Performance and Productivity Benefits Using Multi-Core Processors for the Analysis of Digital Long-Term ECG Recordings

T Hilbel[1,2], RL Lux[3], J Dietzsch[4], M Schliephake[5], HA Katus[2]

[1]University of Applied Sciences Gelsenkirchen, Germany
[2]University of Heidelberg, Germany
[3]CVRTI, University of Utah, Salt Lake City, UT, USA
[4]University of Tübingen, Germany
[5]High Performance Computing Center Stuttgart, Germany

Abstract

Modern Holter recorders allow the acquisition of 12 lead ECGs with a sampling rate of 1KHz or higher and a resolution of 16 bits over more than 24h. While large volumes of data can be easily stored on flash memory cards the analysis of these biosignals requires considerable calculation power and network bandwidth. In general, processing time is important when hundreds of digital recordings from large study groups need to be analyzed. The objective of the following investigation is to address the question: Can performance and productivity benefits in ECG analysis be achieved using multi-core processor technology? Because these processors have two or more processing cores they can perform parallel processing. The results show that segmentation of Holter data and running the same program multiple times simultaneously can dramatically speedup the computing performance.

1. Introduction

Modern Holter ECG recorders allow the acquisition of 12 lead ECGs with a sampling rate of 1000 Hz or higher and a digital resolution of 16 bits over more than 24h. While large volumes of data can be easily stored on flash memory cards the analysis of these digital biosignals require considerable calculation power and network bandwidth. In general processing time is important especially when hundreds of digital recordings from large study groups need to be analyzed. The objective of the following investigation is to address the question: Can performance and productivity benefits in ECG analysis be achieved using multi-core processor technology? Historically, CPU speeds doubled every two or three years. Due to technical limitations current CPU speeds of 3-4 GHz are unlikely to increase substantially and will limit development. While the increase of the clock speed is hindered by the difficulties to cool such a chip, the currently ongoing process of miniaturization allows the integration of more electronic elements to construct multiple processing units (CPUs) on a single chip. Thus future process acceleration will only be achieved using parallel processing such as currently supported by multi-core chips (Fig. 1). All the latest desktop or notebook computers contain multi-core processors. Multi-core processors are single chips that contain two or more distinct CPUs. The two or more processing cores can perform parallel processing under standard multitasking capable operating systems without special programming requirements.

2. Methods

The relative performance changes in execution time of an ECG analysis program that calculates RR-Intervals, HRV parameters, QT-Duration and Eigenvectors from 1000Hz, 16 bit 24 hour Holter data was compared when analyzing 24h of ECG as a single program on dual-core machines or as several parallel program instances running at the same time under Windows XP®. In the single program test the program analyzed hours 1 – 24 of a high resolution 12 lead Holter recording. In the two programs test the same program was started twice. One program was set to analyze hour 1 – 12 of the data while the second program was set to analyze hour 13 – 24 of the biosignals at the same time. So each of the two programs could use one of the processor cores at the same time. The performance was also measured on a Windows 2003® server with two separated Intel XEON® dual-core processors. In the server setting the same program was started up to four times under the assumption that the server provides up to 4 processor cores. Performance was determined in a one program and a two parallel programs test as described above, also in a three parallel programs test while the first program instance was analyzing hour 1 – 8 the second instance did analyze hour 9 – 16 and the third program instance did analyze hour 17 – 24. Final performance was determined in a four parallel programs test while the first program instance was analyzing hour 1 – 6 the second instance did analyze hour 7 – 12, the third program instance did analyze hour 13 – 18 and the fourth program instance did analyze hour 19 – 24 at the same time in parallel (Fig. 2,3,4).

