Incorporating life sciences applications in the architectural optimizations of next-generation petaflop-system

David A. Bader ∗ Vipin Sachdeva
Department of Electrical and Computer Engineering
University of New Mexico, Albuquerque, NM 87131

Abstract
Advances in experimental techniques have transformed biology into a data-intensive science, with a rapid explosion of data at the genomic and proteomic level. Few comprehensive suites of computationally-intensive life science applications are available to the computer science community for optimization of current high-performance architectures specifically targeted towards the computational biology applications. BioSplash represents a wide variety of open-source codes spanning the heterogeneity of algorithms, biological problems, popularity among biologists, and memory traits, gearing the suite to be of importance to both biologists and computer scientists.

1. BioSplash

The BioSplash codes are selected from the following areas: Sequence alignment (pairwise and multiple), Phylogeney reconstruction, Protein structure prediction, Sequence Homology and Gene-finding. We have included parallel codes when available; 4 of the 8 codes in the suite are multi-threaded.

2. Experimental environment: IBM Mambo and Apple G5

Our experimental environment consists of the IBM Mambo simulator and the Apple PowerMac G5 (with IBM PowerPC 970 processors). Mambo [3] is an IBM proprietary full-system simulation tool used to design the Blue Gene and the p-Series within IBM, and is being used for the DARPA High Productivity Computing Systems program [4] aimed at bringing sustained multi-petaflop and autonomic capabilities to commercial supercomputers. IBM’s PowerPC G5 processor [1] is a 64-bit, dual-core state-of-the-art processor designed for symmetric multiprocessing. We created a size varying class of input sets for each code, with the inputs categorized as small, medium and large. For application runs on Mambo, small datasets...
are used due to the overhead of simulating, while for Apple G5, relatively long-running instances of the datasets are described and are for instance among the longest in Genbank and Swissprot, database searches were performed against complete databases such as the Genbank, Swissprot, and PFAM, databases.

3. Performance Analysis

An advantage of the Mambo and MONster tools is the “live graph” capabilities that report accumulated data not only at the end of each run, but also at chosen regular sampling intervals. Using such graphs, we correlate the performance data in phases of the application, and suggest optimizations targeted at separate regions of the application. Through our dual platform analysis, we do a cross comparison and suggest system design parameters for machine configurations that may improve the performance of these codes on next-generation computing systems.

In this abstract, we briefly show the impact of our research with an example of ClustalW, a popular multiple sequence alignment program. Additional results are available in [2]. In the clustalw livegraphs, we find an interesting observation: the instructions per cycle increases in the last phase of the application, even though L1 data miss rate (right plot) increases in the same phase; this increase is due to the fall in branch misprediction rate, implying that the performance of the last phase of clustalw is more closely related to branch mispredicts than L1 data miss rate.

A significant part of our research correlates the performance metrics at the machine level with the phase of the higher-level algorithm. Clustalw livegraph plotted for the first and the third phase shows an order of magnitude higher rate of arithmetic instructions compared to the second phase, denoting the larger time complexity of the algorithm in phase one and three. L1 data cache misses are higher in the second phase, due to unpredictable access pattern; the sequences with the lowest distance are joined together in the guide tree, with the other sequences recomputing distance to the joined sequence. The heuristic is repeated, until all the sequences are joined together into a guide tree formation.

We have recorded a wide variety of instruction and memory level hardware counters for each code using CHUD tool on Apple PowerPc G5, and MAMBO; fully detailed analysis and all graphs for MAMBO and Apple PowerPC G5 are included in [2].

References