

Speed enhancing mechanisms in PIAS: PIAS/ES

A characteristic of our profession is that we often encounter intensive calculation tasks. Although the computer serves us well in this area for decades already, the processing time may still be a bottleneck, also because man has adapted himself to the increased processing power, and demands more extended calculations than without the computer would have been the case. This mechanism also manifests itself with PIAS, so it is worthwhile to strive for an optimized calculation process. For that purpose PIAS is equipped with three mechanisms which increase the speed, namely minimization of disc usage, optimization of control of hardware lock and multithreading. These three options, which are further discussed below, are offered combined in a package with the name *PIAS/ES*, where ES stands for *Enhanced Speed*.

Minimization of disc usage

Like any substantial computer program PIAS uses disc files for the storage of permanent data. Additionally, internal intermediate results are stored in temporary files. Unfortunately, we have experienced that the disc performance gets slower as the operating system version gets newer. Under Windows 95 disc IO is slower than under MS-DOS, while XP on its turn is slower than '95. Microsoft even succeeded to nullify the strongly increased speed of hardware and networks, quite an achievement indeed. This effect did gradually lead to considerable slower PIAS performance. We did do some experiments with the setup of the operating system or the network, however, without significant improvement. Furthermore, we are not acquainted with documentation which describes this problem or suggests a remedy. Because apparently the problem is intractable we decided to work around it.

For that purpose an alternative mechanism was designed, where the disc usage is minimized and the intermediate results are kept as much as possible in RAM memory. In technical terms a RAM-cache is placed between the program and the disc.

Optimization of control of hardware lock

Already since the early days of PIAS, it is protected with a hardware lock. Originally, the time required for a *read* operation was negligible, but also in this field the technological evolution does slow things down. Due to the complexity of the system software, these days reading a lock can take a multitude of the original time, and gradually this has got a decelerating effect on PIAS. Also this effect cannot be solved fundamentally, but it can be got around by minimizing the number of times a lock is read. This mechanism is applied with this option, in such a way that a new equilibrium between speed and protection is found.

Multithreading

This option makes a positive use of the latest computer technology. For a long time past a PC generally has one processor, while this processor contains one core. That implies that the computer can process one task at a time (although the operating system may fool you, and give the impression that multiple tasks are processed simultaneously). However, there is a tendency where a computer is equipped with multiple real or virtual processors. The latter is the case with the latest generation of Intel Pentium processors, which are equipped with the HT technology, where HT stands for *Hyper Threading*. The background of this technology can be read on e.g. <http://arstechnica.com/articles/paedia/cpu/hyperthreading.ars>. Summarized, it implies that a process, a computer program, can split itself into multiple tasks, which can be executed simultaneously. Such a task is called a *thread*, and the difference between *multithreading* and *hyper threading* is that the first processes the threads completely parallel, while the latter does that only partially.

So, the HT technology enables a program to execute tasks parallel, but the software will have to be adapted for that facility, where tasks which are suitable for simultaneous processing are explicitly offered to the processor for parallel processing. That implies that for every function of a software package it must be considered whether or not it is suitable for parallel processing, and it must be adapted accordingly, if appropriate. Limiting ourselves to PIAS, many tasks can be recognized which can be processed parallel, such as calculating damage stability for multiple angles of inclination, or drawing hull lines with Fairway. On the other hand, there are also jobs which are not suitable, such as the calculation of intermediate stages of flooding, where *at first* the final stage must be determined, *whereupon* the water level which corresponds to the filling percentage can be calculated.

Where possible PIAS will use multithreading at time-critical tasks, while for the time being the number of simultaneous tasks will be limited to two. However, in principle PIAS can also be adapted to possible future hardware developments which allow more than two simultaneous tasks.

The speed gain with the present generation of Intel HT processors, which can process two *threads* simultaneously, is at maximum 50%. Because not every task is suitable for parallel processing the effective acceleration with PIAS is less, provisional experiments indicate a percentage of 30 to 40.

Speed gain, price and availability

As indicated, for the multithreading option the acceleration gain can be motivated upon. Concerning the possible gain of processing speed of the other two options, no general statement can be made. It depends on the combination of computer, Windows version, network hardware, network software and hardware lock. For a specific configuration one will have to do its own experiments, SARC can not provide advice in this field.

The price of *PIAS /ES* is 4.000 Euros per license. *PIAS/ES* is only available for the family of Windows XP or 2000 platforms.

Minimization of disc usage is available since October 2004, optimization of control of hardware lock since present. The implementation of multithreading will take some time still, parts are currently being tested, and it can be expected that in the coming months it will be incorporated in a number of PIAS modules. Multithreading will first be implemented in time-critical parts, such as the several damage stability calculations, and some drawing functions in Fairway. Gradually it will be used with other tasks as well.