Eliot Cowley 2/2/15

Professor Smith CMPU-377 Parallel Programming

“Software and the Concurrency Revolution” Summary

The article “Software and the Concurrency Revolution” by Herb Sutter and James Larus of Microsoft warns that as we approach the limits of how fast we can make single-processor computers and start adding multiple cores to compensate, changing the way we program software to become more concurrent in nature will become vital to ensuring good performance. Programmers are still largely taught to program sequentially, and form a mindset that is often difficult to break out of. Further, human beings in general find it difficult to think in terms of concurrency. It is not a simple concept to grasp, and makes programming even more difficult.

Today’s programming languages, such as C, C++, and Java, are not conducive to easy parallel programming. They were designed with sequential programming in mind, and contain only low-level techniques to deal with concurrency, such as p-threads. With nearly all modern machines running on multiple cores, the designers of these languages will have to come up with ways to make programming in parallel easier to understand and implement. We cannot simply start from scratch—we have too many talented people who have invested their lives to develop certain skills, so we must instead somehow adapt current languages to meet the needs of the concurrency revolution.

In order to take advantage of the influx of new, powerful multicore machines, programmers must learn to think in new ways. We must stop limiting ourselves to programming sequentially and open our minds to the possibilities of concurrency. It will require a great deal of effort, but the potential is enormous.