712, 2084912, 2089116, 2093324, 2097536, 2101752, 2105972, 2110196, 2114424, 2118656, 2122892, 2127132, 2131376, 2135624, 2139876, 2144132, 2148392, 2152656, 2156924, 2161196, 2165472, 2169752, 2174036, 2178324, 2182616, 2186912, 2191212, 2195516, 2199824, 2204136, 2208452, 2212772, 2217096, 2221424, 2225756, 2230092, 2234432, 2238776, 2243124, 2247476, 2251832, 2256192, 2260556, 2264924, 2269296, 2273672, 2278052, 2282436, 2286824, 2291216, 2295612, 2300012, 2304416, 2308824, 2313236, 2317652, 2322072, 2326496, 2330924, 2335356, 2339792, 2344232, 2348676, 2353124, 2357576, 2362032, 2366492, 2370956, 2375424, 2379896, 2384372, 2388852, 2393336, 2397824, 2402316, 2406812, 2411312, 2415816, 2420324, 2424836, 2429352, 2433872, 2438396, 2442924, 2447456, 2451992, 2456532, 2461076, 2465624, 2470176, 2474732, 2479292, 2483856, 2488424, 2492996, 2497572, 2502152, 2506736, 2511324, 2515916, 2520512, 2525112, 2529716, 2534324, 2538936, 2543552, 2548172, 2552796, 2557424, 2562056, 2566692, 2571332, 2575976, 2580624, 2585276, 2589932, 2594592, 2599256, 2603924, 2608596, 2613272, 2617952, 2622636, 2627324, 2632016, 2636712, 2641412, 2646116, 2650824, 2655536, 2660252, 2664972, 2669696, 2674424, 2679156, 2683892, 2688632, 2693376, 2698124, 2702876, 2707632, 2712392, 2717156, 2721924, 2726696, 2731472, 2736252, 2741036, 2745824, 2750616, 2755412, 2760212, 2765016, 2769824, 2774636, 2779452, 2784272, 2789096, 2793924, 2798756, 2803592, 2808432, 2813276, 2818124, 2822976, 2827832, 2832692, 2837556, 2842424, 2847296, 2852172, 2857052, 2861936, 2866824, 2871716, 2876612, 2881512, 2886416, 2891324, 2896236, 2901152, 2906072, 2910996, 2915924, 2920856, 2925792, 2930732, 2935676, 2940624, 2945576, 2950532, 2955492, 2960456, 2965424, 2970396, 2975372, 2980352, 2985336, 2990324, 2995316, 3000312, 3005312, 3010316, 3015324, 3020336, 3025352, 3030372, 3035396, 3040424, 3045456, 3050492, 3055532, 3060576, 3065624, 3070676, 3075732, 3080792, 3085856, 3090924, 3095996, 3101072, 3106152, 3111236, 3116324, 3121416, 3126512, 3131612, 3136716, 3141824, 3146936, 3152052, 3157172, 3162296, 3167424, 3172556, 3177692, 3182832, 3187976, 3193124, 3198276, 3203432, 3208592, 3213756, 3218924, 3224096, 3229272, 3234452, 3239636, 3244824, 3249916, 3255012, 