Breaking the Bottleneck

Winfried Gerum

by Winfried Gerum

This column talks about software techniques which improve the performance of M programs. Being a frequent contributor to *M Computing*, some people think I might be the proper person to give you the best on this topic.

Sorry, there is no $MIRACLE to remove all your performance headaches.

Before talking about performance, I have to give you a serious warning! At last year's MTA conference someone did a survey citing several capabilities and asking respondents to rank them in order of importance. To my astonishment the survey did not mention reliability or any related "ability."

If a piece of software does NOT do its job CORRECTLY and RELIABLY under normal and exceptional conditions, there is absolutely no reason to ask about performance. If software does not do the task required, it does not matter how fast it executes. So your first priority should be correctness and reliability.

If reliability has been properly taken care of, the next point of importance is CHANGEABILITY and MAINTAINABILITY.

As we live in an era of accelerating change, you have to deliver solutions for tomorrow's problems by doing correct and reliable changes on today's software. If you cannot secure that, and you worry instead about today's performance, then think: unless you continue to serve your customers with reliable software solutions you soon might be out of business. Then it does not help if people remember that yesterday your software did run faster than anything under the sun!

In order to achieve these objectives you might buy fancy tools labeled according to the latest fads. But ultimately it boils down to the fact that you (or your programmers) have to develop a quality-conscious mindset. We do not need structured programming or object-oriented programming. But structured thinking and "object" oriented thinking are helpful to achieve the quality goals.

In my experience, performance comes almost automatically if you remove the hunt for ever more performance from your thinking and if you replace that by the quest for quality. You can only guarantee correctness and reliability if you completely understand what is going on. Having the full understanding usually you get an elegant algorithm that is not only reliable, but superior in performance! This is so because nothing unnecessary is done. The golden rule is "AVOID ALL SIDE-EFFECTS WHATSOEVER."

When talking about performance, people usually think about the speed of data processing systems (i.e. software and hardware). We continue to do so, because in the early days of EDP, this kind of performance was a very serious issue. Computers were very expensive and people negligibly cheap. Today, people are expensive and the hardware has become negligibly cheap. The logical conclusion should be that today we should worry about the performance of people instead of performance of machines.

Tuning people for performance is an art just as tuning of hardware and software has been. Time is too short to even try to get into this topic of human performance. Nevertheless I urge you to think about it!

The famous Volkstorf book *125 Ways to Make Your MUMPS Apps Run Faster* is not completely out of date, as it shows that there is no single way to make systems run faster. A lot of the basic ideas presented by Volkstorf are still valid. But most of the things he says about programming are obsolete. It no longer makes sense to fine-tune software to the idiosyncrasies of a particular version of one M implementation.

What is efficient in one system might be completely inefficient in a different system (e.g., implementing the argumentless NEW you can either copy the complete local symbol table to a stack or you can leave the local symbol table as it is and start a new one, just changing a pointer. Both methods have been implemented. They have dramatically different execution times for the same command. Fine tuning for one of these implementations gives you a serious performance penalty if you switch implementations. Not an issue for you, because you never changed your implementor? But your implementor may decide to re-implement a feature in a different way, messing up your fine-tuning).
Obviously, a lot of people are still

interested in improving performance. OK, a job has to be done reliably and correctly. But it is of no use if the result comes late. Imagine a 24-hour weather forecast that needs more than 24 hours of computation time! In such time-critical missions, where you need all the performance you can get, just re-read Volkstorf! Usually just a small percentage of a software package does the bulk of the work. To achieve user satisfaction of the whole package it might be sufficient to take a closer look just at the most frequently used pieces of software.

You don't dig your garden with a spoon nor do you use a spade to build a highway. One should always look for the most appropriate tool for the problem at hand. Therefore, in the M Technology environment you should be aware that while M is the language of choice for a lot of areas, it is by no means superior everywhere.

Today's standard solution to EDP performance problems is buying more of the latest hardware. Just an example: an M statistics package for a big retailer used to run for a whole weekend on the old VAX. Using an alpha with 500 MB of RAM disk the same software did the same job in just under two hours. It is unlikely to get a similar improvement in speed by programming tricks. **M**

Winfried Gerum is with Winner Software GmbH in Röttenbach, Germany. You may contact him by phone at 011-49-9195-940022 or by fax at 011-49-9195-940030.

Student Research Competition

The M Technology Association announces a rewarding challenge for students: The 1997 Michael Distaso Memorial Fund Competition for innovative and outstanding research in the M language.

$1,000 cash award and one-year membership in MTA.

Present your paper at the 1997 MTA conference in Boston, travel expenses paid.

Don't miss out! Deadline for submission of papers is February 17, 1997.

For Information contact:

M Technology Association
1738 Elton Rd.
Silver Spring, MD 20903
Phone: 301-431-4070
Fax: 301-431-0017
E-mail: MTA1994@aol.com