1 Descrizione

Il benchmark è costruito sul calcolo algebrico di moltiplicazione matrice per vettore. La computazione considerata è uno stream di matrici, il vettore secondo operando della moltiplicazione rimane costante per tutta l'esecuzione dell'applicazione. Per l'implementazione si è adottato il paradigma data parallel *Map*. I canali di comunicazione sono quelli del supporto, quindi vengono trasmessi i riferimenti alle strutture dati condivise in memoria, ne segue che la computazione segue il modello *multicast-compute-gather*, in quanto, sui moduli worker della *Map* viene attuata la distribuzione del riferimento di un elemento dello stream piuttosto che una distribuzione delle partizioni di un elemento dello stream. Il calcolo della moltiplicazione viene svolto per mezzo del prodotto scalare di ogni riga per il vettore secondo operando, il partizionamento delle matrici è quindi realizzato per righe, ciascun processo worker calcola, per una matrice, un certo numero di prodotti scalari che costituiscono una partizione del vettore risultato per quella matrice.

2 Misure sul benchmark

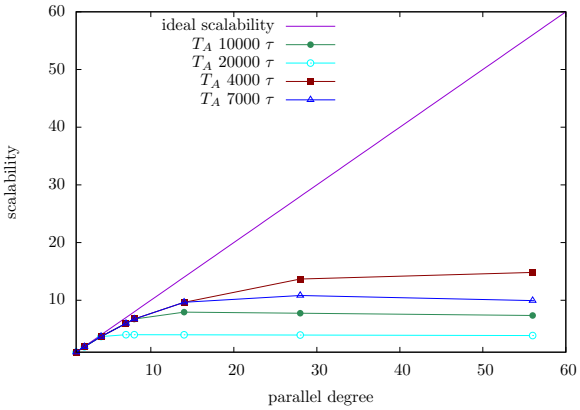
2.1 Map Complete Time

```
1  void * generator_task(void *args)
2  {
3    ...
4    pthread_barrier_wait(...);
5    atomic_compiler_barrier();
6    complTime_start = get_clock_cycle();
7    atomic_compiler_barrier();
8    for (i=0; i<m; i++) {
9      a = get_clock_cycle();
10     sym_send(ch_out, A_set[i]);
11     while (get_clock_cycle() - a < Ta)
12       ;
13   }
14   ...
15 }
16 void * worker_task(void *args)
17 {
18   if (x == y)
19   ...
20   pthread_barrier_wait(...);
21   for (i=0; i<m; i++) {
22     A = (int *)sym_receive(ch_in);
23     if (ch_left != NULL) sym_send(ch_left, A);
24     if (ch_right != NULL) sym_send(ch_right, A);
25     for (j=0; j<g; j++) {
26       C[j*rank] = 0;
27       for (k=0; k<M; k++)
28         C[j*rank] = *(A+j*M+k) * B[k] + C[j*rank];
29     }
30     asymin_send(ch_out, C);
31   }
32   ...
33 }
34 void * collector_task(void *args)
35 {
36   ...
37   pthread_barrier_wait(...);
38   for (i=0; i<m; i++) {
39     (void)asymin_receive(ch_in);
40   }
41   atomic_compiler_barrier();
42   complTime_stop = get_clock_cycle();
43   fprintf(outputfile, ``%d %f\n'', n, complTime_stop-complTime_start);
44   ...
45 }
```

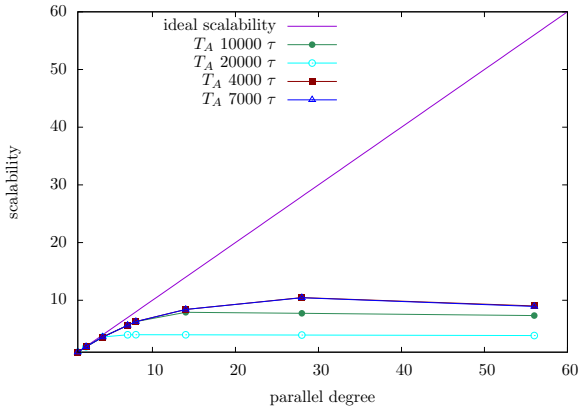
Figure 1: Grafici di scalabilità del tempo di completamento dello stream al variare del tempo di interarrivo

(a) Implementazione con solo UDN

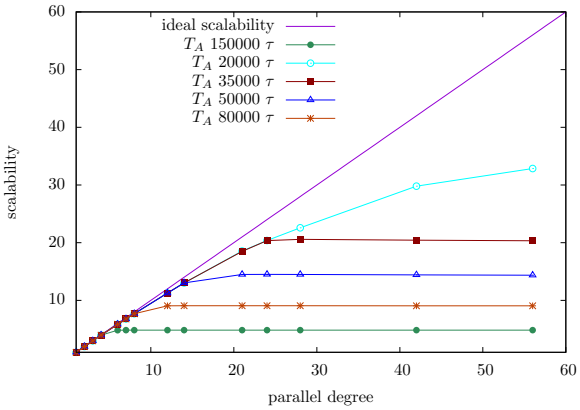
(b) Implementazione con solo SM



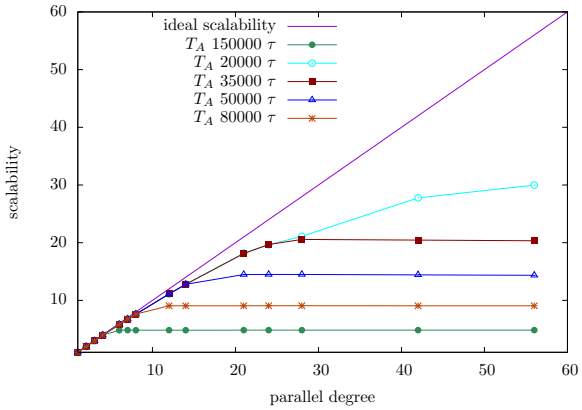
(a1) Scalabilita dell implementazione UDN con M=56 al variare del tempo di interarrivo



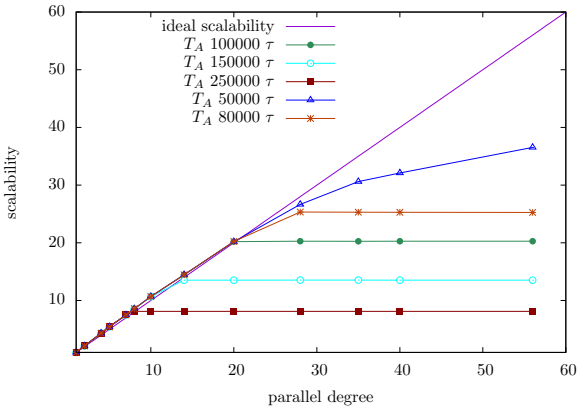
(b1) Scalabilita dell implementazione SM con M=56 al variare del tempo di interarrivo