3260112, 3265216, 3270324, 3275436, 3280552, 3285672, 3290796, 3295924, 3301056, 3306192, 3311332, 3316476, 3321624, 3326776, 3331932, 3337092, 3342256, 3347424, 3352596, 3357772, 3362952, 3368136, 3373324, 3378516, 3383712, 3388912, 3394116, 3399324, 3404536, 3409752, 3414972, 3420196, 3425424, 3430656, 3435892, 3441132, 3446376, 3451624, 3456876, 3462132, 3467392, 3472656, 3477924, 3483196, 3488472, 3493752, 3499036, 3504324, 3509616, 3514912, 3520212, 3525516, 3530824, 3536136, 3541452, 3546772, 3552096, 3557424, 3562756, 3568092, 3573432, 3578776, 3584124, 3589476, 3594832, 3600192, 3605556, 3610924, 3616296, 3621672, 3627052, 3632436, 3637824, 3643216, 3648612, 3654012, 3659416, 3664824, 3670236, 3675652, 3681072, 3686496, 3691924, 3697356, 3702792, 3708232, 3713676, 3719124, 3724576, 3729932, 3735392, 3740856, 3746324, 3751796, 3757272, 3762752, 3768236, 3773724, 3779216, 3784712, 3790212, 3795716, 3801224, 3806736, 3812252, 3817772, 3823296, 3828824, 3834356, 3839892, 3845432, 3850976, 3856524, 3862076, 3867632, 3873192, 3878756, 3884324, 3889896, 3895472, 3901052, 3906636, 3912224, 3917816, 3923412, 3929012, 3934616, 3940224, 3945836, 3951452, 3957072, 3962696, 3968324, 3973956, 3979592, 3985232, 3990876, 3996524, 4002176, 4007832, 4013492, 4019156, 4024824, 4030496, 4036172, 4041852, 4047536, 4053224, 4058916, 4064612, 4070312, 4076016, 4081724, 4087436, 4093152, 4098872, 4104596, 4110324, 4116056, 4121792, 4127532, 4133276, 4139024, 4144776, 4150532, 4156292, 4162056, 4167824, 4173596, 4179372, 4185152, 4190936, 4196724, 4202516, 4208312, 4214112, 4219916, 4225724, 4231536, 4237352, 4243172, 4248996, 4254824, 4260656, 4266492, 4272332, 4278176, 4284024, 4289876, 4295732, 4301592, 4307456, 4313324, 4319196, 4325072, 4330952, 4336836, 4342724, 4348616, 4354512, 4360412, 4366316, 4372224, 4378136, 4384052, 4389972, 4395896, 4401824, 4407756, 4413692, 4419632, 4425576, 4431524, 4437476, 4443432, 4449392, 4455356, 4461324, 4467296, 4473272, 4479252, 4485236, 4491224, 4497216, 4503212, 4509212, 4515216, 4521224, 4527236, 4533252, 4539272, 4545296, 4551324, 4557356, 4563392, 4569432, 4575476, 4581524, 4587576, 4593632, 4599692, 4605756, 4611824, 4617896, 4623972, 4630052, 4636136, 4642224, 4648316, 4654412

