From: Uzi Vishkin [mailto:vishkin@umiacs.umd.edu]
Sent: Thursday, August 18, 2011 5:15 PM
To: amtech
Cc: Lambis, Barbara
Subject: AMTech Comments

Dear sir/madam,

The main revision, or rather refinement, I propose to the AMTech program is *to allow even a new start-up to play the role of industry-lead for a new AMTech consortium*.

The attached write-up provides context to my comment.

Sincerely

Uzi Vishkin, Professor
The University of Maryland Institute for Advanced Computer Studies (UMIACS) and Electrical and Computer Engineering Department
A. James Clark School of Engineering
Room 2365, A.V. Williams Building
College Park, MD 20742-3251
Phone 301-405-6763. Shared fax: 301-314-9658
Home page: http://www.umiacs.umd.edu/~vishkin/

Input to NIST on how to best structure the AMTech program

Summary The main revision, or rather refinement, I propose to the AMTech program is *to allow even a new start-up to play the role of industry-lead for a new AMTech consortium*.

Introduction

For context, I will use below a concrete manufacturing technology example I am familiar with as a basis for my proposed revision. I would like to incorporate here the appendix at the end, which is a verbatim copy of my white paper that was posted recently on the NIST web site. The white paper points out that application software development is a key form of manufacturing and that its productivity for modern commodity computers, which are all 'multi-core machines', has been greatly diminished in all such commercially available computers. The white paper suggests is that it has become a national interest to restore the productivity of application software development.

Proposed revision in AMTech program

The problem in addressing the software productivity restoration manufacturing technology example is that the industry that would directly benefit from significant improvements in manufacturing productivity is the application software industry (ASI) – a different industry than the chipmakers industry, the industry that is in a position to bring about such improvements.

The (few) established chipmakers have no shortage of funds. However, it is hard to make a case to them that they will benefit in the short term from addressing this ASI problem.

However, the ASI is also unlikely to lead the effort: 1. It is against the ASI culture (and with no recent precedent) to push for a new general-purpose hardware platform. 2. A whole new ASI should evolve by 'application dreamers', only after the new computer system will be available. In other words, software productivity restoration program will restore the climate for innovation that allowed past innovators, such as Steve Jobs over 30 years ago, to create new applications on shoe string budget in their garage in the higher software productivity days of uniprocessing.

What is needed is a first-of-its-kind convincing prototype of a many-core machine whose programming can be handled effectively by most programmers, perhaps with some modest proper training.

In my opinion, a new start-up company dedicated to the software productivity restoration objective is more likely to deliver the needed outcomes than established players.

Besides funding of at least a beta version of the prototype by such new start-up company, a revised AMTech program should allow a consortium led by a start-up with sub awards to fund research by universities, government laboratories, and U.S. businesses.

Overall the crux of the vision of the AMTech program is fully applicable to the software productivity restoration example. It is an industrial manufacturing opportunity, where the development of precompetitive enabling manufacturing technologies for long-term economic growth and job creation is badly needed. Both the development of the technology and the creation of conditions necessary for efficient transfer of the technology pose non-trivial challenges. Overall, they provide an ideal opportunity for targeting Federal assistance to eliminate critical barriers to innovation, increase the efficiency of domestic innovation efforts, though more productive software development, and collapse the time scale to deliver new software application products and services based on companion scientific and technological advances.

APPENDIX Application software for many-core computers: a manufacturing technology example

The text below is a verbatim copy of the one-page paper U. Vishkin. Restoring software productivity crucial to economic recovery: The multi-core dilemma. White paper, the Technology Innovation Program (TIP), National Institute of Standards and Technology (NIST). Input to help TIP focus the TIP program on areas of critical national need. July 2011, http://www.nist.gov/tip/wp/pswp/upload/266_restoring_software_productivity_crucial_to_economic_recovery.pdf .

Traditionally, Americans look to the manufacturing sector for bettering the recovery prospects of the economy. Software production is the quintessential 21st century mode of manufacturing. But, if most programmers are unable to design software for mainstream computers, this could preempt such prospects. Yet, this is exactly today's reality.

Today's mainstream desktop and laptop computers are all so-called multi-core machines. Instead of using a single processor (the 'computer brain' also called 'core'), desktops and laptops began using several 'cores' starting around 2005. The number of cores in each computer is expected to continue doubling every 1 to 2 years. Programmers are now expected to program these multiple cores to solve together a single task automatically. In fact, technological constraints forced the transition to multi-cores, as single-core computers started overheating. This transition caught chipmakers and much of the research community unprepared. Consequently, chipmakers are now busy building and designing multi-core processors that most programmers cannot handle. The industry now finds itself coping with the challenge of how to build and program a completely new type of machine, without ready knowledge.

Government investment in this general domain is limited to seeking innovative ways of educating computer science majors to program these hard-to-program multicores. These are legitimate efforts, but given current knowledge, expedient, satisfactory results are unlikely. Machines must be built and tested for ease-of-programming before they are placed in the market, not after. All current government programs dedicated to building better computers are limited to 'beyond silicon' technologies that could be relevant for the 2020s but not earlier, or to extremely large supercomputers for scientific applications.

All of these are, of course, legitimate efforts, but for the nearer future, the full faith of the economy is being placed on the few chipmakers that currently dominate the market. However, these chipmakers appear unable, or unwilling, to address the problem that matters most to the recovery of the US job market.

A new government program to facilitate commercial many-core machines whose programming can be handled effectively by most programmers, perhaps with some modest proper training, is needed.

Uzi Vishkin is Professor of Electrical and Computer Engineering at the University of Maryland Institute for Advanced Computer Studies (UMIACS).