How to hold CPU vendors’ feet to the Software Spiral fire?

Andy Grove (Intel’s business leader till 2004) termed “software spiral (SWS)” the exceptionally resilient business model behind general-purpose CPUs. Application software is the cornerstone of SWS: Code written once could yet benefit from performance scaling of later CPU generations. But, after nearly two decades of multicore CPUs domination, the jury is out: extension of SWS from serial single core CPUs to parallel general-purpose multicore CPUs has failed, missing goals stated by CPU vendors themselves.

The serial RAM algorithmic model has long been disfavored by architects due to abstracting away architecture features. However, as long as it was part of the specs for single core CPUs, they had to bite the bullet and continue supporting the RAM model. The transition to multicores reversed the flow from algorithm-model-driven specs to architecture. Vendors started forcing algorithmicists and application developers to bend over backwards to meet manycore hardware. The result: too few program today’s manycores for parallelism. Ever changing hardware designs further exacerbated the problem. Not only that convergence to a manycore software spiral was never achieved, it disappeared from vendors’ narratives.

It is finally time to revert back to what worked prior to derailment of the software spiral. General purpose parallel algorithms representing millions of application programmers must drive specs for manycore CPUs. Extensive hardware and software prototyping that my team did at UMD has demonstrated technical feasibility. There is also new hope from the business side. Competition among CPU vendors is getting fierce for the first time since the 1990s. However, the main challenge is to establish an appealing application-based proposition. Introspection on the most impactful recent applications will lead to constructive lessons towards such a proposition.