Software Pipelining com balanceamento



WSCAD 2023

[]



✓
0s

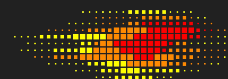


```
1 memory = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24]
2 for i in range(20):
3     f3 = memory[i]
4     f4 = f3*f3
5     f4 = f4+f3
6     memory[i] = f4
7
8 print(memory)
```

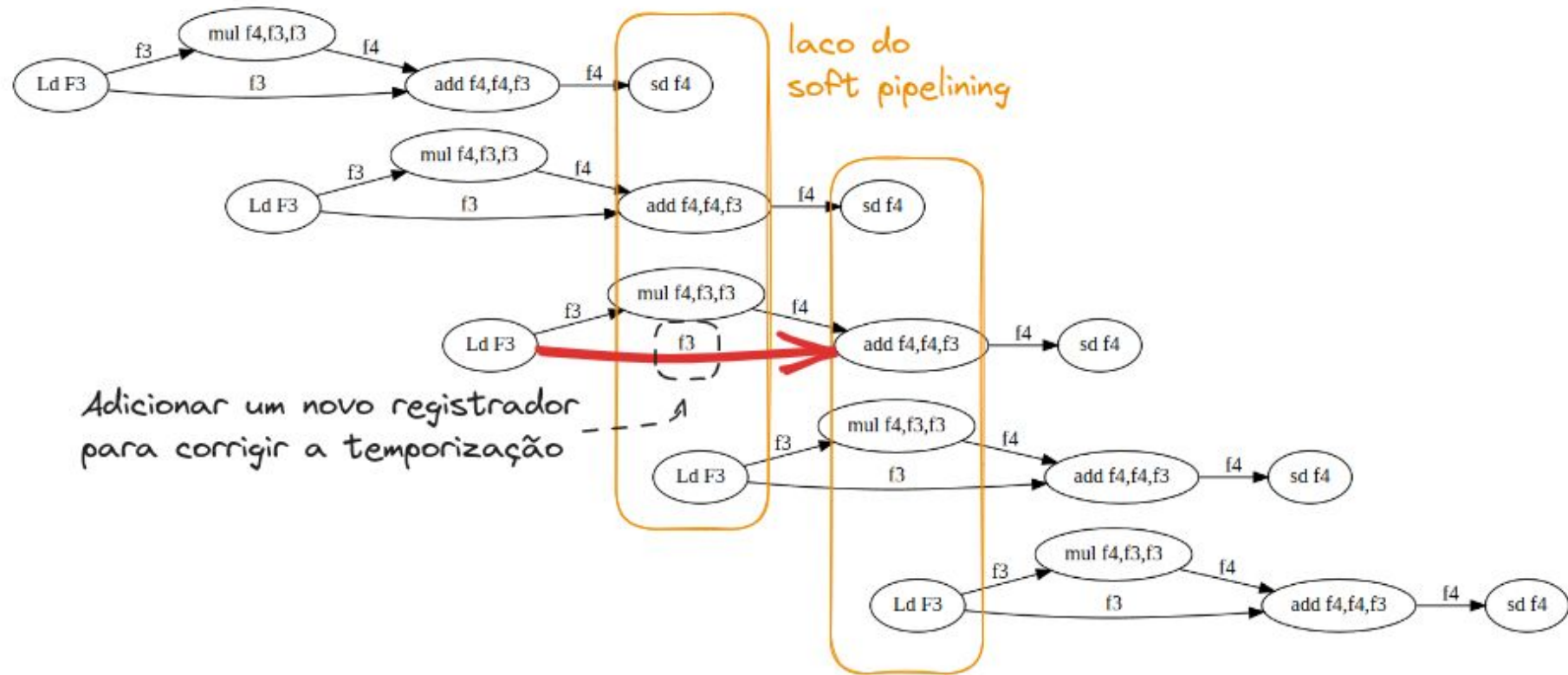
$$v[i] = (v[i] * v[i] + v[i])$$

[0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 182, 210, 240, 272, 306,

Software Pipelining com balanceamento



WSCAD 2023



Original

```
2 for i in range(20):
3     f3 = memory[i]
4     f4 = f3*f3
5     f4 = f4+f3
6     memory[i] = f4
7
8 print(memory)
```

[0, 2, 6, 12, 20, 30, 42, 56, 72, 90,

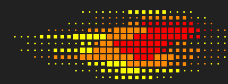
Software Pipelining com balanceamento

WSCAD 2023

```
10 f3 = memory[0]
11 f5 = f3*f3
12 f4 = f5+f3 # 1
13 f3 = memory[1]
14 f5 = f3*f3
15 f6 = f3 #2
16 f3 = memory[2] # 3
17 for i in range(20):
18     memory[i] = f4 # SD
19     f4 = f5+f6 # add f4,f5,f3
20     f5 = f3*f3 # mul f5,f3,f3
21     f6 = f3
22     f3 = memory[i+3] # ld f3
23
24
25 print(memory)
```

