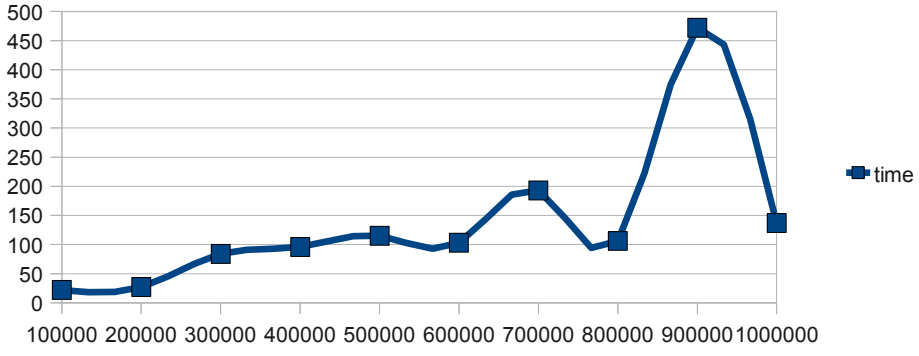
Wolfgang Huettinger
0685854
whuettin@uoguelph.ca

This a further analysis of the program runs of the application given in the assignment 1 in the programming course CIS*6100.

After I programmed the application I let it run with several modifications. Due to the fact that I was not very successful installing pilot on my machine which probably lies in the lack of Intel compiler I decided to limit myself to the machine mako.sharcnet.ca.
This resulted in a 60 minute limit within the program can run.

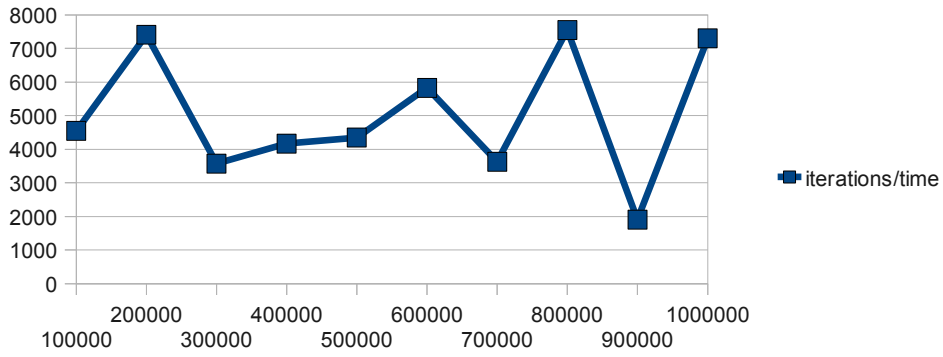
Never the less I found interesting results. With a grid size of only 100x100 I increased the number of iterations and got the following run time:

iterations	time
100000	22
200000	27
300000	84
400000	96
500000	115
600000	103
700000	193
800000	106
900000	472
1000000	137



As always there is a statistical error due to the case that I use a imperfect system. But by making a small modification of the numbers: iterations over time I get more information out of the data:

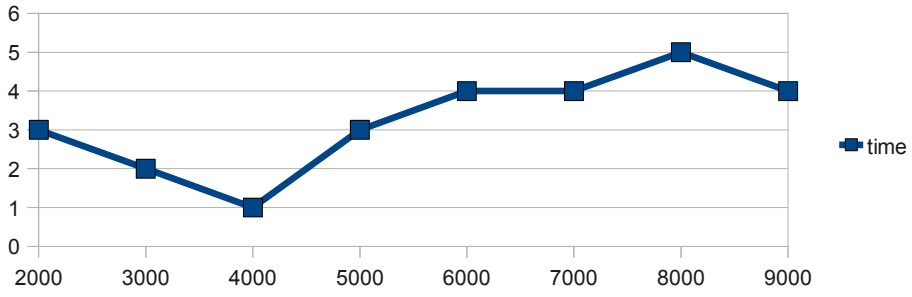
iterations	iterations/time
100000	4545.45
200000	7407.41
300000	3571.43
400000	4166.67
500000	4347.83
600000	5825.24
700000	3626.94
800000	7547.17
900000	1906.78
1000000	7299.27



With this modification I think I can say that the amount of time spent in calculations is very close linearly dependent on the number of iterations.

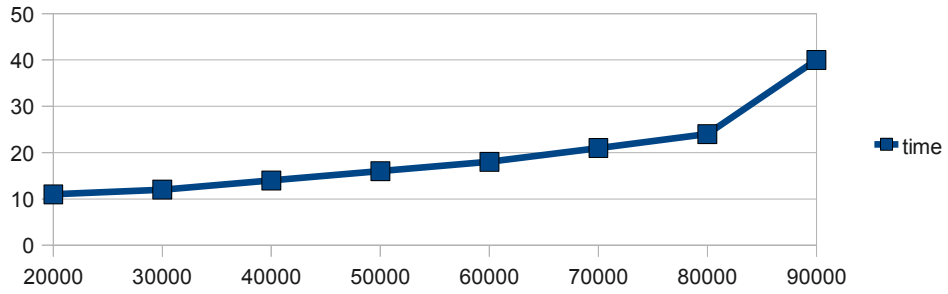
Further on I tried to modify the grid size and played with the Y dimension. At first I was only made small increases which resulted in nearly no changes in the amount of time a run took.

gridY	time
2000	3
3000	2
4000	1
5000	3
6000	4
7000	4
8000	5
9000	4



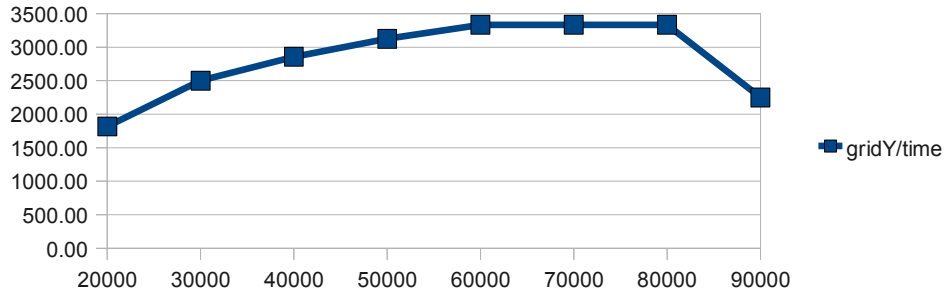
So I went on and increased the grid size and found interesting results:

gridY	time
20000	11
30000	12
40000	14
50000	16
60000	18
70000	21
80000	24
90000	40



By modifying the numbers again as I did above to see more:

gridY	gridY/time
20000	1818.18
30000	2500.00
40000	2857.14
50000	3125.00
60000	3333.33
70000	3333.33
80000	3333.33
90000	2250.00



As mentioned above there is probably a statistical error. But someone can say that the amount of run time depends close to linear by increasing the grid size.

After all I can say that the program is not perfect. One problem of adaption of the grid: Due to a problem with the pilot implementation the grid x coordinate takes more than one variable change. If the user wants to change the x coordinate he has to change 4 other values in PI_Write and PI_Read calls.

I had problems with the Pilot installation. I could not let the program run with the deadlock detection. The program did not stop until I removed it.