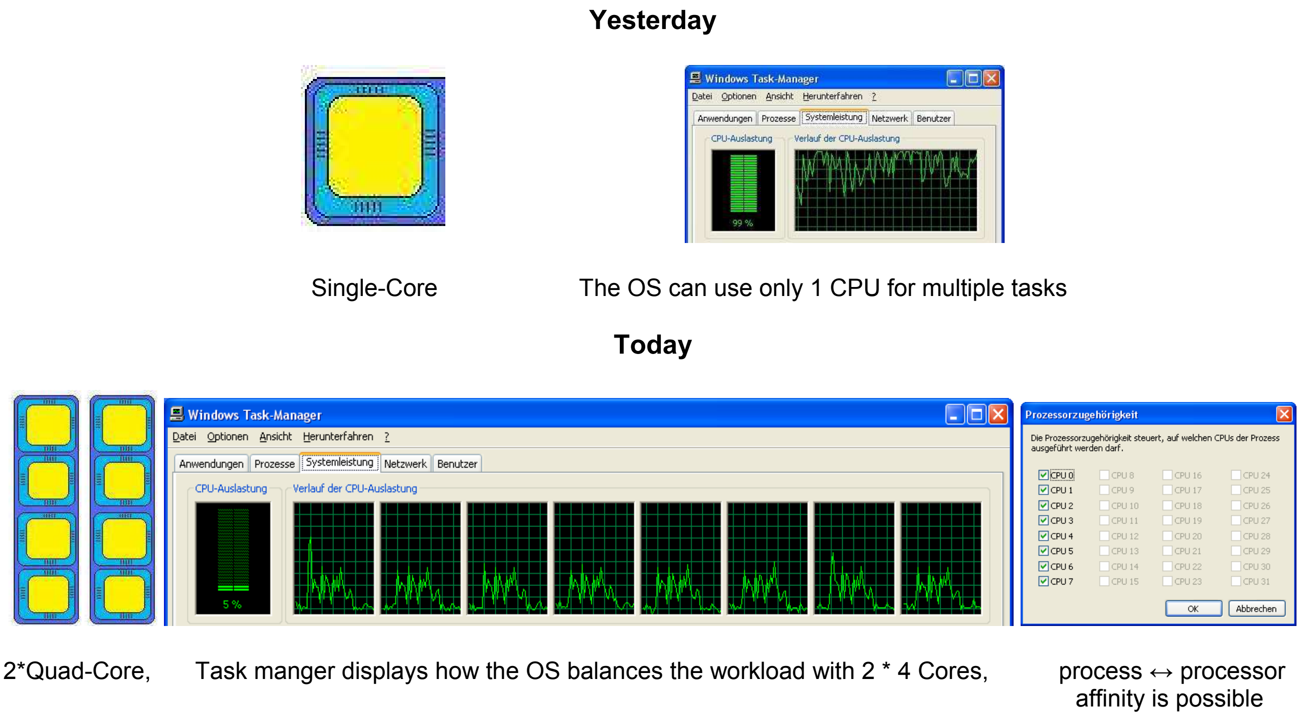

Figure 1: Today modern microprocessors can have up to 4 CPUs on one chip and the multiprocessing operating system (OS) can balance the workload. Also on multi-core machines the processor affinity setting controls which CPUs the process will be allowed to execute on.

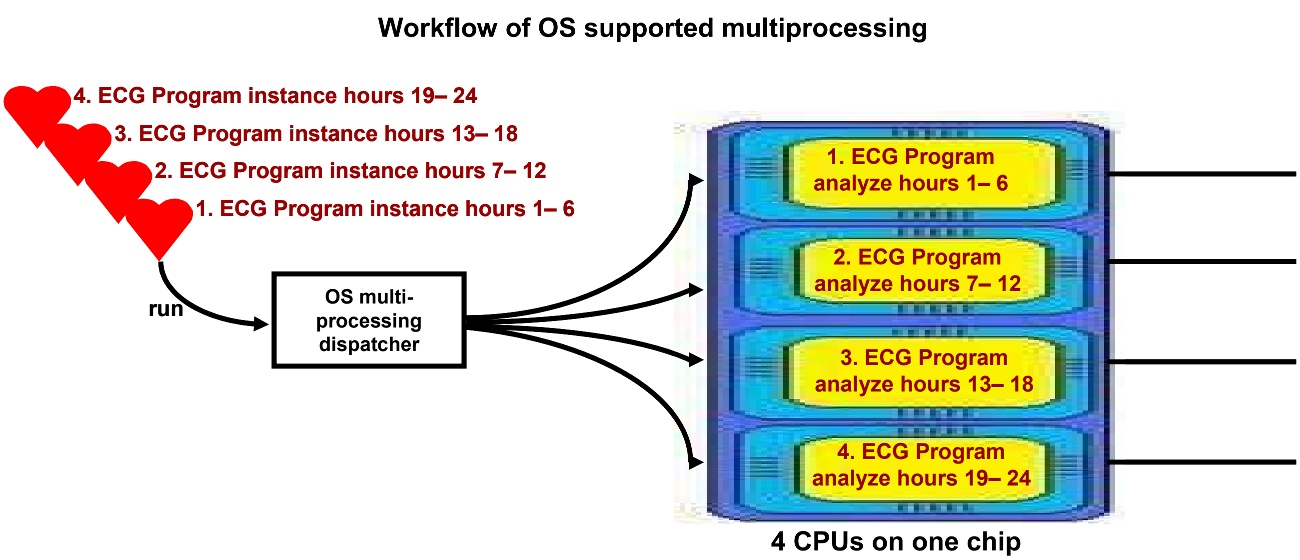

Figure 2: Advantage of multi-core chips. ECG analysis example using operating system (OS) supported multiprocessing.