[0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156,

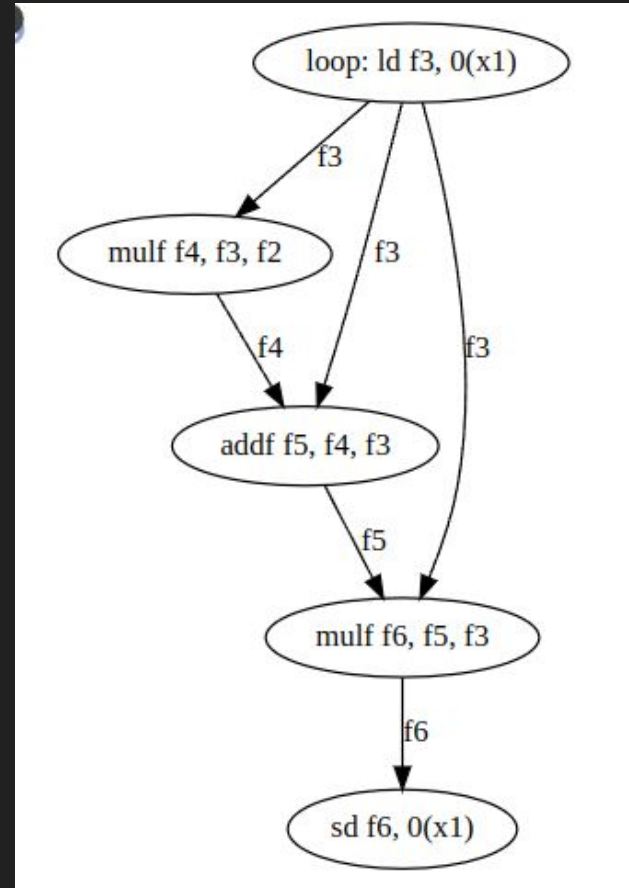
Exercício para os estudantes



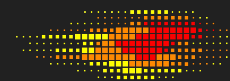
WSCAD 2023

$$v[i] = ((2 * v[i] + v[i]) * v[i])$$

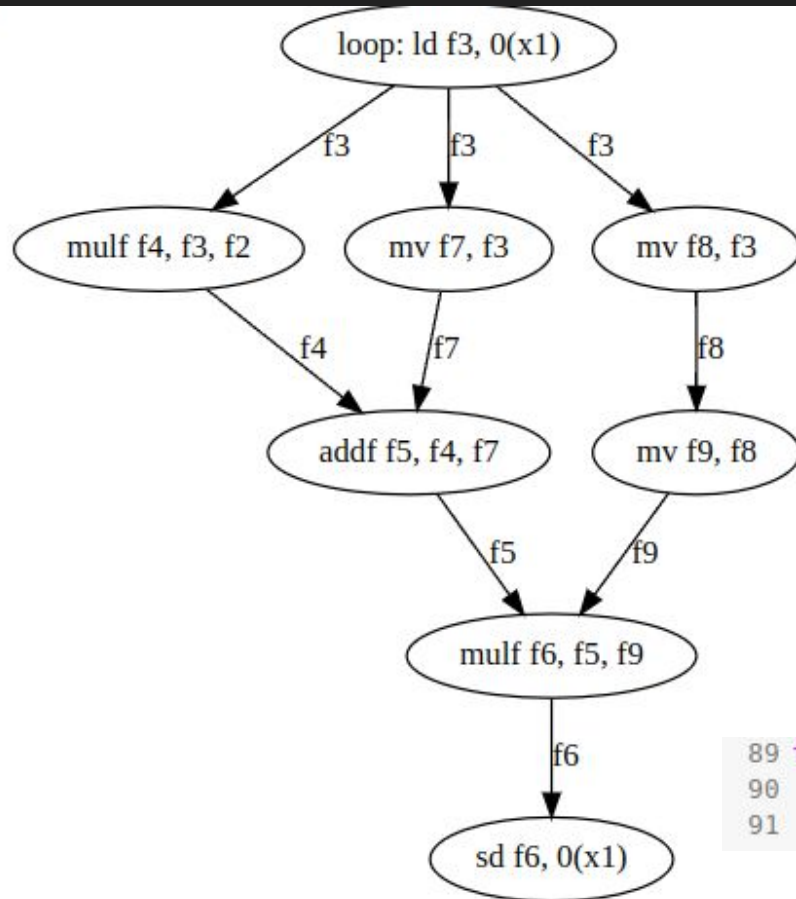
```
4 x1 = 0
5 f2 = 2
6 x5 = T
7 while True:
8     f3 = v[x1] # ld f3, 0(x1)
9     f4 = f3 * f2 # mulf f4, f3, f2
10    f5 = f4 + f3 # addf f5, f4, f3
11    f6 = f5 * f3 # mulf f6, f5, f3
12    v[x1] = f6 # sd f6, 0(x1)
13    x1 = x1 + 1 # addi x1, x1, 4
14    if (x1 == x5): # bne x1, x5, loop
15        break
```