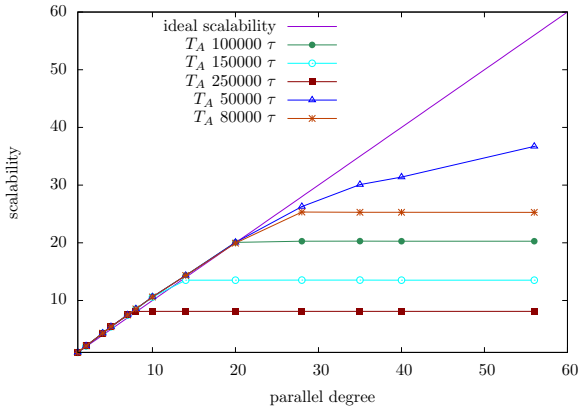
(a2) Scalabilita dell implementazione UDN con M=168 al variare del tempo di interarrivo



(b2) Scalabilita dell implementazione SM con M=168 al variare del tempo di interarrivo



(a3) Scalabilita dell implementazione UDN con M=280 al variare del tempo di interarrivo

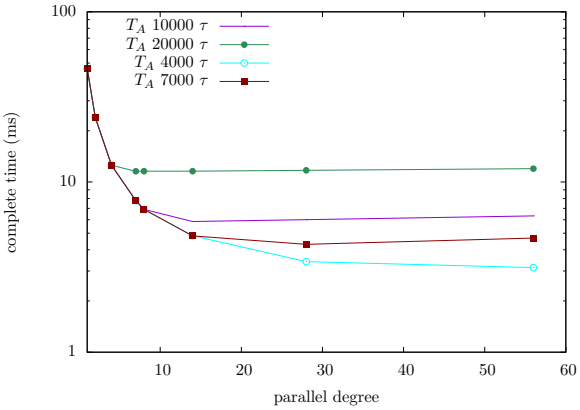


(b3) Scalabilita dell implementazione SM con M=280 al variare del tempo di interarrivo

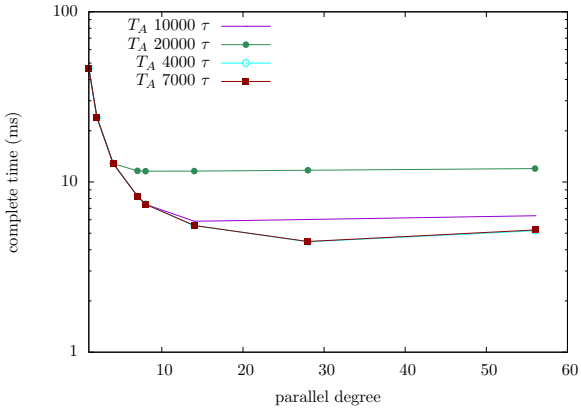
Figure 2: Grafici del tempo di completamento al variare del tempo di interarrivo

(a) Implementazione con solo UDN

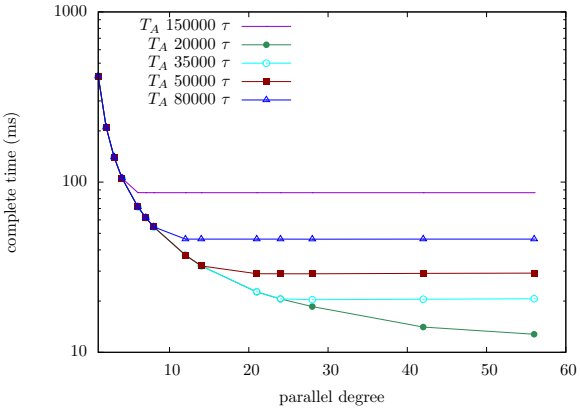
(b) Implementazione con solo SM



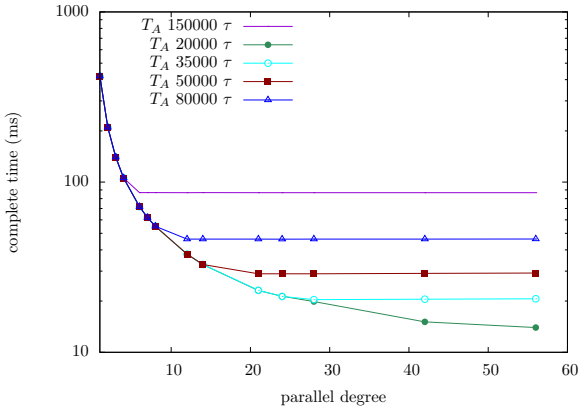
(a1) Tempo di completamento dell implementazione UDN con M=56 al variare del tempo di interarrivo



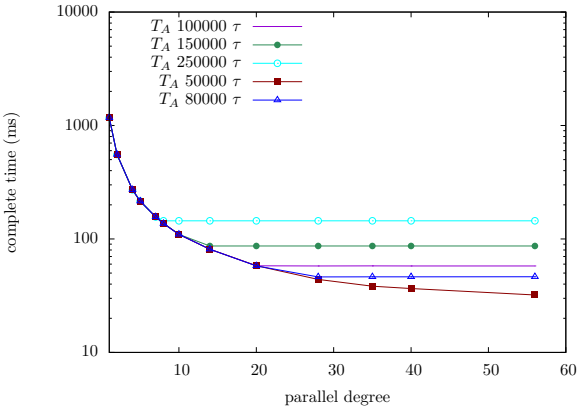
(b1) Tempo di completamento dell implementazione SM con M=56 al variare del tempo di interarrivo



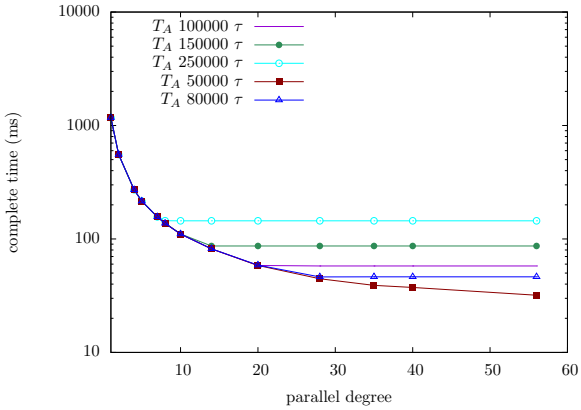
(a2) Tempo di completamento dell implementazione UDN con M=168 al variare del tempo di interarrivo