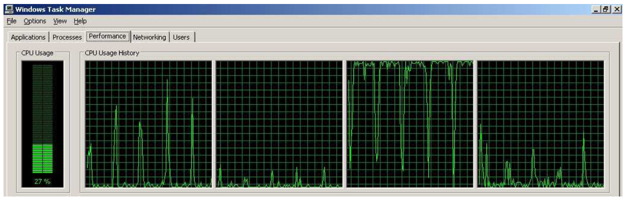

Figure 3: When one instance of the program is used, the processor executes using one CPU core while the CPU usage of the other cores is low.

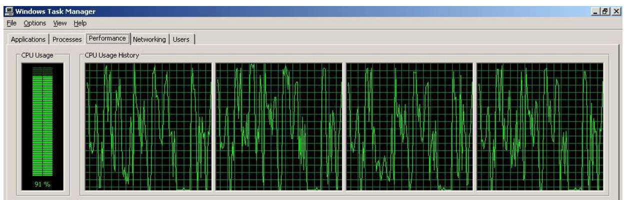

Figure 4: After segmentation of the ECG data into four 6 hour segments and executing the same program 4 times, the OS uses all CPUs for calculation.

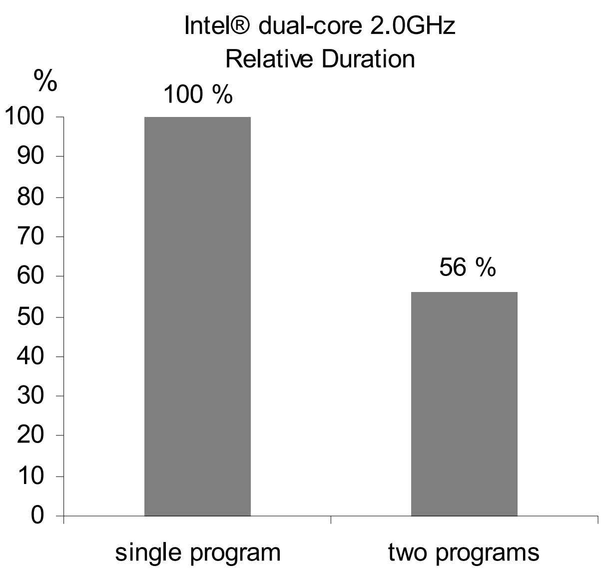

Figure 5: Relative shortening when running a program parallel on an Intel® Core Duo® 2 GHz Processor.

3. Results

As shown in Fig. 5 and 6 an ECG analysis program can take advantage of a dual-core processor and ran 44% or 46% faster when it was running twice while analyzing the first 12h in one program instance and hour 13h – 24h in a second program instance. Fig. 7 also shows the relative shortening of duration running the program in parallel up to 4 times on two dual-core processors. Two, three or four parallel instances of the same program shortened the time of the biosignal analysis by 43 %, 53% and 61 % respectively. It is notable that the relative effect of acceleration of this test was greatest when running two programs in parallel, while an additional third or fourth process, does not contribute more than about 10% each in shortening of scientific calculation. The limitation of speedup with an additional third or fourth run of the same program may be due to limitation in the memory bandwidth and an extensive hard disk access while analyzing the large volume of Holter data (approx. 510 MB).

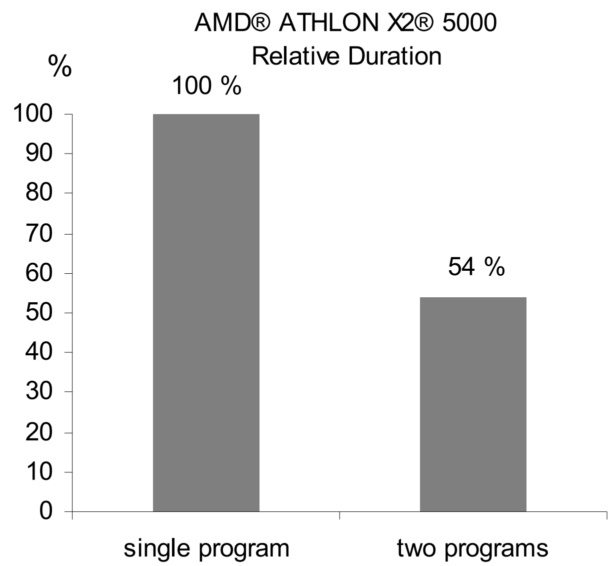

Figure 6: Relative shortening when running a program parallel on an AMD® Athlon X2® 5000 processor.