Grafo, Software Pipelining e validação



WSCAD 2023



```
41 while True:
42     l[x1] = f6
43
44     f6 = f5 * f9
45
46     f5 = f4 + f7
47     f9 = f8
48
49     f4 = f3 * f2
50     f7 = f3
51     f8 = f3
52
53     f3 = l[x1 + 4]
54
55     x1 = x1 + 1
56     if (x1 == (x5 - 4)):
57         break
58
```

```
89 for i in range(T):
90     if v[i] != l[i]:
91         print("Wrong calculation in software pipeline at position", i)
```

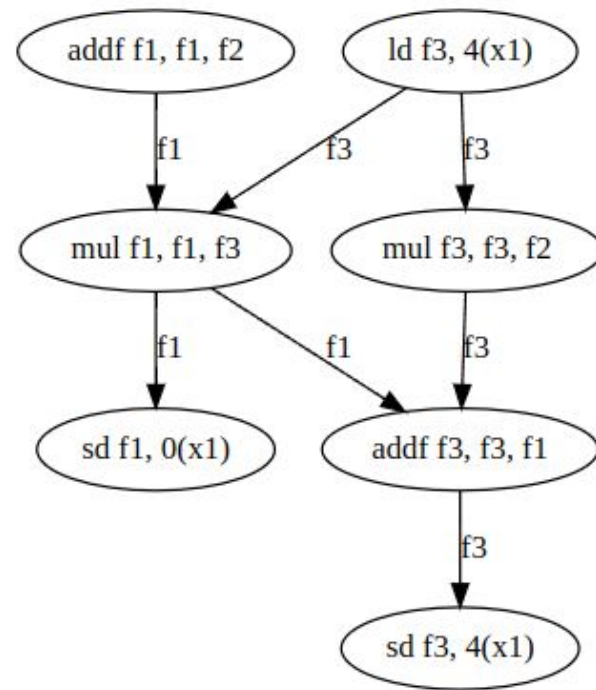
geração do grafo de dependência



WSCAD 2023

```
1 instructions = (  
2   'ld f1, 0(x1)',  
3   'addf f1, f1, f2',  
4   'ld f3, 4(x1)',  
5   'mul f1, f1, f3',  
6   'sd f1, 0(x1)',  
7   'mul f3, f3, f2',  
8   'addf f3, f3, f1',  
9   'sd f3, 4(x1)'  
10 )
```

Parser
e
Gerador do Grafo
(extra)



Versão Simplificada de uma GPU

- Warps com 4 threads
- Banco de Registradores com 64 registros
- Máximo de 32 threads por Multiprocessador
- Máximo de 4 blocos por Multiprocessador
- Máximo de 16 threads por bloco
- Instruções
 - Ld 4 ciclos cache, 20 ciclos memoria
 - Sd 4 ciclos cache, 20 ciclos memoria
 - Add e Mul 3 ciclos
- Ordem de swap dos Warps é aleatória

Planilha CSV para entrada do trace

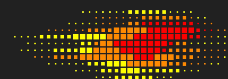


WSCAD 2023

Trecho código a = b + c

```
[ ] 1 #@title Trecho código a = b + c
    2 %%writefile exemplo.csv
    3 PC; Inst; Warp; Fetch; Decode; Exec; W
    4 0; Ld r1; w1; 1; 2; 3-6; 7;
    5 1; Ld r2; w1; 2; 3; 4-7; 8;
    6 0; Ld r1; w2; 3; 4; 5-8; 9;
    7 1; Ld r2; w2; 4; 5; 6-9; 10;
    8 x; nop; ; 5
    9 x; nop; ; 6
   10 2; add r1,r1,r2; w1; 7; 8; 9-11; 12
   11 x; nop; ; 8
   12 2; add r1,r1,r2; w1; 9; 10; 11-13; 14
   13 x; nop; ; 10
   14 3; sd r1; 11; 12; 13-17
   15 x; nop; ; 12
   16 3; sd r1; 13; 14; 15-19
```