(b2) Tempo di completamento dell implementazione SM con M=168 al variare del tempo di interarrivo



(a3) Tempo di completamento dell implementazione UDN con M=280 al variare del tempo di interarrivo



(b3) Tempo di completamento dell implementazione SM con M=280 al variare del tempo di interarrivo

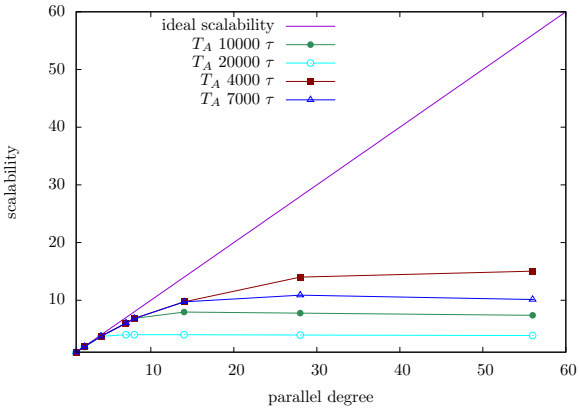
2.2 Map service time

```
void * generator_task(void *args...) {
  ...
  pthread_barrier_wait(...);
  for (i=0; i<m; i++) {
    a = get_clock_cycle();
    sym_send(ch_out, A_set[i]);
    while (get_clock_cycle() - a < Ta)
      ;
  }
  ...
}

void * worker_task(void *args...) {
  ...
  servicet_stop = 0;
  pthread_barrier_wait(...);
  for (i=0; i<m; i++) {
    A = (int *)sym_receive(ch_in);
    atomic_compiler_barrier();
    if (rank == 0) {
      servicet_start = servicet_stop;
      servicet_stop = get_clock_cycle();
      if (servicet_start != 0)
        servicet_sum += servicet_stop - servicet_start;
    }
    atomic_compiler_barrier();
    if (ch_left != NULL) sym_send(ch_left, A);
    if (ch_right != NULL) sym_send(ch_right, A);
    for (j=0; j<g; j++) {
      C[j*rank] = 0;
      for (k=0; k<M; k++)
        C[j*rank] = *(A+j*M+k) * B[k] + C[j*rank];
    }
    asymin_send(ch_out, C);
  }
  if (rank == 0)
    fprintf(output_file, ``%d %f\n'', n, (double)servicet_sum/m);
  ...
}

void * collector_task(void *args...) {
  ...
  pthread_barrier_wait(...);
  for (i=0; i<m; i++) {
    (void)asymin_receive(ch_in);
  }
  ...
}
```

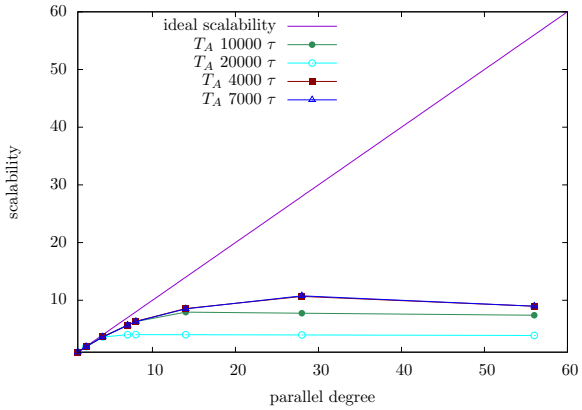
Figure 3: Grafici di scalabilità del tempo di servizio dello stream al variare del tempo di interarrivo

(a) Implementazione con solo UDN

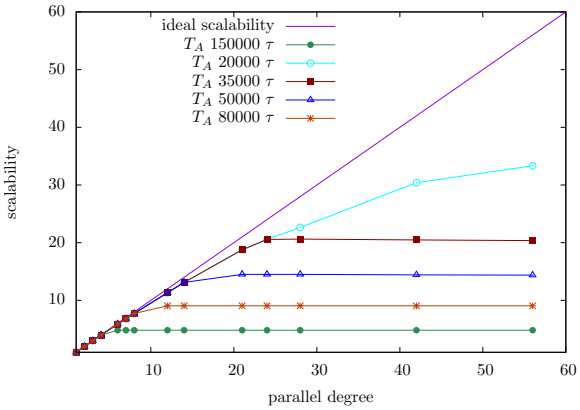
(b) Implementazione con solo SM



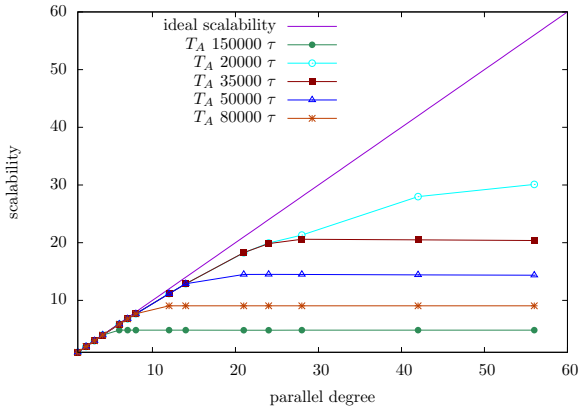
(a1) Scalabilita dell implementazione UDN con M=56 al variare del tempo di interarrivo



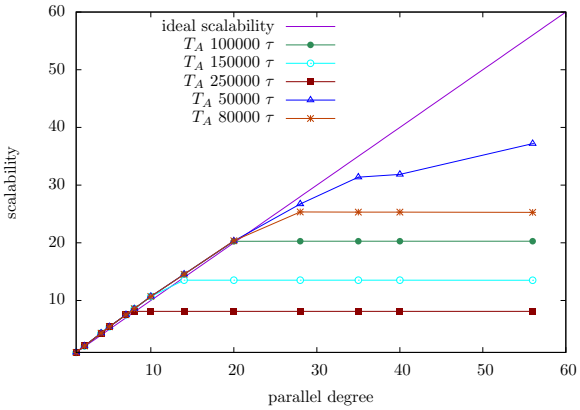
(b1) Scalabilita dell implementazione SM con M=56 al variare del tempo di interarrivo



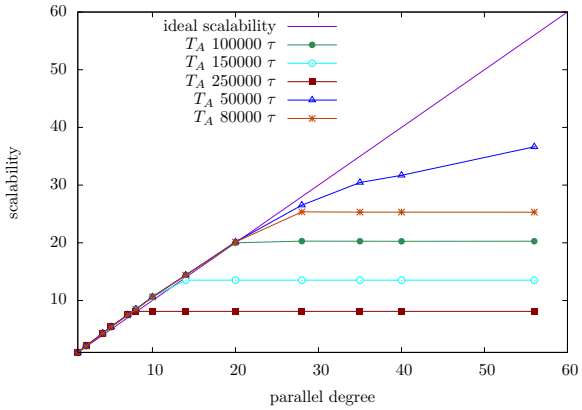
(a2) Scalabilita dell implementazione UDN con M=168 al variare del tempo di interarrivo



