As the HPC community starts focusing its efforts towards exascale, it becomes clear that we are looking at machines with a billion way concurrency. Although parallel computing has been at the core of the performance gains achieved until now, scaling over 1,000 times the current concurrency can be challenging. As discussed in this paper, even the smallest memory access and synchronization overheads can cause major bottlenecks at this scale. As we develop new software and adapt existing algorithms for exascale, we need to be cognizant of such pitfalls. In this paper, we document our experience with optimizing a fairly common and parallelizable visualization algorithm, threshold of cells based on scalar values, for such highly concurrent architectures. Our experiments help us identify design patterns that can be generalized for other visualization algorithms as well. We discuss our implementation within the Dax toolkit, which is a framework for data analysis and visualization at extreme scale. The Dax toolkit employs the patterns discussed here within the framework’s scaffolding to make it easier for algorithm developers to write algorithms without having to worry about such scaling issues.