Parallel Performance Wizard: A Performance Analysis Tool for Partitioned Global-Address-Space Programming Models

Hung-Hsun Su, Adam Leko, Dan Bonachea, Hans Sherburne, Max Billingsley III, Alan D. George

PPW Overview

- Computationally intensive parallel applications are constantly being developed in many scientific fields using parallel programming models ranging from:
  - Message-passing based: MPI, etc.
  - PGAS based: Unified Parallel C (UPC), SHMEM, Co-array Fortran (CAF), Titanium, etc.
- Performance optimization is often needed to minimize the application's overall execution time
- Several performance analysis tools available to facilitate the optimization process
  - However, majority of the tools support MPI with only a few supporting PGAS models
- Parallel Performance Wizard (PPW) was designed and developed to improve performance analysis tool support for PGAS models
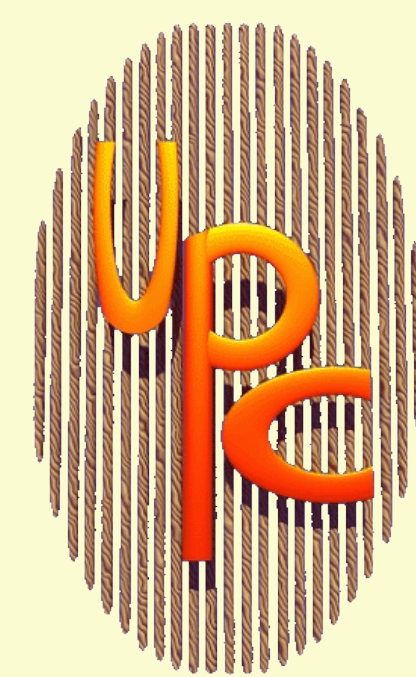 - Version 0.4 supporting Berkeley UPC and Quadrics SHMEM now available at http://ppw.hcs.ufl.edu/

Unoptimized PGAS Application
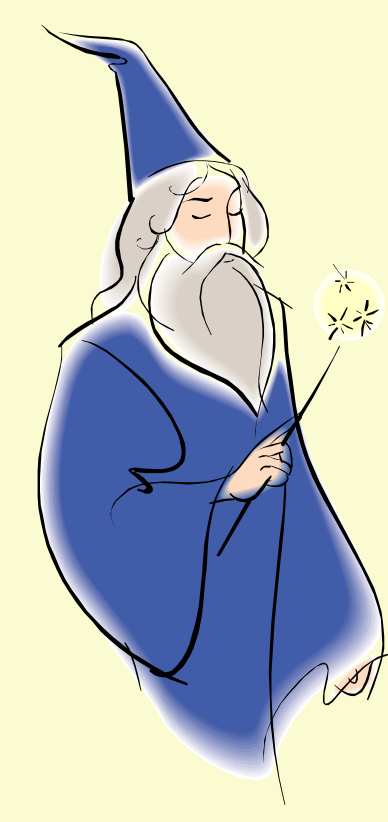
Optimized PGAS Application