(b2) Scalabilita dell implementazione SM con M=168 al variare del tempo di interarrivo



(a3) Scalabilita dell implementazione UDN con M=280 al variare del tempo di interarrivo

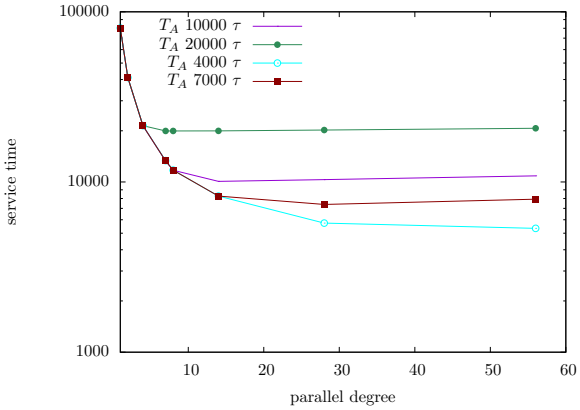


(b3) Scalabilita dell implementazione SM con M=280 al variare del tempo di interarrivo

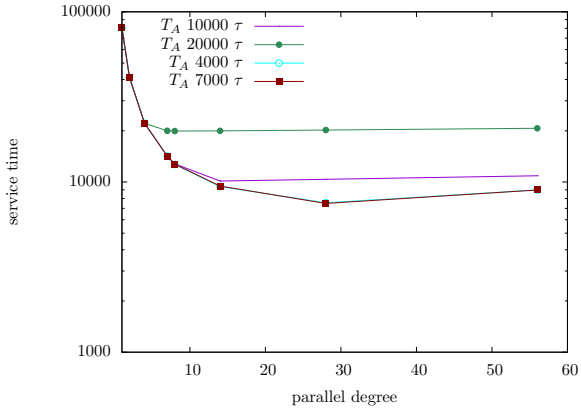
Figure 4: Grafici del tempo di servizio al variare del tempo di interarrivo

(a) Implementazione con solo UDN

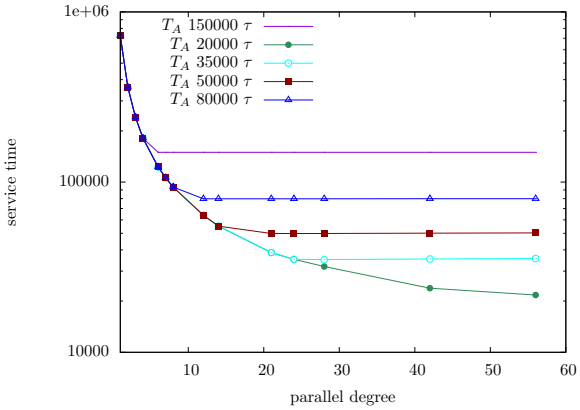
(b) Implementazione con solo SM



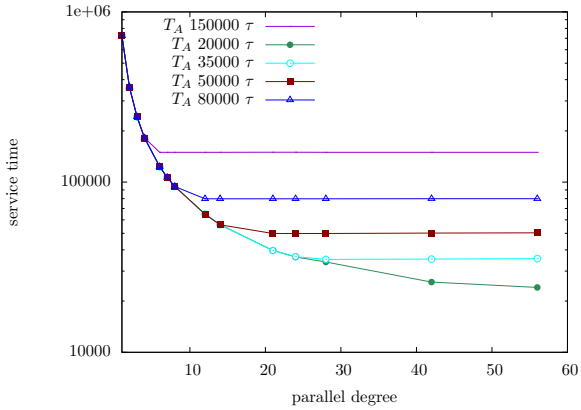
(a1) Tempo di servizio dell implementazione UDN con M=56 al variare del tempo di interarrivo



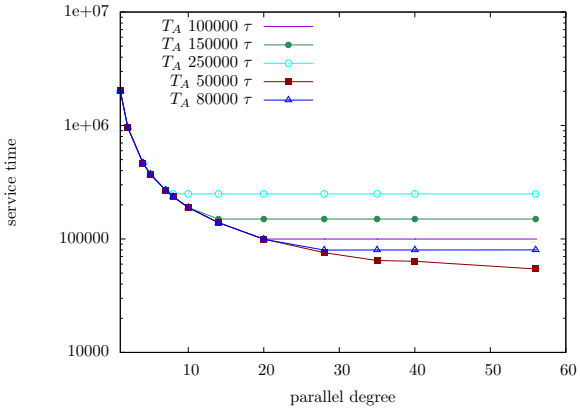
(b1) Tempo di servizio dell implementazione SM con M=56 al variare del tempo di interarrivo



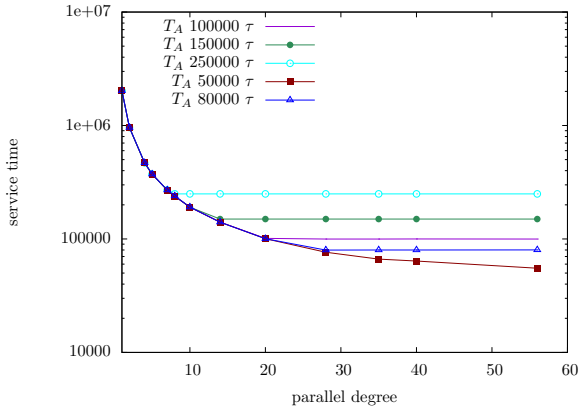
(a2) Tempo di servizio dell implementazione UDN con M=168 al variare del tempo di interarrivo



(b2) Tempo di servizio dell implementazione SM con M=168 al variare del tempo di interarrivo



(a3) Tempo di servizio dell implementazione UDN con M=280 al variare del tempo di interarrivo



(b3) Tempo di servizio dell implementazione SM con M=280 al variare del tempo di interarrivo

