

October 22, 2020

# Boosting RSE productivity through advanced tools

## Xpert Network Panel

Robert Sinkovits  
*San Diego Supercomputer Center*  
*XSEDE Co-PI and ECSS Co-Director*  
*[sinkovit@sdsc.edu](mailto:sinkovit@sdsc.edu)*

# XSEDE

Extreme Science and Engineering  
Discovery Environment



Supported by OAC 15-48562.

# Introduction

I am the co-PI of the NSF-funded XSEDE project and co-director of XSEDE's Extended Collaborative Support Service (ECSS). Our goal is to

*Improve the productivity of the XSEDE user community through successful, meaningful collaborations to optimize their applications, improve their work and data flows, and increase their effective use of the XSEDE digital infrastructure and broadly expands the XSEDE user base by engaging members of underrepresented communities and domain areas*

Our team consists of approximately 70 computational scientists spread across 10 institutions and charging 28 FTE of effort. We support all domains of science and a diverse set of computing paradigms.

## `printf` and `time` are the tools of choice

Our users do not generally use advanced tools. The most popular debugging tool is still C/C++ `printf` and the most popular performance analysis tool is the Linux `time` utility.

Attention spans are short and demands on time are high. If a tool is not intuitive and super easy to use, it's unlikely to get adoption.

`gprof` can hardly be called an “advanced tool”, but the barrier to usage is very low (compile with `-pg`) and the report is easy to interpret.

Intel compiler optimization reports (`-qopt-report`

# Adoption of `gprof` and Intel reports is a big success

`gprof` can hardly be called an “advanced tool”, but

- barrier to entry is very low (compile with `-pg`)
- report is easy to interpret
- easy to introduce and teach to non-expert

Intel compiler optimization reports are also

- easy to implement (compile with `-qopt-report=[n]`)
- output requires more interpretation, but still very useful even if the users don't fully understand vectorization etc.

# Need more advanced tool usage

printf, time, gprof and Intel reports are a great start, but we need to go much further

- Limited functionality
- Don't tell us about parallel execution, bottlenecks, I/O
- For the truly advanced tools, maybe we should just focus on the RSEs and evangelize to the users who really need them