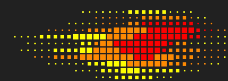

Visualização do Trace



WSCAD 2023

	PC	Inst	Warp	Fetch	Decode	Exec	W
0	0	Ld r1	w1	1	2	3-6	7.0
1	1	Ld r2	w1	2	3	4-7	8.0
2	0	Ld r1	w2	3	4	5-8	9.0
3	1	Ld r2	w2	4	5	6-9	10.0
4	x	nop		5	-	-	-
5	x	nop		6	-	-	-
6	2	add r1,r1,r2	w1	7	8	9-11	12.0
7	x	nop		8	-	-	-
8	2	add r1,r1,r2	w1	9	10	11-13	14.0
9	x	nop		10	-	-	-
10	3	sd r1	11	12	13-17	-	-
11	x	nop		12	-	-	-
12	3	sd r1	13	14	15-19	-	-

Outro formato



WSCAD 2023

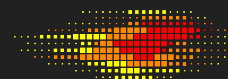
```
2 %%writefile exemplo.csv
3 warp; 1;2;3;4;5;6;7;8;9;0;1;2;3
4 w1; ld; ld; ; ; ; ; add; ; ; ; sd
5 w2; ; ; ld; ld; ; ; ; ; add; ; ; ; sd
```



	warp	1	2	3	4	5	6	7	8	9	0	1	2.1	3.1
0	w1	ld	ld					add				sd	-	-
1	w2			ld	ld					add				sd

[illegible]

Resultados dos Estudantes para Visualização com Html



WSCAD 2023

```
<ipython-input-286-bd4adbe72cb3>:42: FutureWarning: this method is deprecated in favour of `Style`  
display(HTML(styled_df.render()))
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	warp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	Ld r1	Ld r2					nop	add r1,r1,r2			nop	sd r1		
2	2			Ld r1	Ld r2			nop		add r1,r1,r2		nop		sd r1	
3	3					Ld r1	Ld r2	nop			add r1,r1,r2	nop			sd r1

```
<ipython-input-286-bd4adbe72cb3>:42: FutureWarning: this method is deprecated in favour of `Style`  
display(HTML(styled_df.render()))
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	warp	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	Ld r1	Ld r2							nop	add r1,r1,r2				sd r1			
2	2			Ld r1	Ld r2					nop		add r1,r1,r2				sd r1		
3	3					Ld r1	Ld r2			nop			add r1,r1,r2				sd r1	
4	4							Ld r1	Ld r2	nop				add r1,r1,r2				sd r1

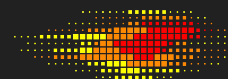
Resultados dos Estudantes para Visualização com cores



WSCAD 2023

	Warp	Inst	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
0	w1	Ld r1	F	D	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	W
1	w1	Ld r2	-	F	D	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
2	w1	Mult r1,r1,r1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F	D	E	E
3	w1	Mult r2,r2,r2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F	D	E
4	w1	Mult r1,r1,r2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	w2	Ld r1	-	-	F	D	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	W	-	-
15	w2	Ld r2	-	-	-	F	D	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	W	-	-
16	w2	Mult r1,r1,r1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F	D	E	E	W	-
17	w2	Mult r2,r2,r2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F	D	E	E	W

Resultados dos Estudantes para Visualização com cores II



WSCAD 2023



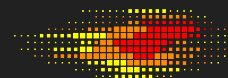
Código: parte_1

	Warp	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	w1	ld	-	ld	-	-	-	-	-	-	mul	-	-	-	mul	-	-	-	sd	-	-	-
1	w2	-	ld	-	ld	-	-	-	-	-	-	mul	-	-	-	mul	-	-	-	sd	-	-
2	w3	-	-	-	-	ld	-	ld	-	-	-	-	mul	-	-	-	mul	-	-	-	sd	-
3	w4	-	-	-	-	-	ld	-	ld	-	-	-	-	mul	-	-	-	mul	-	-	-	sd