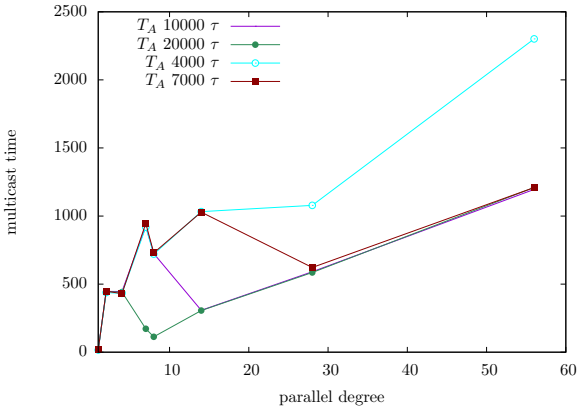
2.3 Multicast service time

```
void * generator_task(void *args...) {
  ...
  pthread_barrier_wait(...);
  for (i=0; i<m; i++) {
    a = get_clock_cycle();
    sym_send(ch_out, A_set[i]);
    while (get_clock_cycle() - a < Ta)
      ;
  }
  ...
}

void * worker_task(void *args...) {
  ...
  pthread_barrier_wait(...);
  for (i=0; i<m; i++) {
    A = (int *)sym_receive(ch_in);
    atomic_compiler_barrier();
    if (rank == 0)
      multicast_start = get_clock_cycle();
    atomic_compiler_barrier();
    if (ch_left != NULL) sym_send(ch_left, A);
    if (ch_right != NULL) sym_send(ch_right, A);
    atomic_compiler_barrier();
    if (rank == 0)
      multicast_sum = get_clock_cycle() - multicast_start;
    atomic_compiler_barrier();
    for (j=0; j<g; j++) {
      C[j*rank] = 0;
      for (k=0; k<M; k++)
        C[j*rank] = *(A+j*M+k) * B[k] + C[j*rank];
    }
    asymin_send(ch_out, C);
  }
  if (rank == 0)
    fprintf(output_file, ``%d %f\n'', n, (double)multicast_sum/m);
  ...
}

void * collector_task(void *args...) {
  ...
  pthread_barrier_wait(...);
  for (i=0; i<m; i++) {
    (void)asymin_receive(ch_in);
  }
  ...
}
```

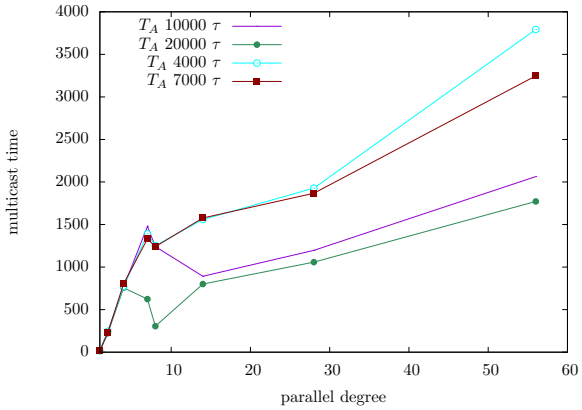
Figure 5: Grafici del tempo di multicast al variare del tempo di interarrivo

(a) Implementazione con solo UDN

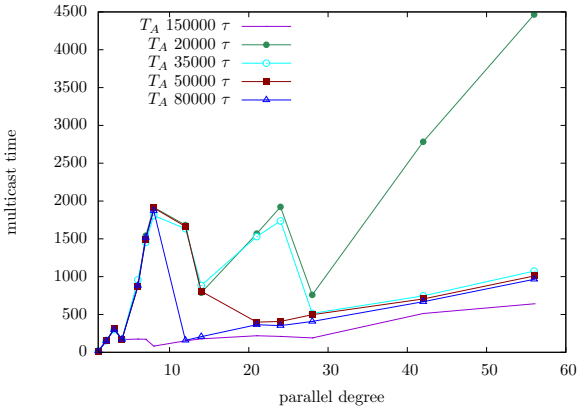
(b) Implementazione con solo SM



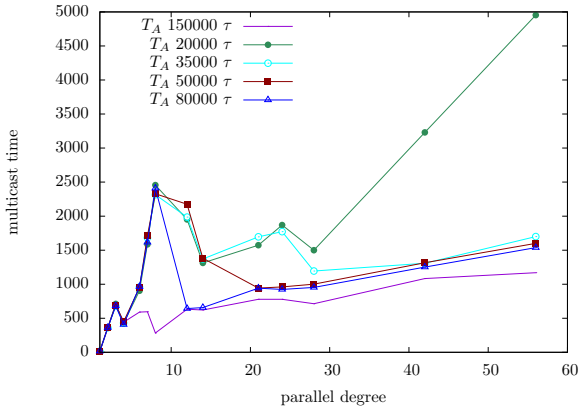
(a1) tempo di multicast dell implementazione UDN con M=56 al variare del tempo di interarrivo



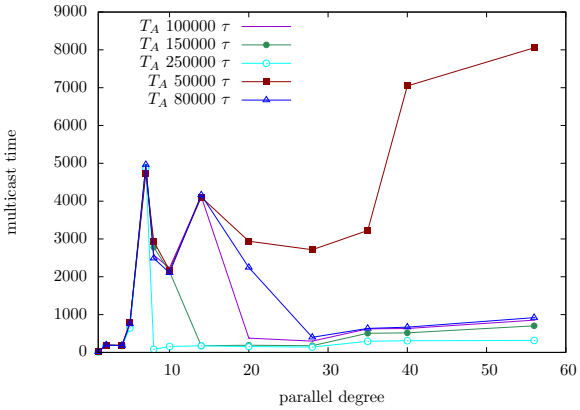
(b1) tempo di multicast dell implementazione SM con M=56 al variare del tempo di interarrivo



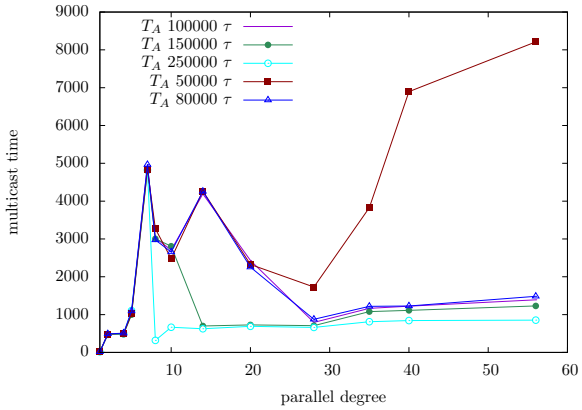
(a2) tempo di multicast dell implementazione UDN con M=168 al variare del tempo di interarrivo



(b2) tempo di multicast dell implementazione SM con M=168 al variare del tempo di interarrivo

