Moore's Law Goes Multicore:

The economic and strategic consequences of a fundamental change in how computers work

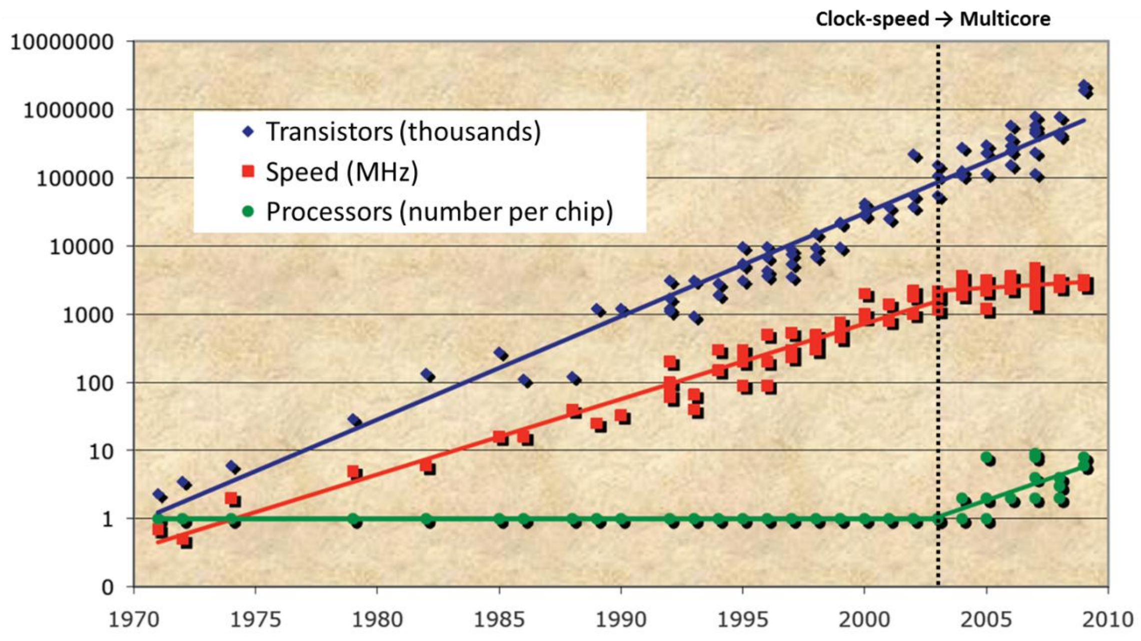
Neil Thompson, Berkeley Haas School of Business

When Intel announced that they were no longer focusing on speeding up their chips, it represented a fundamental shift in how hardware improvements impact software. Only two years earlier, in 2002, they had claimed that the Pentium 4 chip would scale to 10 GHz, a speed improvement consistent with what software designers had come to expect. But, in 2004, heat build-up in their chips caused them to abandon these plans in favor of putting multiple processors (or 'cores') on the same chip. Other chip manufacturers soon followed.

This changeover had a profound impact on software performance. Previously, faster hardware speeds translated directly into faster software computation. Afterwards, only software that could take advantage of multiple processors would get this benefit – and not all of them could. Computer scientist Herb Sutter summarized this, saying "most current applications will no longer benefit from the free ride without significant redesign".