Application Optimization

Performance Data Gathering

- Traditional instrumentation techniques (source instrumentation, binary instrumentation, wrapper library, etc.) not sufficient for programs based on the PGAS model due to
  - Aggressive compiler optimizations
  - Wide range of PGAS implementation techniques
  - One-sided memory operations and other aspects of PGAS models
- Global-Address-Space Performance (GASP) interface was developed to facilitate the instrumentation process (http://gasp.hcs.ufl.edu/)
  - Specifies the interaction between program, compiler and the analysis tool
  - Permits tool developers to support PGAS models on all platforms and languages with an implementation of the GASP interface
- GASP support available for Berkeley UPC 2.3.16+
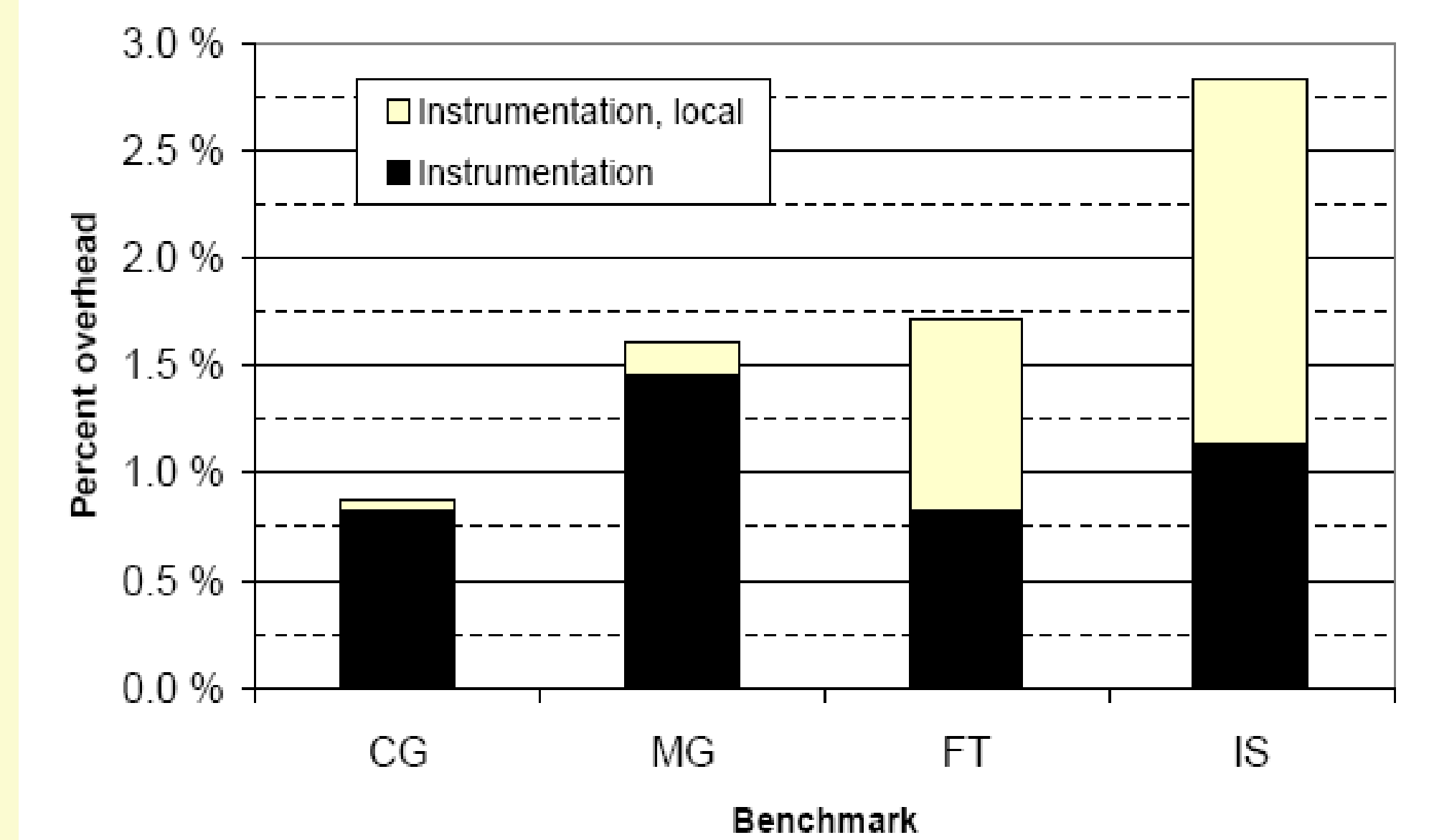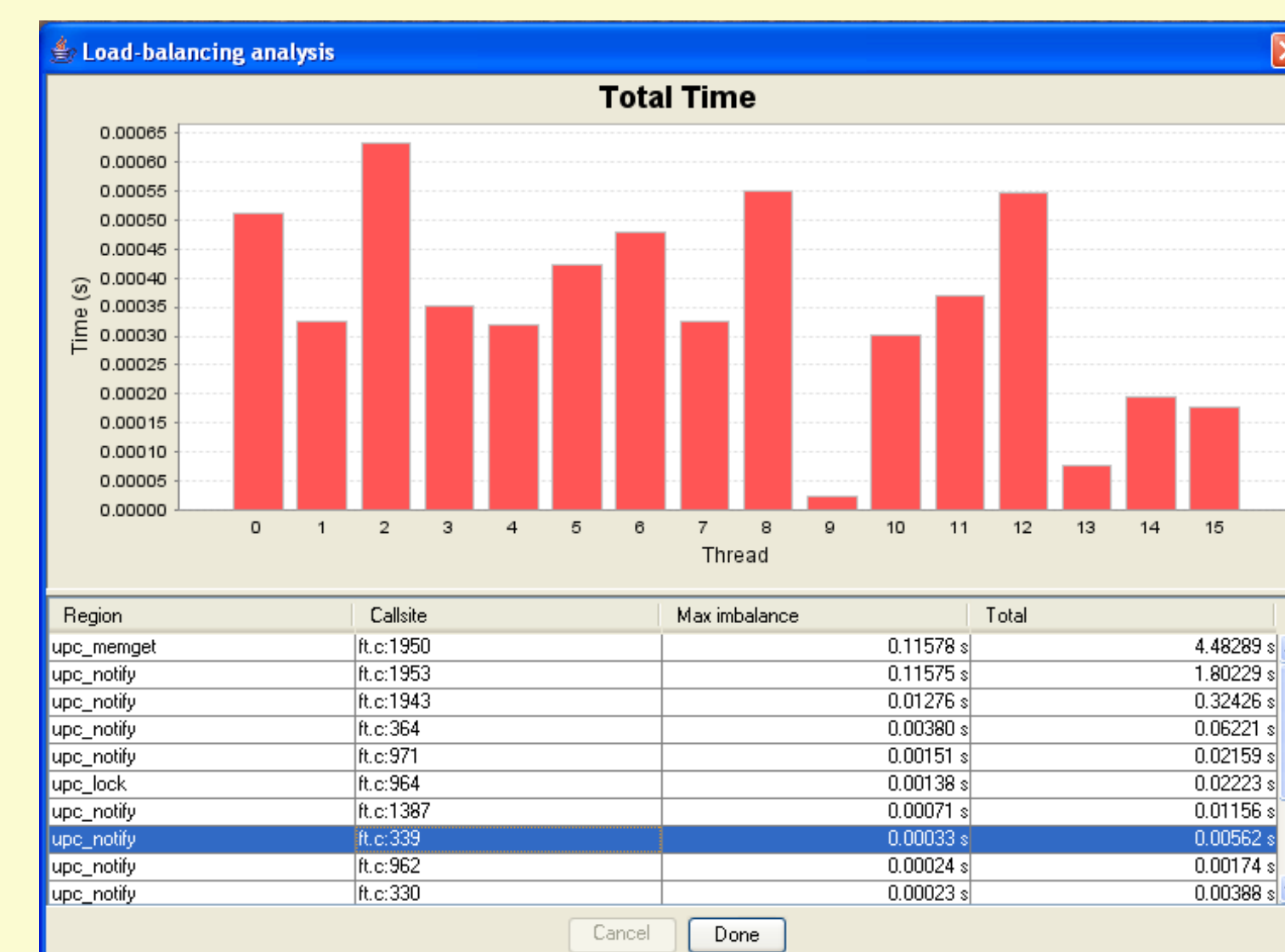 - GASP support for SHMEM, Titanium, and other UPC implementations in development
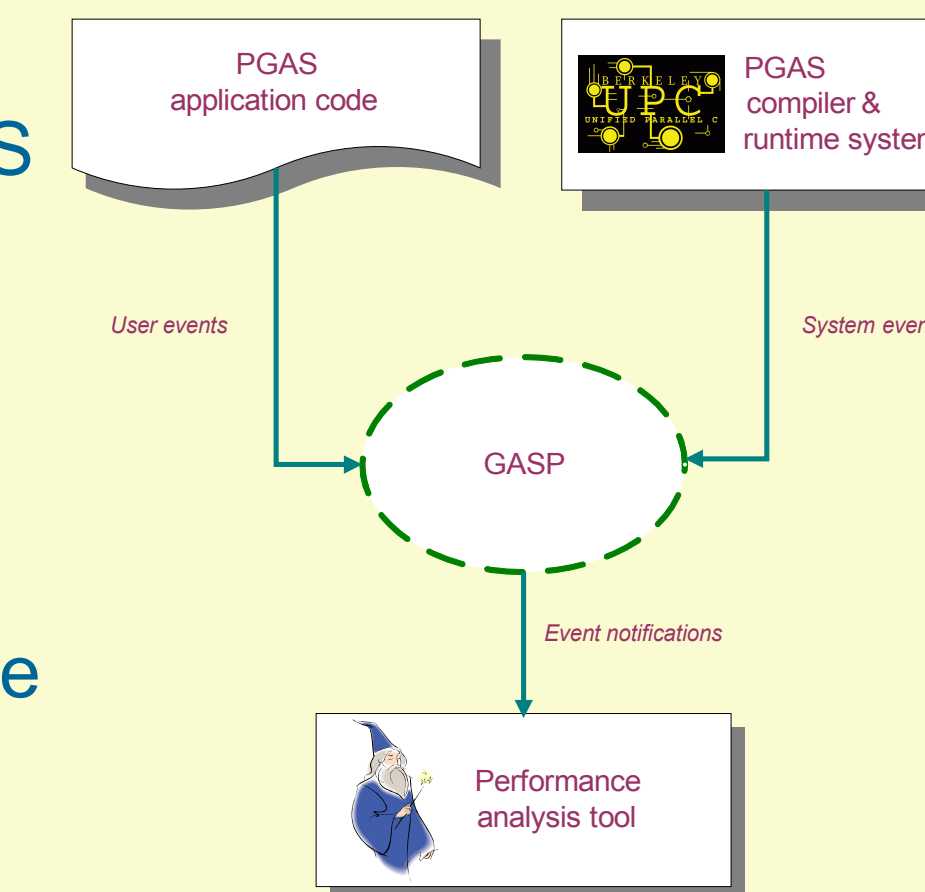
High-level system organization of a GAS application executing in a GASP-enabled implementation
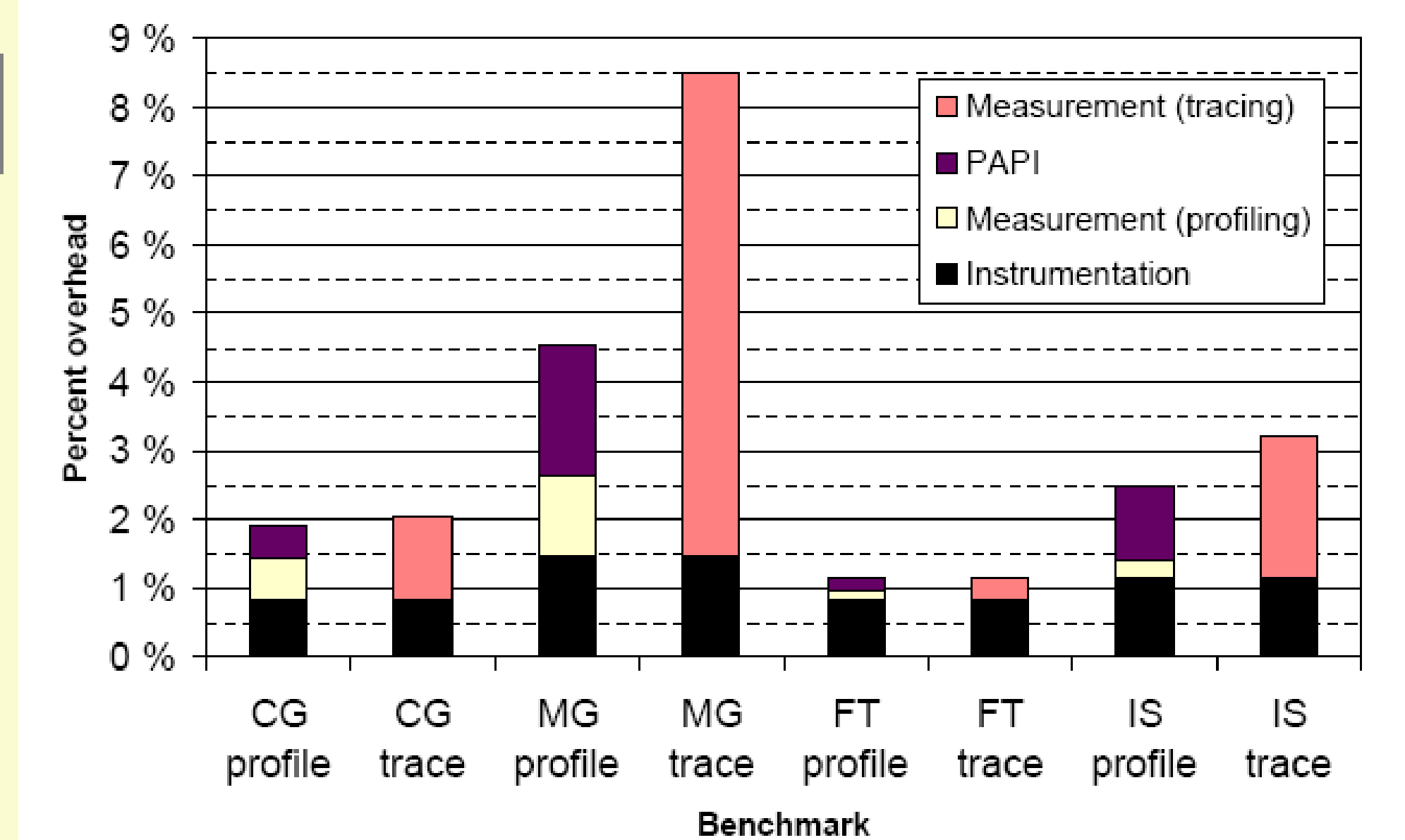
Berkeley UPC GASP overhead for NAS benchmark 2.4 class B on a 32-node, 2-GHz Opteron/Linux cluster with a Quadrics QsNetII interconnect