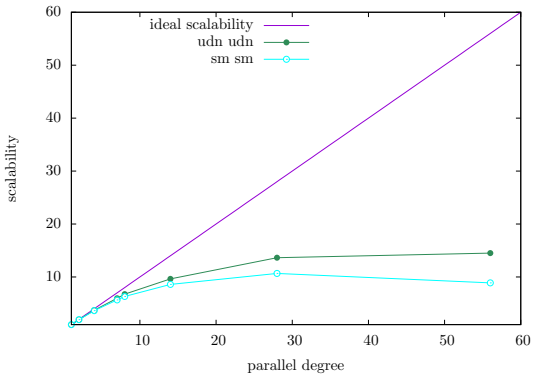


(a3) tempo di multicast dell implementazione UDN con M=280 al variare del tempo di interarrivo

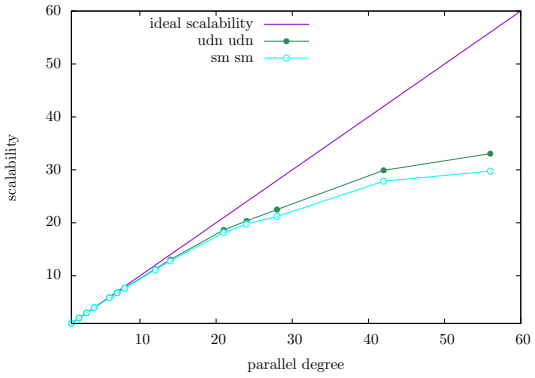


(b3) tempo di multicast dell implementazione SM con M=280 al variare del tempo di interarrivo

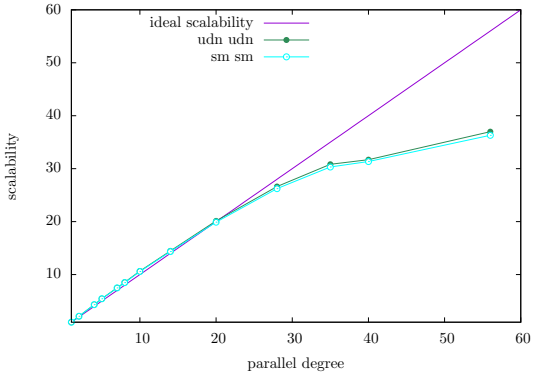
2.4 Confronto scalabilità delle due implementazioni



(a) Confronto della scalabilita nelle diverse implementazioni, Ta=181, M=56

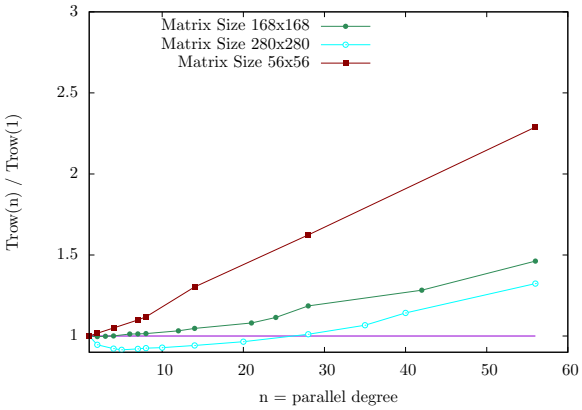


(b) Confronto della scalabilita nelle diverse implementazioni, Ta=181, M=168

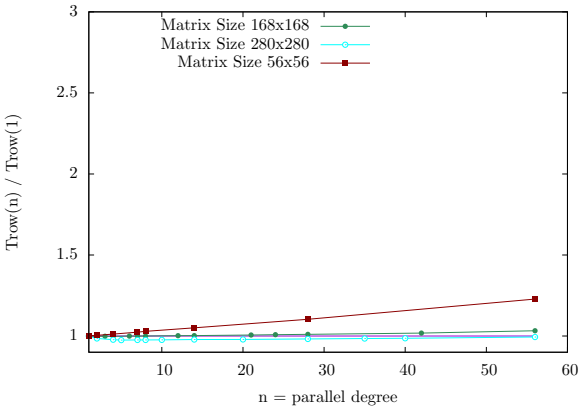


(c) Confronto della scalabilita nelle diverse implementazioni, Ta=181, M=280

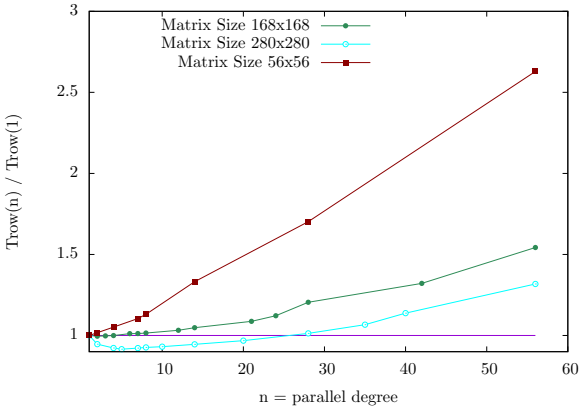
2.5 Row calculation time



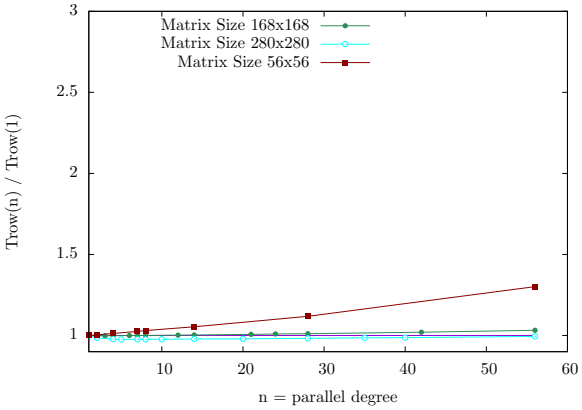
(d) Tempi di calcolo di una singola Row · Col con Ta=181, canali UDN e dati Int



(e) Tempi di calcolo di una singola Row · Col con Ta=181, canali UDN e dati Float



(f) Tempi di calcolo di una singola Row · Col con Ta=181, canali SM e dati Int



(g) Tempi di calcolo di una singola Row · Col con Ta=181, canali SM e dati Float

2.6 Data

```
1 46.572873 46.577579 46.568166 0.004707 1.000000 1.000000
2 23.940019 24.001436 23.878603 0.061417 1.945398 0.514034
4 12.540415 12.581304 12.499527 0.040888 3.713822 0.269264
7 7.763002 7.789110 7.736894 0.026108 5.999338 0.166685
8 6.896573 6.918742 6.874404 0.022169 6.753046 0.148081
14 4.831994 4.855326 4.808662 0.023332 9.638438 0.103751
28 3.404672 3.434254 3.375089 0.029582 13.679109 0.073104
56 3.140727 3.150936 3.130518 0.010209 14.828692 0.067437
```

