CS267 Assignment 0: Describe a parallel application

Zhongyin Hu, M.S., Mechanical Engineering

I am currently working in Computer Aided Design and Manufacturing Lab (CADML) advised by Prof. Sara McMains. My aim behind taking CS267 is to learn parallel programming fundamentals, get acquainted with supercomputers, and understand implementation details for effectively programming GPU and CUDA.

Parallel Computing Application in CAM software

In recent years, Computer-Aided Design and Computer-Aided Manufacturing (CAD/CAM) that involve complex, large-scale data processing have reached increasingly high levels of complexity. Consequently, processing could take hours and sometimes even days or weeks on modern hardware. Accelerating the processing functions in these software packages not only increases user satisfaction, but also enables higher accuracy, better decision-making, and more efficient work procedures in organizations using the software. Software acceleration has thus become a high priority for software organizations in CAD/CAM field.

Using parallel computing to accelerate highly complex computational processes is not a new concept. This approach has been tested and proven, and with the recent influx of affordable multicore and General-Purpose Graphics Processing Unit (GPGPU)-based technologies, it is more relevant now than ever before.

However, choosing the right technology to accelerate time-consuming computational processes is far from a straightforward decision. The various options for implementing a parallel or distributed system offer substantial differences in the resulting acceleration potential, as well as in direct development costs and indirect/long-term costs (maintenance, infrastructure, energy, and so on). This is particularly true when considering platforms for migrating existing code to a parallel or distributed architecture. Choosing a less-than-ideal system may incur dramatically increased costs – both direct and indirect – compared to a better alternative[1].

A major new capability in CAM software: Parallel Computing

Computing several threads simultaneously reduces programming and calculation times for complicated parts. For example, SolidCAM 2012 has claims of up to 70% savings in machining time and dramatic tool life increase with its new time saving "Parallel Computing" [2]. Suited for the moldmaking industry, version 21 of Sescoi’s WorkNC CADCAM software also features parallel computing algorithms for improved calculation speed[3]. Another CAM software incorporating this technology is PowerMill[4]. It claims leveraging parallel computing has significantly reduced toolpath calculation times in the most recent version.
Incorporating parallel computing into CAM software provides three important benefits:

- The ability to calculate or edit one set of tool paths in the foreground while the program calculates another set in the background, with minimal degradation in processing speed. Known as background processing, this capability eliminates the need to wait for each calculation to be completed before preparing for the next operation. The user simply adds tool paths to a queue, and the program will calculate each in sequence. User can start parallel simulation and, while it's simulating in the background, he can in parallel continue to work defining additional operations. User can start G-code generation and, while it's generating G-code, he can in parallel continue to work defining additional operations.

- The ability to split calculation of a single complex tool path among multiple processing units to reduce overall calculation time. Known as parallel processing, this happens automatically—the user doesn’t need to do anything to activate it.

- The application of parallel processing to both foreground and background calculations to provide even greater performance gains.

(a) Percentage of CPU Usage
Parallel computing in CAM software packages can be done in two ways. It can be done locally on user computer, using the power of multi-threading on multi-core CPUs. The other option is on a remote computer, which means parallel computing is executed on the external computer on the network.

Despite of the great benefits of using parallel computing in CAM, there also exist challenges for its future development. Performance gains depend heavily on the specific tool paths and machining strategies employed by the user. It’s still a difficult question to answer exactly how much faster it will go because it really depends on what specific parts is cut and what operation is done. It’s noted that the current version of PowerMill provides only a fraction of the performance gains expected to be available in future upgrades. It’s also noted that some CAM software companies (for example, Delcam) develop all their own codes, as opposed to relying on libraries from external suppliers, so it has the freedom to rewrite virtually any part of the program that it deems suitable for parallelization. There is a large space for improved algorithms to further speed things up.

Reference