Incremental raw instrumentation cost for profiling remote and local GAS accesses in the Berkeley UPC GASP implementation

Performance Analysis & Visualization

- Current PPW version supports simple load-balancing analysis
- Advanced semi-automatic bottleneck detection & resolution in development
  - Designed to support parallel programming models in general
  - Also includes scalability analysis and call-path analysis
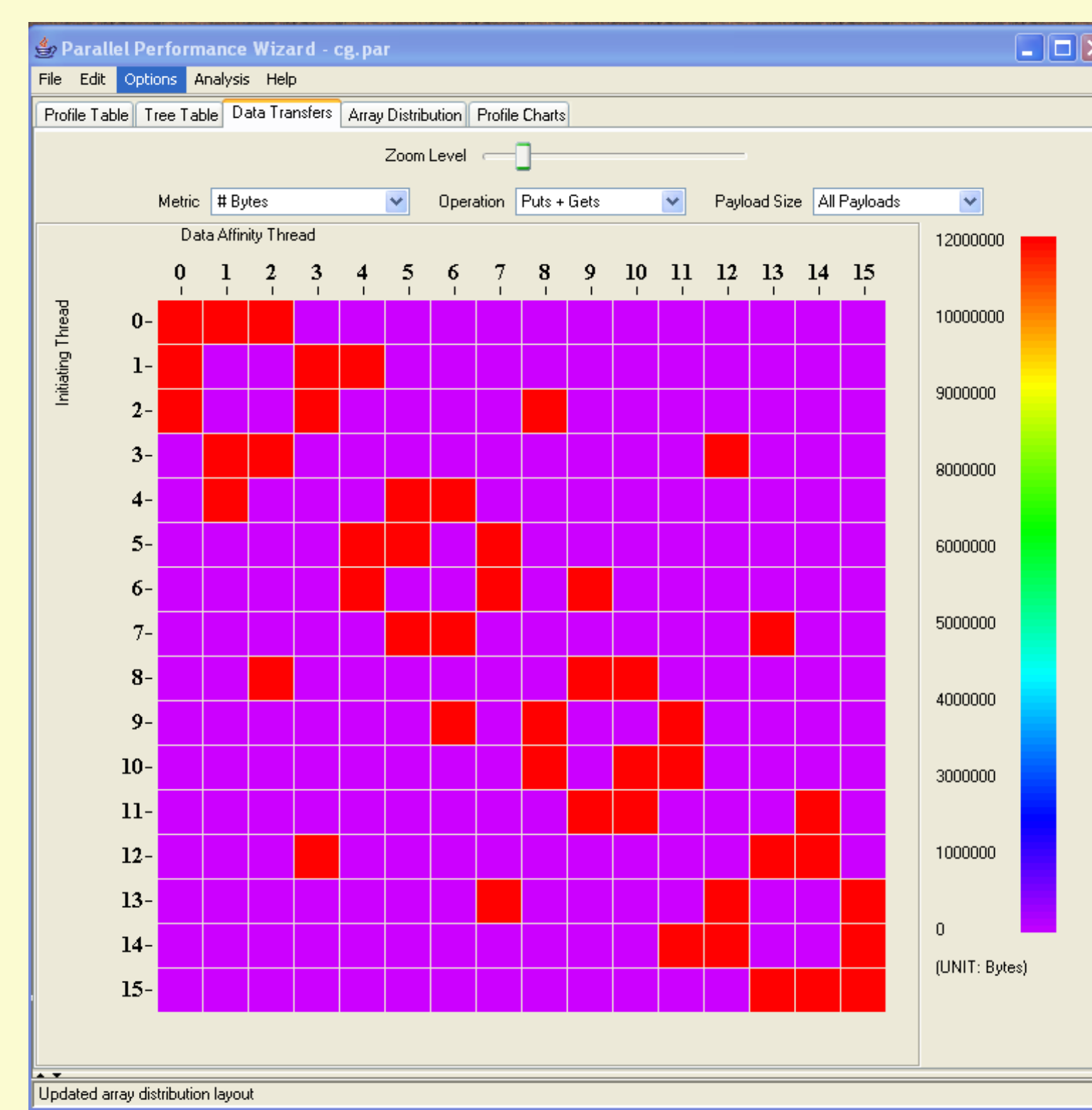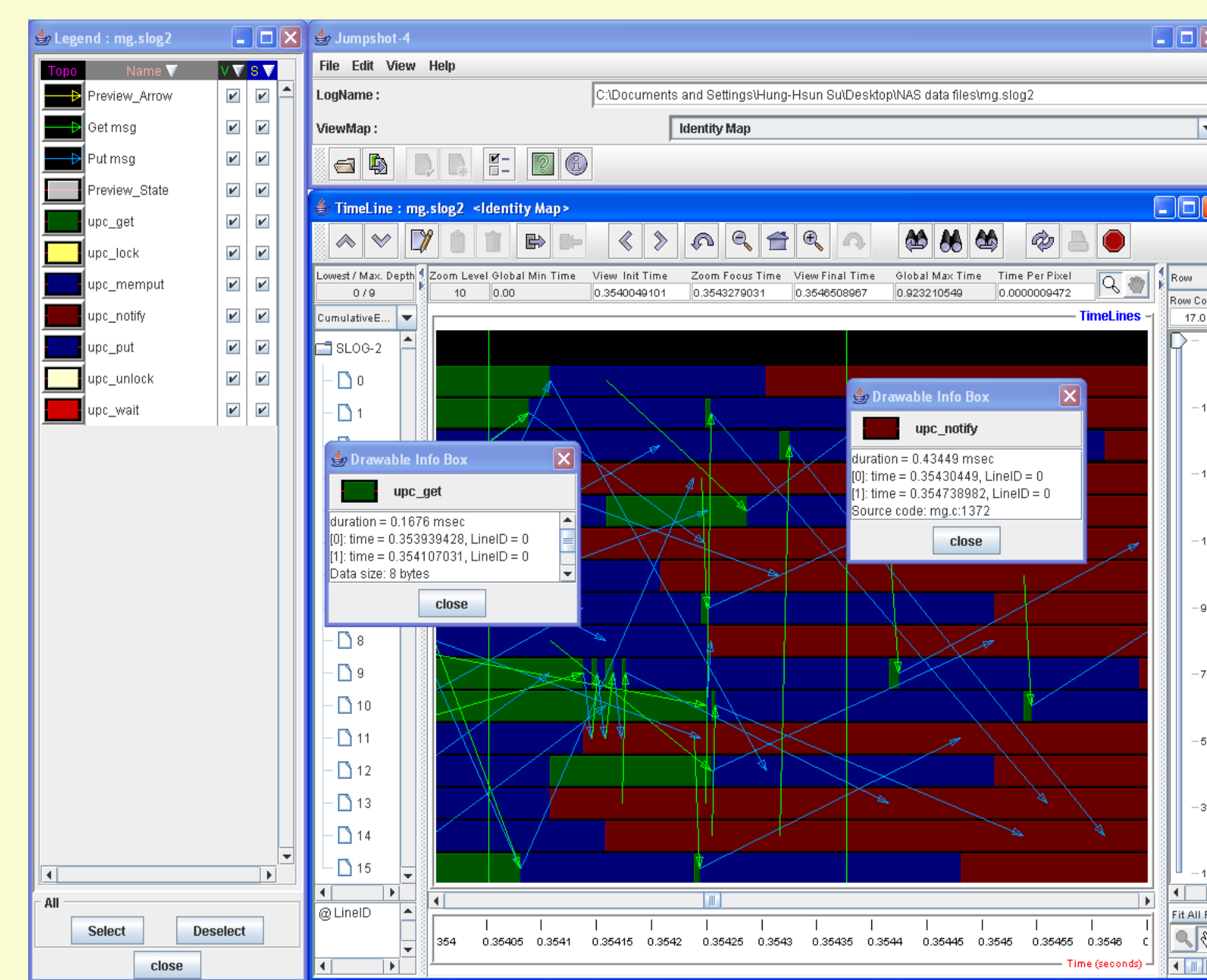 - Generalization of widely deployed pattern-matching technique
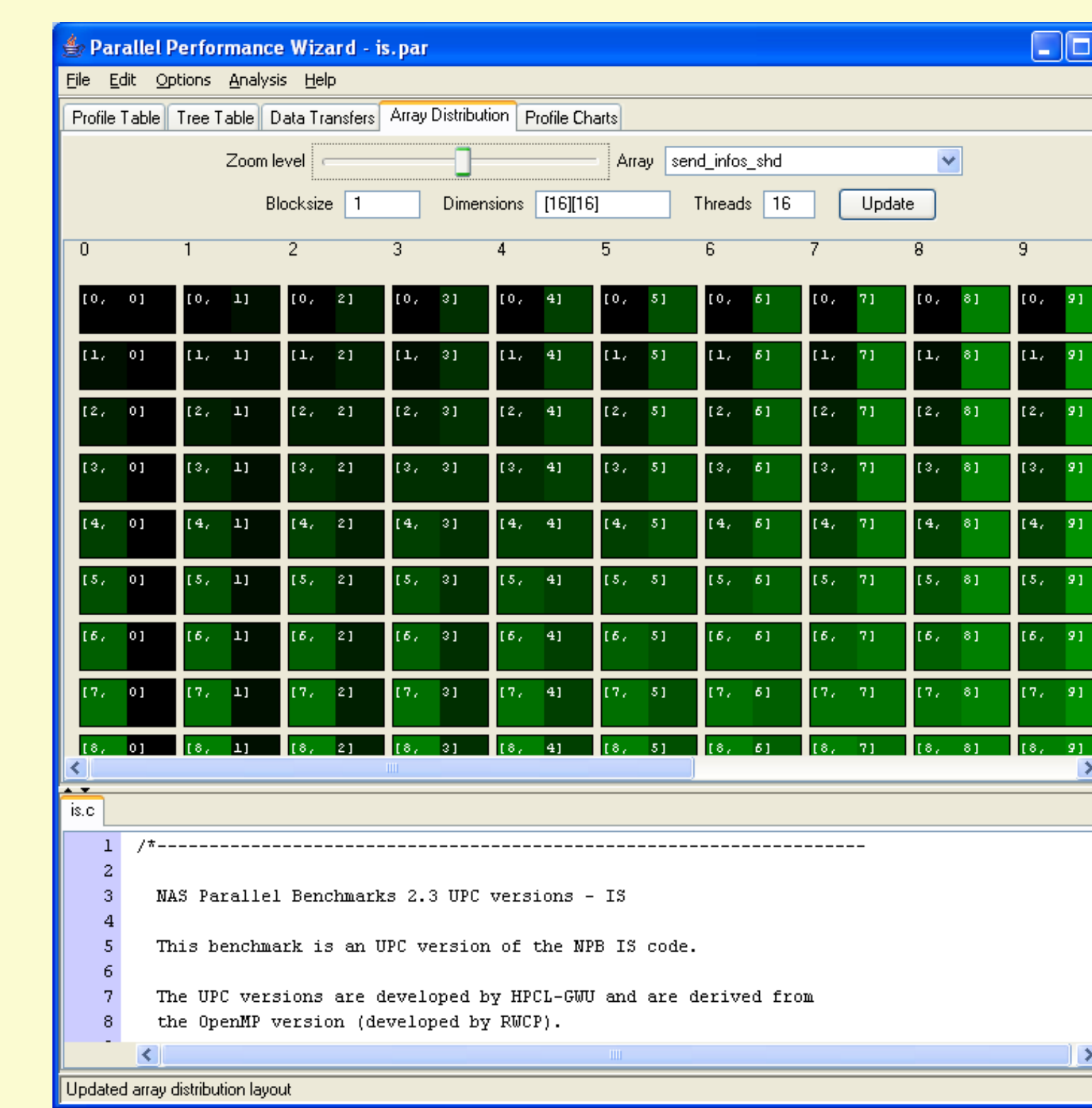
Data Visualizations
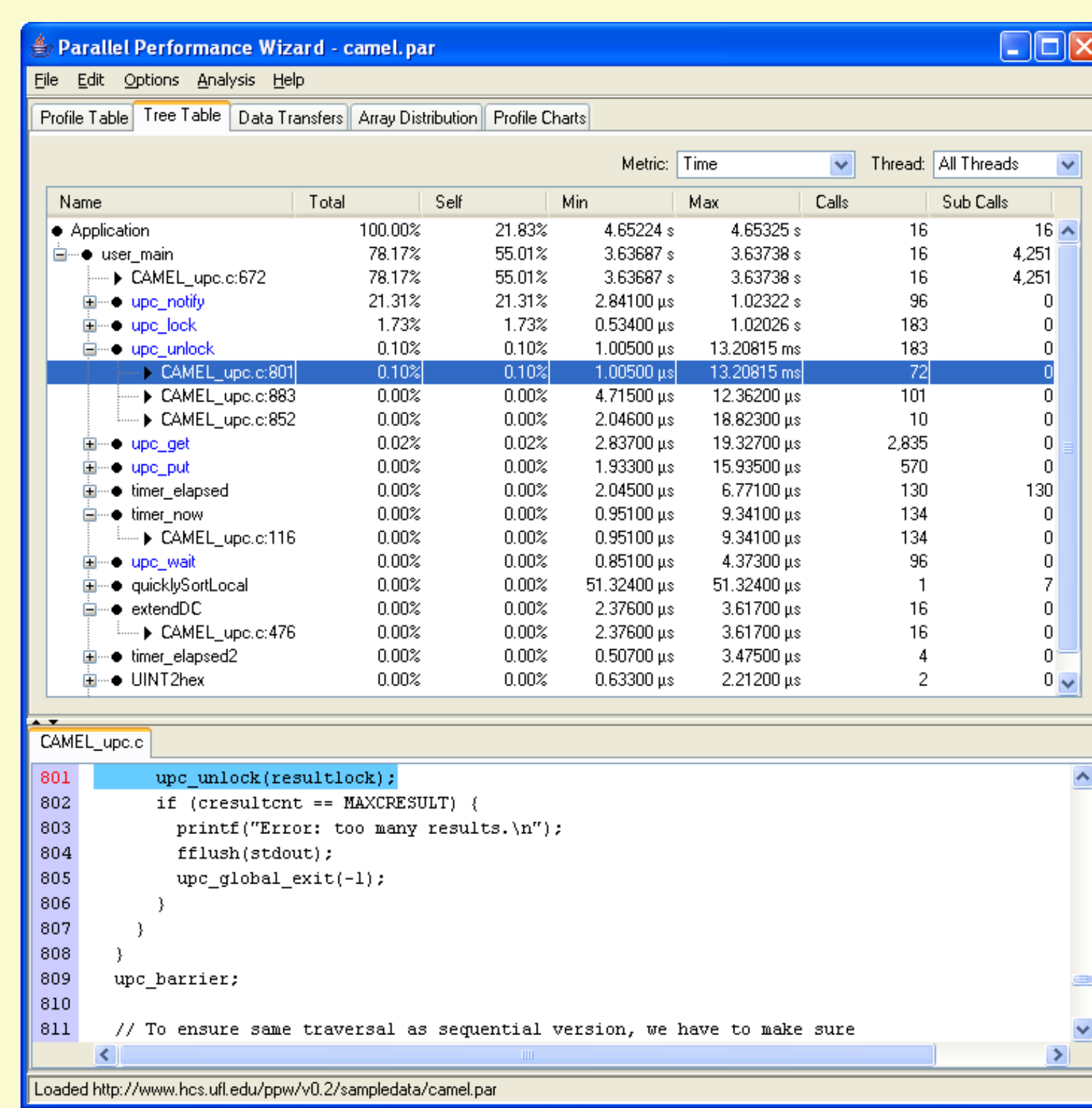