Listing 1: complete time: UDN, M=56, T_{A}=4000

```
1 80306.783600 80339.122000 80276.256000 21.842300 1.000000 1.000000
2 41090.840000 41258.916000 41032.488000 85.303538 1.954372 0.511673
4 21424.034800 21517.142000 21370.560000 62.165576 3.748443 0.266777
7 13351.709200 13505.258000 13222.912000 91.890952 6.014719 0.166259
8 11738.808800 11794.696000 11702.608000 39.525586 6.841136 0.146175
14 8262.596400 8422.300000 8190.248000 85.364889 9.719316 0.102888
28 5731.913600 5808.128000 5660.768000 49.787773 14.010467 0.071375
56 5340.577600 5475.144000 5257.590000 73.023736 15.037097 0.066502
```

Listing 2: service time: UDN, M=56, T_{A}=4000

```
1 1172.604179 1172.730687 1172.497302 0.088870 1.000000 1.000000
2 554.896234 555.023791 554.638248 0.151277 2.113195 0.473217
4 270.980234 271.176319 270.806410 0.132843 4.327268 0.231093
5 215.143113 215.386131 214.985249 0.163714 5.450345 0.183475
7 156.426926 156.434588 156.416539 0.007640 7.496179 0.133401
8 137.168518 138.592423 136.675336 0.822187 8.548639 0.116978
10 109.756940 110.067086 109.594116 0.192402 10.683645 0.093601
14 81.106906 81.112192 81.102273 0.003559 14.457513 0.069168
20 58.089348 58.981450 57.776880 0.515355 20.186217 0.049539
28 44.008273 44.380543 43.599370 0.346627 26.645085 0.037530
35 38.308934 39.305718 37.767630 0.617401 30.609157 0.032670
40 36.530533 37.216905 36.148452 0.429463 32.099290 0.031153
56 32.095102 32.656763 31.680585 0.363380 36.535300 0.027371
```

Listing 3: complete time: UDN, M=280, T_{A}=50000

```
1 2023768.777600 2024256.074000 2023380.242000 311.804099 1.000000 1.000000
2 957795.432000 958710.060000 957027.954000 605.016532 2.112945 0.473273
4 467164.594400 467897.934000 466548.600000 604.307672 4.332025 0.230839
5 371265.169600 372382.774000 369883.160000 1002.447347 5.451006 0.183452
7 268932.414000 269631.154000 268520.218000 408.045018 7.525195 0.132887
8 236590.966400 238950.566000 234135.138000 2051.827267 8.553872 0.116906
10 189140.739200 189813.032000 188529.550000 487.286184 10.699804 0.093460
14 139142.949600 139317.330000 139017.336000 103.618498 14.544530 0.068754
20 99578.224800 100195.734000 99081.104000 399.249887 20.323407 0.049204
28 75727.979200 76134.740000 75268.626000 289.770418 26.724188 0.037419
35 64494.868400 65239.048000 63932.220000 450.363999 31.378757 0.031869
40 63544.873600 63910.226000 62880.038000 375.831233 31.847868 0.031399
56 54419.289600 55483.402000 53446.486000 670.050446 37.188445 0.026890
```

Listing 4: service time: UDN, M=280, T_{A}=50000

```
1 46.633566 46.640408 46.626723 0.006842 1.000000 1.000000
2 23.972188 24.053576 23.890800 0.081388 1.945320 0.514054
4 12.857453 12.897841 12.817065 0.040388 3.626968 0.275712
7 8.240676 8.255497 8.225856 0.014821 5.658949 0.176711
8 7.403380 7.486033 7.320727 0.082653 6.298956 0.158756
14 5.549339 5.691791 5.406887 0.142452 8.403445 0.118999
28 4.454536 4.460204 4.448867 0.005668 10.468783 0.095522
56 5.174670 5.248622 5.100719 0.073952 9.011891 0.110965
```

Listing 5: complete time: SM, M=56, T_{A}=4000

```
1 80508.637600 80535.508000 80481.106000 18.470758 1.000000 1.000000
2 41231.674000 41444.214000 41157.326000 108.304456 1.952592 0.512140
4 22132.855600 22224.392000 22006.220000 85.550438 3.637517 0.274913
7 14191.646000 14227.322000 14166.200000 22.955103 5.672960 0.176275
8 12672.152000 12901.366000 12560.782000 129.896767 6.353194 0.157401
14 9396.887600 9647.740000 9083.762000 199.155636 8.567585 0.116719
28 7552.140000 7601.304000 7465.862000 45.492339 10.660374 0.093805
56 8971.953600 9202.742000 8755.614000 146.864666 8.973368 0.111441
```

Listing 6: service time: SM, M=56, T_{A}=4000

```
1 1172.625483 1172.661315 1172.606780 0.021175 1.000000 1.000000
2 555.325347 555.565762 555.196072 0.143666 2.111601 0.473574
4 271.317352 271.771757 271.003461 0.317606 4.321970 0.231376
5 215.455648 215.861921 215.100179 0.333805 5.442538 0.183738
7 156.537881 156.724819 156.250732 0.183856 7.491001 0.133494
8 137.309426 139.010772 136.565928 0.993934 8.540022 0.117096
10 110.437681 110.660172 110.191696 0.200325 10.617984 0.094180
14 81.787152 81.922388 81.658911 0.094188 14.337527 0.069747
20 58.407352 58.561989 58.300898 0.097276 20.076676 0.049809
28 44.623162 45.251591 44.280006 0.391794 26.278404 0.038054
35 38.978256 39.445141 38.416002 0.370590 30.084093 0.033240
40 37.362734 37.550192 36.985557 0.225755 31.384895 0.031862
56 31.940741 32.340694 31.383280 0.358839 36.712532 0.027239
```

Listing 7: complete time: SM, M=280, T_{A}=50000

```
1 2024263.975200 2024884.306000 2023758.474000 455.386393 1.000000 1.000000
2 959490.920800 961129.294000 958163.018000 1181.196625 2.109727 0.473995
4 468727.324000 470108.612000 467919.892000 800.515927 4.318639 0.231554
5 372576.783600 373591.678000 371093.200000 911.939768 5.433146 0.184055
7 269896.960000 270497.758000 268984.630000 597.479158 7.500136 0.133331
8 238152.980000 240016.576000 235129.462000 2182.643253 8.499847 0.117649
10 190952.532800 192352.862000 189913.284000 809.444213 10.600875 0.094332
14 140639.802800 140828.402000 140170.974000 237.775445 14.393251 0.069477
20 100481.775200 100769.664000 100300.788000 166.141682 20.145583 0.049639
28 76329.412400 76718.618000 75959.086000 259.386559 26.520104 0.037707
35 66487.960400 66989.034000 66043.926000 340.735403 30.445572 0.032845
40 63860.870000 64372.984000 63317.352000 400.053244 31.698033 0.031548
56 55241.196000 56769.636000 52561.614000 1491.433074 36.644101 0.027290
```

