|  |  |
| --- | --- |
|  | Your primary problem is software not written for multi-core. If your software supports multi-core then multi-core is better than clock speed.  Look at Jeff Atwood's excellent article on [Choosing Dual core or Quad Core](http://www.codinghorror.com/blog/archives/000942.html).  for most software, you hit a point of diminishing returns very rapidly after two cores. In [Quad-Core Desktops and Diminishing Returns](http://www.codinghorror.com/blog/archives/000655.html), I questioned how effectively today's software can really use even four CPU cores, much less the inevitable eight and sixteen CPU cores we'll see a few years from now.  You are answered here (highlight copied from Jeff's article),  However, there were some surprises in here, such as Excel 2007, and the Lost Planet "concurrent operations" setting. It's possible software engineering will eventually advance to the point that clock speed matters less than parallelism. Or eventually it might be irrelevant, if we don't get to make the choice between faster clock speeds and more CPU cores. But in the meantime, **clock speed wins most of the time. More CPU cores isn't automatically better**. Typical users will be better off with the fastest possible dual-core CPU they can afford. |