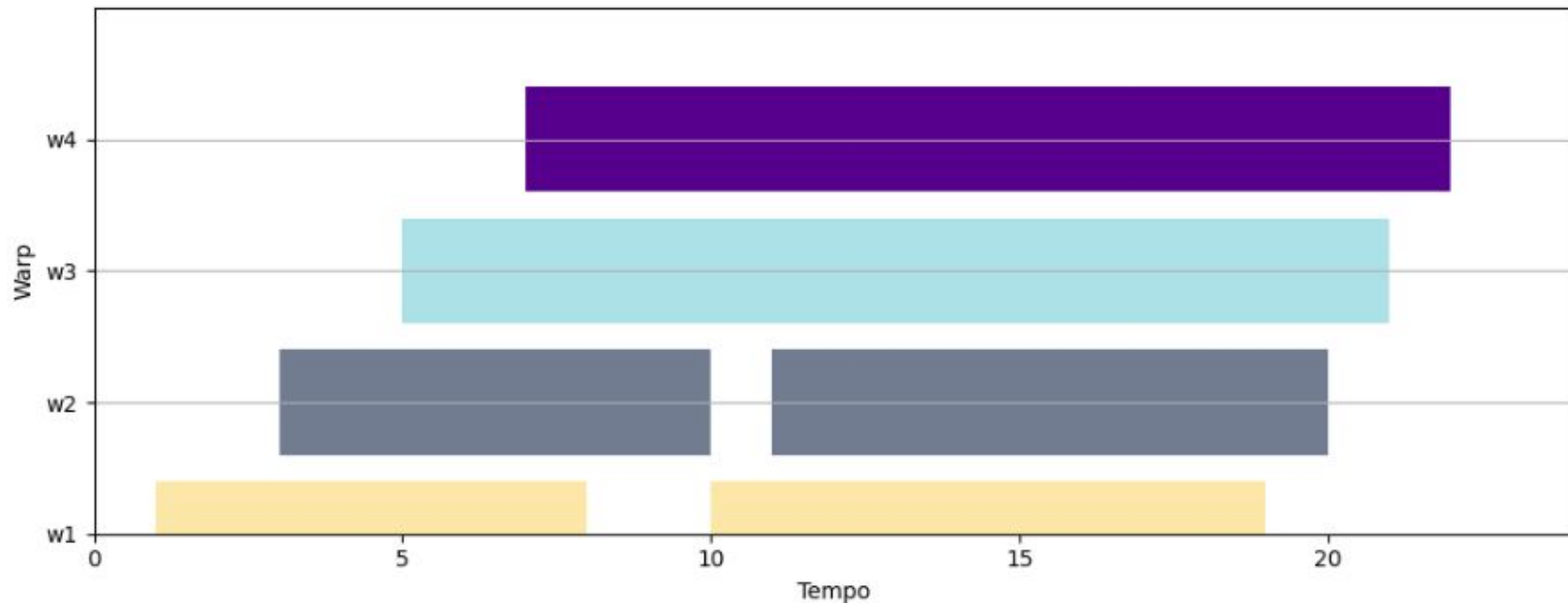
Legenda:

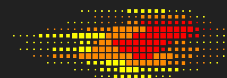
- w1: darkslategrey
- w2: indigo
- w3: green
- w4: red

Resultados dos Estudantes para Visualização com barras



WSCAD 2023





Considerações Finais

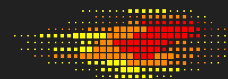
- Duas novas ferramentas de Ensino: Software Pipelining e Escalonamento de GPU
- Estudantes gostaram de usar Python e de aprender recursos de visualização
- Estudantes não tiveram dificuldades (comparado aos anos anteriores) para resolver questões dos temas trabalhados nas avaliações



Perguntas ?
ricardo@ufv.br

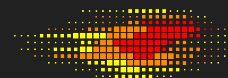
Acknowledgments

Financial support from FAPEMIG APQ-01577-22, CNPq, and UFV. This work was also carried out with the support of the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Financing Code 001



Obrigado pela atenção e perguntas ?
ricardo@ufv.br





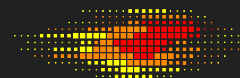
outros links de Colab
ricardo@ufv.br

Livro Lógica Digital



Minicurso Educomp 2023





WSCAD 2023

outros links de Colab
ricardo@ufv.br

lab. de Ideias



SBESC 2020 ESSE