Listing 8: service time: SM, M=280, T_{A}=50000

```
1 16.242400 16.352000 16.172000 0.086259 1.000000 1.000000
2 437.554000 452.224000 428.072000 9.224755 0.037121 26.938999
4 432.136400 445.704000 387.080000 22.568567 0.037586 26.605452
7 918.366400 1031.870000 845.408000 64.491110 0.017686 56.541299
8 720.580800 765.184000 690.224000 25.415599 0.022541 44.364183
14 1032.088400 1064.280000 958.502000 37.658195 0.015737 63.542851
28 1078.153600 1111.918000 1051.084000 23.183473 0.015065 66.378959
56 2300.732400 2383.672000 2222.814000 62.187720 0.007060 141.649781
```

Listing 9: multicast service time: UDN, M=56, T_{A}=50000

```
1 16.340800 16.388000 16.172000 0.084443 1.000000 1.000000
2 244.598400 260.948000 218.956000 15.175001 0.066807 14.968569
4 769.862800 832.634000 639.930000 67.942743 0.021226 47.112920
7 1391.232400 1500.162000 1259.182000 77.017069 0.011746 85.138573
8 1252.989200 1351.056000 1139.006000 72.317936 0.013041 76.678571
14 1559.718800 1733.266000 1327.094000 131.612729 0.010477 95.449354
28 1927.965200 2037.566000 1876.946000 58.558603 0.008476 117.984750
56 3792.382800 5128.220000 3379.034000 672.987729 0.004309 232.080608
```

Listing 10: multicast service time: SM, M=56, T_{A}=50000

```
1 16.257200 16.386000 16.172000 0.104350 1.000000 1.000000
2 150.333600 160.586000 143.706000 6.679758 0.108141 9.247201
3 308.527600 349.930000 288.174000 21.362479 0.052693 18.977905
4 172.990400 175.690000 169.064000 2.547146 0.093977 10.640848
6 869.913600 959.672000 757.010000 81.267899 0.018688 53.509436
7 1542.417200 1761.002000 1393.990000 136.373739 0.010540 94.875944
8 1917.032400 1998.320000 1865.732000 47.555478 0.008480 117.918977
12 1682.530400 1795.816000 1493.630000 104.371807 0.009662 103.494476
14 787.048800 865.706000 683.760000 61.314898 0.020656 48.412322
21 1568.388400 1824.058000 1306.722000 194.189619 0.010366 96.473464
24 1922.515600 2112.286000 1760.940000 118.716412 0.008456 118.256256
28 759.130400 1038.136000 623.620000 149.250060 0.021416 46.695027
42 2783.569600 3053.820000 2553.088000 180.120929 0.005840 171.220727
56 4465.223200 4658.222000 4040.096000 226.231230 0.003641 274.661270
```

Listing 11: multicast service time: UDN, M=168, T_{A}=50000

```
1 16.192000 16.380000 16.064000 0.102886 1.000000 1.000000
2 362.937200 377.768000 352.526000 10.149624 0.044614 22.414600
3 711.163200 753.160000 680.418000 30.169340 0.022768 43.920652
4 426.780800 499.354000 405.258000 36.588317 0.037940 26.357510
6 904.210400 1145.686000 742.978000 181.769801 0.017907 55.843034
7 1586.216800 1748.692000 1267.128000 171.110207 0.010208 97.962994
8 2455.836400 2600.016000 2250.078000 116.094435 0.006593 151.669738
12 1951.270000 2140.142000 1750.972000 125.590734 0.008298 120.508276
14 1313.540800 1540.918000 1127.132000 155.223608 0.012327 81.122826
21 1573.766000 1775.300000 1193.588000 202.310289 0.010289 97.194046
24 1868.891600 2377.578000 1508.376000 283.998626 0.008664 115.420677
28 1499.463600 1588.302000 1388.352000 73.536301 0.010799 92.605212
42 3230.684000 3428.478000 3012.490000 146.075578 0.005012 199.523468
56 4953.270000 5109.020000 4760.144000 132.691822 0.003269 305.908473
```

Listing 12: multicast service time: SM, M=168, T_{A}=50000

```
1 16.164800 16.172000 16.136000 0.014400 1.000000 1.000000
2 195.049600 201.594000 189.812000 4.081222 0.082875 12.066317
4 184.612800 187.246000 181.306000 2.208185 0.087561 11.420667
5 798.118000 1285.348000 564.144000 273.576851 0.020254 49.373825
7 4724.094400 5537.634000 4264.986000 491.549694 0.003422 292.245769
8 2934.568400 3858.104000 2015.358000 733.945570 0.005508 181.540656
10 2172.013600 2448.634000 1794.464000 223.417365 0.007442 134.366871
14 4104.190800 4872.960000 3409.462000 545.325687 0.003939 253.896788
20 2939.463600 4453.300000 1747.392000 1011.519187 0.005499 181.843487
28 2714.732000 3653.982000 2194.900000 509.990550 0.005954 167.940958
35 3223.376000 3497.330000 2943.992000 242.854323 0.005015 199.407107
40 7045.316800 8074.182000 5892.610000 755.210515 0.002294 435.843116
56 8053.298000 9055.652000 7208.224000 661.302127 0.002007 498.199668
```

Listing 13: multicast service time: UDN, M=280, T_{A}=50000

```
1 16.164800 16.172000 16.136000 0.014400 1.000000 1.000000
2 477.264000 525.666000 453.120000 25.400649 0.033870 29.524894
4 501.680400 588.842000 453.270000 49.187659 0.032221 31.035361
5 1016.081600 1180.494000 913.536000 91.441456 0.015909 62.857666
7 4843.610800 5607.058000 4090.604000 616.532145 0.003337 299.639389
8 3269.133600 3959.012000 2282.134000 707.812790 0.004945 202.237801
10 2479.208000 3111.158000 1800.060000 465.771195 0.006520 153.370781
14 4267.509200 4942.838000 3610.872000 465.472879 0.003788 264.000124
20 2324.714000 4008.764000 1606.246000 856.422083 0.006953 143.813348
28 1729.702400 2839.424000 1122.482000 689.217153 0.009345 107.004256
35 3827.683600 4611.220000 2487.026000 810.790973 0.004223 236.791275
40 6894.591600 7180.134000 6521.028000 263.803277 0.002345 426.518831
56 8211.370400 9169.862000 7502.674000 661.629644 0.001969 507.978472
```

Listing 14: multicast service time: SM, M=280, T_{A}=50000