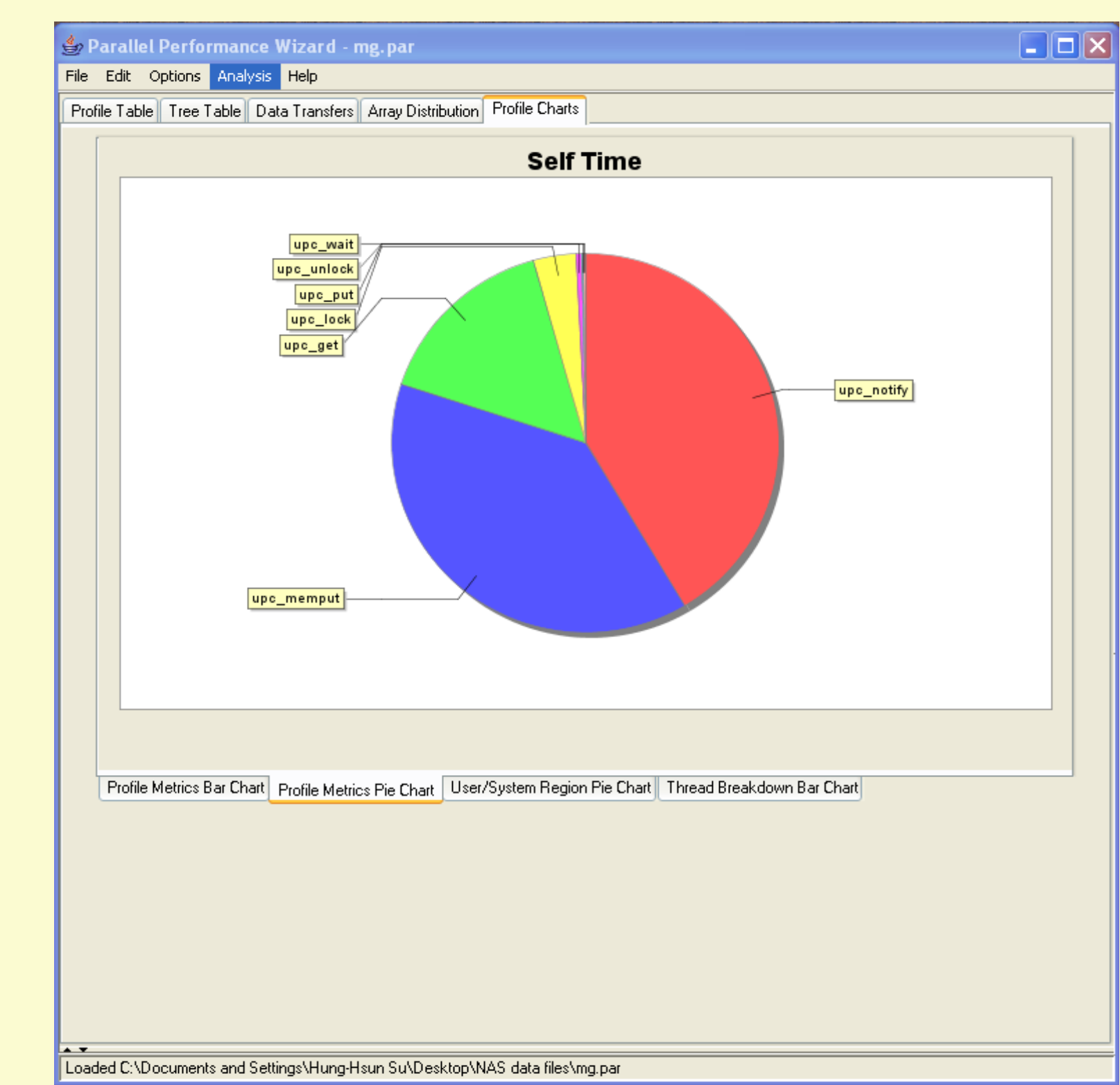

Tree table visualization

Data transfer visualization

Timeline visualization (through export to Jumpshot)

Array distribution visualization

Percentage breakdown visualization