Moore's Law and its Impact on Software Development

Moore's law is the hypothesis that transistor density doubles every 2 years. This is based on historical trends. This is not a physical law, rather an observation. The expected exponential increase in density, leads to exponential increase in transistor speeds, as smaller transistors switch faster.

However, this is no longer the case. There are a few physical limitations. First being power consumption. Transistors consume power as they switch. Additionally, increased transistor density leads to increased power consumption. Increased power leads to increased temperatures, which requires more sophisticated and powerful cooling solutions. Increased power and temperatures are therefore are limiting factors in terms of electronics design.

In concrete terms, the dynamic power consumption of transistors are described by:

$$P = \alpha * CFV^2$$

α is percent of time switching
C is capacitance (related to size)
F is the clock frequency
V is voltage swing (from low to high)

The important thing to note is the V, or the voltage swing, whose square is proportional to the power dissipated. As described by the Dennard scaling theory, the voltage swing should scale with the transistor size. Therefore, as density increases to meet Moore's Law, the transistors must be reduced in size, and therefore use smaller voltages. The key issue here is that to meet physical laws, the threshold voltage to turn the transistor on can only be so low. The reduced voltage swing also introduces more noise in the signals, which leads to errors.

Additionally, there is leakage power to consider, the power which is dissipated when the transistor is off. Smaller transistors lead to thinner insulators, which leads to increased leakage power. This leads to additional problems.

The conjunction of issues described above lead to what is called the power wall. To remedy this lack of performance improvement from vertical scaling, one may scale horizontally instead: by adding more processors to the system. As a result, concurrent programming which must now make use of processors working in parallel become very important. The responsibility of achieving improved performance is now shared between hardware, and software, as the hardware needs to be adjusted to account for the diminishing adherence to Moore's Law.