

9 August 2008

By: Marius Oiaga, Technology News Editor

[The Future of Windows Is Manycore/Multicore](#)

Parallel computing will reshape Microsoft's platform



Windows Logos
Microsoft

Microsoft has little say in the matter, as does the entire ecosystem of software developers that will have to step up their game and fall in line with the evolution of hardware, further and further into the territory of parallel computing. However, the Redmond company is indeed ready to ride the wave of manycore/multicore, and has been gearing up for the moment when 8, 16 and 64-core machines will become the standard. The consensus is that, in as little as four years from now, mainstream processors will be massively parallel. At the [Microsoft Financial Analyst Meeting 2008](#), the company's Chief Research & Strategy Officer, Craig Mundie, indicated that Windows was following along with the transition to parallel computing. "We actually started our series work to anticipate this change here in 2002. We've been investing to get ready for this real fundamental shift for quite a while. The OS will have to change to deal with the heterogeneous many-core world. Where you are today is just a transition point from single-core machines to - I'll say at a steady state point where you're going to have a lot of processors on a chip, but they won't all look the same," he stated. Of course, the Redmond giant is not saying all that much about how Windows will be tailored to manycore/multicore processors, but a consistent collection of project codenames and Microsoft Research initiatives reveal that efforts are being poured into embracing parallel computing. And such scenarios are not limited to the company's current proprietary platform. **The Anchor of Single-Core Programs** Following the transition of Bill Gates out of his role as fulltime Microsoft Chairman, while Ray Ozzie took on the title of Chief Software Architect, Craig Mundie became responsible for looking beyond the immediate horizon, and preparing the company for the future. Microsoft 2.0's forays into parallel computing fall into Mundie's backyard, but the manycore/multicore vision is not limited to Windows. In fact, the Windows client is likely to be one of the last software products to make the jump, as it is firmly anchored into the single-core past by a world of legacy applications. "The many-core world today is one where it's still a homogeneous, symmetric, multiprocessor architecture. Frankly, the operating system, Vista, does a great job on those [multiprocessor machines](#). And, in fact, it's the same technology that runs the 128-processor data center servers. The problem is the applications. In the data center, we know how to take workloads and smear them out across those things, and you get good efficiency. But all the world's programmers were given a set of tools and a set of techniques for the last 30 or 40 years where when they wrote an application, they squashed out all the parallelism that was intrinsic in it. And it's that serial manifestation that limits our ability to use those additional cores," Mundie added. The issue is not whether Microsoft can make the jump to multicore/manycore architectures or not. The real issue is how the company will accomplish the transition while having to drag along an immense environment of legacy software products and perpetuating compatibility. Windows cannot afford to jump ahead into the parallel computing era and to leave behind all the adjacent items that make it a platform, in the hope that they will catch up. This because consumers are not necessarily dependent on Windows, but on the environment of hardware products and software solutions orbiting around the operating system. "And so today it's the applications that haven't gotten there, not so much the operating system. Both will evolve. The fundamental breakthrough that's required is to make it easy enough for people to do complex parallel applications that can use a lot of CPU in a client device," Mundie commented. **Hammering Away at Manycore/Multicore Ready Products** Just because Microsoft is not touting its parallel computing efforts, it doesn't mean that they are inexistent. In fact, many groups inside the Redmond company are exploring, building, planning, incubating, prototyping, researching,

testing a variety of solutions. And this is happening not only in Microsoft Research, but also in the more "mundane" divisions. The work currently carried out by the likes of the Platform and Security Incubations team, the Concurrency Development Platform Team, the Parallel Computing Platform Application team, and the Parallel Computing Platform team offers illustrative examples in this context. Beyond laboring to integrate [hypervisor-based virtualization technologies](#) into the Windows client, the Platform and Security Incubations team is also getting ready for the need to adapt operating systems Windows and [Singularity](#) to heterogeneous, many-core processors. And, taking into consideration that there is a strong multi-core focus on non-Windows Singularity, the same should be the case with [Midori](#). "The two- or four-core CPUs found in today's machines will give way to processor chips with dozens of cores on them, some of which may not even share instruction sets let alone implementations. At the same time, hypervisors currently used for server consolidation will become a standard element of the client desktop, giving us the ability to partition the physical hardware among multiple virtual machines, some specialized for particular tasks, executing in parallel," explained a member of the Platform and Security Incubations team. Throughout Microsoft, there is a strong emphasis placed on the year 2012, made virtually synonymous with the advent of manycore/multicore processors. And the generalized scenario for which the software company is gearing up is that of the client hardware platform becoming massively parallel, which will drive Windows after it, and which, in turn, will force the migration of Windows-centric software. **Massive Parallelism Will Lead to Experiences You Could Once Only Dream of**"Massive parallelism will lead to experiences you could once only dream of. To enable these next generation experiences, we must develop simpler ways of expressing parallelism in code and provide tools to make developers successful at doing so," revealed a member of a v1 team associated with the Concurrency Development Platform. "Our team's mission is to enable the masses of developers to readily exploit parallelism, and our roadmap begins with the next version of Visual Studio."So it's not just Windows, it's much more. But [Visual Studio](#) is just another example in the migration to multicore/manycore. The Concurrency Development Platform Team is building technology associated with runtimes, programming models, managed and native libraries and languages designed to signal the readiness for the forthcoming many-core platforms. And of course, there are always the efforts from the Parallel Computing Platform team, yet another v1 group. "The Parallel Computing team will find success by applying research and incubation results to innovate 360 degrees in the software stack - libraries, languages, runtime systems, and code generation - and are actively working on both native and managed technologies. Managed technologies include PLINQ (Parallel version of LINQ), Task Parallel Library (formerly known as C# futures), a concurrency runtime with fine-grained work scheduling algorithms, concurrent data structures, deeper library support for parallelism, transactional memory, etc., and closely collaborate with other teams in Microsoft like MSR, Windows, and CLR. Native technologies include building a new user mode cooperative scheduler with fine grained work scheduling, resource manager to support multiple schedulers, a new asynchronous messaging model and parallel patterns libraries (example parallel for)," explained one of its members. In the end, what the Parallel Computing team is building is a massively parallel and asynchronous ecosystem mechanism, starting with .NET and native code, and ending with intelligent runtime scheduler support. All for the sake of enabling developers, partners, and customers to leverage the benefits of parallel processors. **And It's All Pointing to Windows**But make no mistake about it, as diversified as Microsoft's efforts to keep the pace with the evolution to manycore/multicore are, from [Tahiti](#), [Midori](#), and Singularity to Visual Studio, in the end, it all comes down to the Windows platform. The transition to massively parallel mainstream processors by 2012, two years after [Windows 7](#), and just about when [Windows 8](#) should hit, will find the Windows operating system and the software solutions designed to integrate with it dealing with 10 to 100 times more processing power than what Vista can manage today.