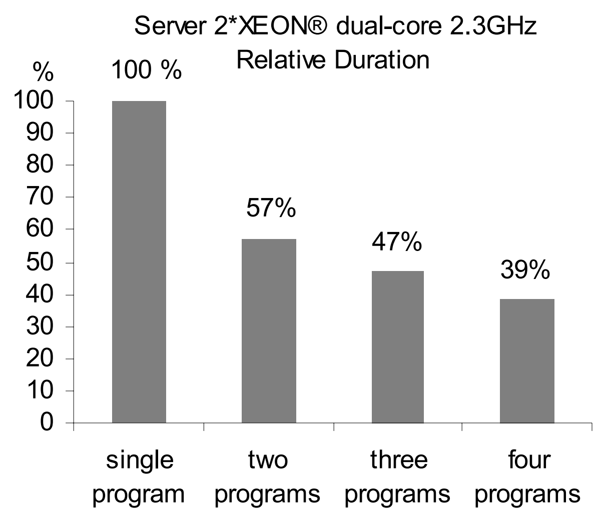

Figure 7: Relative shortening when running the same program parallel on a server with two Intel® XEON® dual-core processors.

4. Discussion and conclusions

Running the same program multiple times in parallel on modern multi-core processors can accelerate the numeric processing speed considerably. This can be a great benefit for the analysis of long-term ECG recordings with high data volumes. In the near future, the benefits of multi-core and modern IT techniques including parallel computing or distributed computing, should be considered for large volume ECG data sets. It is an advantage, when the operating system is able to speedup the calculations by multiprocessing such as shown in this publication. The performance and productivity benefits shown in this paper rely 100% on the multiprocessing capability of the operating system. The operating system does manage the processor affinity and controls the parallel execution of the different program instances. A limitation of our multiprocessing method is that the same program needs to be started several times and the 24h recording time intervals of the Holter has to be segmented for the individual program instances before program execution. An optimized method for new program developments is multithreading. Using multithreading instructions, software can get the maximum performance out of a multi-core CPU. For multithread programming the software developers needs additional software tools which are either compiler extensions (OpenMP)[1], language extensions (Parallel C) or special software library extensions (parallel Language Integrated Query; PLinq, PThreads)[2,3]. In general the maximal speedup that can be achieved by the usage of multiple CPUs and parallel processing is limited by the sequential fractions of a process like calculations or algorithms that can not be implemented as parallel processes. Another limitation for speedup is the communication between the multiple cores. The limitation of parallel computing is described in Amdahl's law [4]. For example, if 75% of a computer program can be parallelized, the maximum speedup by parallel processing would be four times [5]. Another limitation for speedup could be the overhead caused by synchronization and communication between the multiple cores. For scientific large data sets with the need of high performance computing, like long-term ECGs, it is interesting to investigate commercial graphics processors (GPUs) or the Cell Broadband Engine™ of the Playstation 3™ as a cost-efficient and very powerful computing resource. The Cell Broadband Engine™ of the Playstation 3™ game console has multiple so called synergistic processing elements [6] that can be used as parallel data crunchers and this capability is not only used for gaming but also for scientific calculations. The trendsetting Stanfords University Folding@home™ project does use Playstation 3™ processors in the private home of the gamers to do distributed scientific calculations worldwide [7]. The goal of this remarkable project is to research protein folding and misfolding to gain an understanding of how these are related to diseases like Alzheimer, Parkinson and cancer. The Folding@home™ project is platform independent and demonstrates the advantage of distributed computing for medical research. But considerable challenge remains in that software developers will need to write dedicated, multithreading programs for multi-core processors or distributed computer resources capable of simultaneously producing faster calculations.

References

[1] Chapman B, Jost G, van der Paas R. Using OpenMP : Portable Shared Memory parallel Programming. Cambridge (Mass.), London: The MIT Press; 2008.

[2] Rattz. JC jr. Pro LINQ: Language Integrated Query in C# 2008 (Windows.Net). Berkeley: Apress; 2007.

[3] Butenhof DR. Programming with POSIX Threads. Boston, San Francisco, New York, ….: Addison-Wesley; 1997.

[4] Amdahl G. Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. AFIPS Conference Proceedings (30) 1967:483-485.

[5] Amdahl's Law. [Online]. 2002 -2008. [cited 2008 Sept 12]. Available from: URL:http://en.wikipedia.org/-wiki/Amdahl_law

[6] Bartlett J. An indroduction to Linux on the Playstation 3. [Online]. 2007 Jan 3. [cited 2008 Sept 12]. Available from: URL:http://www.ibm.com/developerworks/library/pa-linuxps3-1/

[7] Pande V and Stanford University. Folding@Home.distributed computing. [Online]. 2000 - 2008. [cited 2008 Sept 12]. Available from: URL:http://folding.stanford.edu

Address for correspondence
Thomas Hilbel MD
Fachhochschule Gelsenkirchen
Fachbereich Physikalische Technik
University of Applied Sciences
Department of Physical Engineering
Neidenburger Strasse 43
45877 Gelsenkirchen
Germany

Tel.: +49(0)209 9596 750
Fax.: +49(0)209 9596 554
E-Mail: thomas.hilbel@fh-gelsenkirchen.de