RunView: One-Glance, Zero-Effort
Representation of Program Efficiency within the Cactus Computational Framework

Anna Neshyba  Dr. David Koppelman  Dr. Steven Brandt
Louisiana State University  Center for Computation & Technology  Willamette University

Introduction
Consider this situation: a physicist is developing new simulation models. When their code completes execution, their attention will likely be focused on correctness and new discoveries. As a scientist, not a professional programmer, the physicist may not intuitively know how well-tuned their models are, and major inefficiencies can pass unnoticed.

Project Description
Develop RunView, which will:
- identify inefficiencies within a program
- arrange that inefficiency information in graphics that are:
  a) understandable after a single glance
  b) informative so the domain expert is tempted to fix the obvious inefficiency

In a well-tuned program, at least one hardware resource is saturated (operating at peak performance). Important resources include: floating-point computation (e.g., 1 TFLOP), and data transfer (e.g., 200 GB/s). Saturation is prevented by impeding events and obstacles such as cache misses and synchronization delays.

RunView shows select impeding events in two graphics, Routine and Timeline.

RunView is built for the Cactus Computational framework.

Background
Cactus Computational Framework: Cactus is a programming framework that does calculations on large matrices of data, and designed for use in the sciences [3].

Performance Application Programming Interface (PAPI): PAPI is a widely used API for data collection on various efficiency issues, including cache misses and total stalled time. This API provided RunView with performance data for producing graphics.

Elements of Art: The Elements of Art are defined as color, form, shape, line, space, and texture. With these visual tools, we designed features to represent inefficiency information.

SVG/HTML/JQuery/JavaScript: The graphics produced were SVGs written and saved to HTML documents, that used JavaScript, and JQuery for basic interaction with the graphic.

Characteristics and Features

Existing Tuning Tools:
- Gprof: collects information for specially compiled programs [2]
- HPC Toolkit: produces a graphical display of performance data [1]

However, both of these tools require effort on the part of the user to get started.

Limitations and Future Work
Limitations:
- needs testing by users
- accounts for only select obstacles

Future Work:
- design and incorporate other forms of inefficiencies
- further interactive features

References

Acknowledgments
This project is supported by the National Science Foundation under award ODI-1560412 with additional support from the Center for Computation & Technology at Louisiana State University.

Computer support is provided by HPCT@LSU computing, the Division of Electrical and Computer Engineering at LSU, and by a hardware donation by NVIDIA.

The current programs used for testing RunView are: McLachlan [4], Fluam [5].