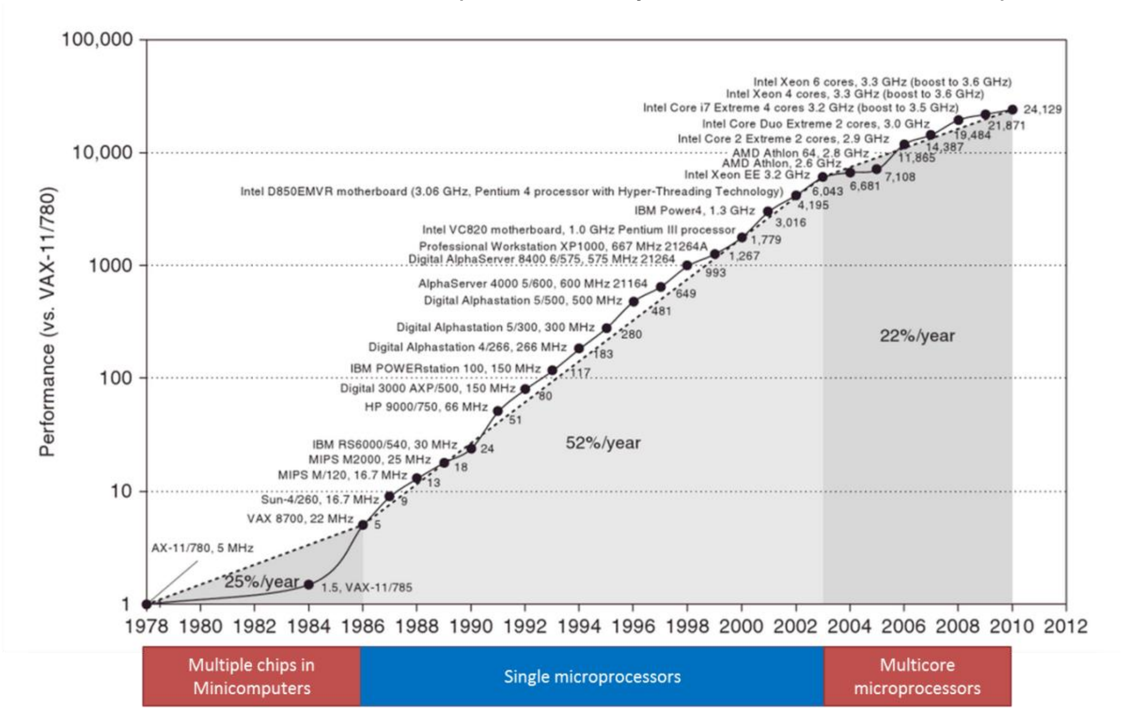
This paper explores how the switch to multicore computing is impacting firms and their productivity. It takes as a starting point the shock of Intel abandoning processor speed-ups so shortly after promising to continue a 30+ year history of them. It then argues that this changeover will impact firms differentially based on which software they are using and how well it can be run on multiple processors. Finally, it uses the differences in how firms are impacted to address the long-elusive question of the causal impact of how information technology adds to firm productivity.

The graph below, based on data gathered by Stanford computer scientists, shows Moore's Law, the long-persistent trend of increasing transistor densities in computer chips. It also shows that until 2004 these increases translated directly into chip speed-ups, but that afterwards it led to multiple processors.



The consequence of this switch to multicore can be seen in software performance, for example in the performance of SPEC, a widely used benchmark that tests how fast processors can complete a series of important computational tasks. The graph below, produced by computer scientists at Stanford and Berkeley, shows the dramatic improvements in benchmark performance in the single-processor era, and the subsequent slow-down of performance gains after the switch to multicore:

SPEC Benchmark Performance (from Hennessy and Patterson, footer added)



Source: Computer Architecture, A Quantitative Approach by Hennessy and Patterson

With some computer programs getting smaller gains in performance after 2004, it is natural to wonder whether firms that rely on these programs might also have worse performance (in this case, smaller productivity growth). This is the key question posed by this paper: do we see a divergence in productivity between the firms that rely on software that takes advantage of multicore computing, and those firms that cannot.

This paper uses three types of data to address this question. The first is data on the software that firms use. This consists of two establishment-level panel datasets, one for Sweden and the other for the United States. The second is a survey of computer scientists which I ran in collaboration with the Berkeley Parallel Computing Lab. This survey determines how much parallelism is present in different types of software. The third is a panel of firm-level productivity data. I currently have this for the Swedish firms (public and private), and am working on getting this for the U.S. firms.

Initial results from analyzing the Swedish data support the predicted direction and statistical significance of this effect. In the next few months, this will become more concrete with a preferred specification and robustness checks. Over the longer term, in addition to extending this analysis to the productivity of U.S. firms, I will also extend it to cloud computing, as parallelism is also needed in order to scale up software applications for the cloud.