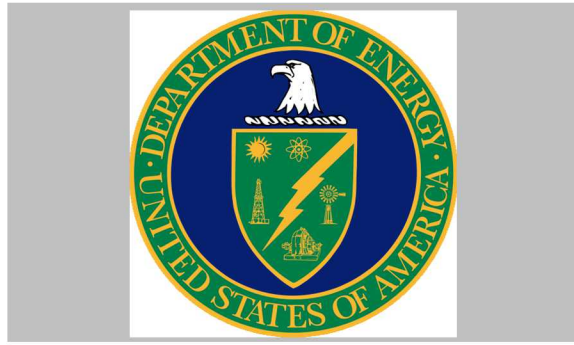
SAND2018-7304PE



LDPXI Application Performance Survey Tool

Douglas M. Pase, PhD

2018-07-06

U.S. DEPARTMENT OF ENERGY | National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND NO. 2018-**XXXX** PE

The Need For A Reliable Survey Tool (Benefits)

Sandia National Laboratories

- Rapid triage of application performance
- Application strong and weak performance scaling studies
- Application performance regression testing – what changed with the new version?
- Cluster resource use studies – what resources are being used by applications?
- Architecture performance studies – does this architecture provide what applications need?

What Does A Survey Tool Do?

Sandia National Laboratories

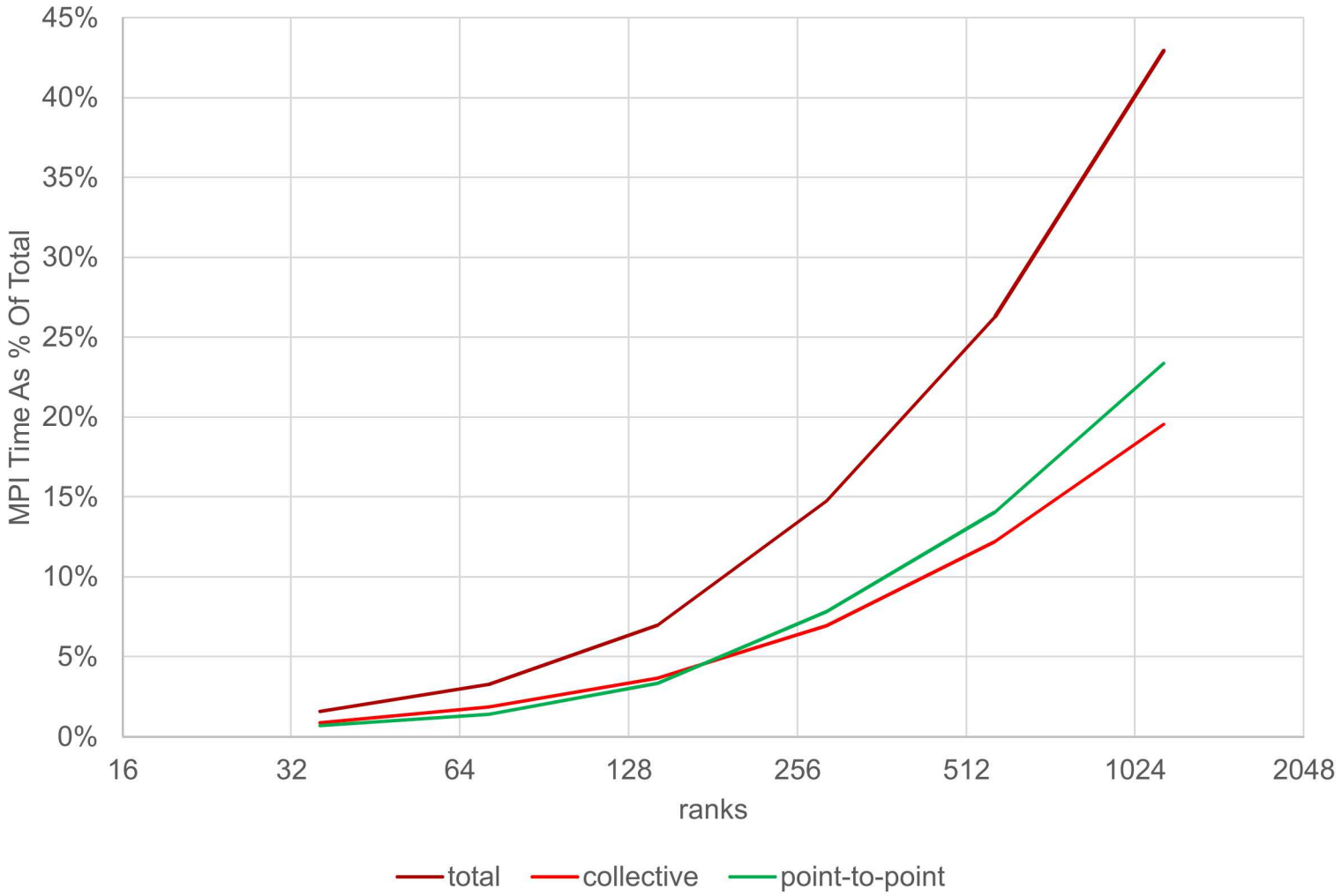
- Give a first glimpse into how an application performs
  - Is it dominated by CPU, communication, or I/O performance?
  - Does it use cache effectively? Vectorize well?
  - Does it have a high proportion of floating-point operations?
  - What is the communication profile? Is there a load imbalance?
- Production applications
  - Very large code base (millions of lines of code)
  - Complex library structure (10's of packages, multiple languages)
  - Optimized code, often without debug symbol tables
- Production scales
  - Long-running codes
  - Many MPI ranks and cores
  - Large data sets

An Example of a Production Application

Sandia National Laboratories

- The following case was part of a recent study of tools at Sandia
- This case has a medium-sized code base (around 250,000 of lines of code)
- Complex library structure (multiple packages, multiple languages, several not available in source)
- Production binary, optimized compiles, no debug symbols or tables
- The data is from a small production test case, part of a strong scaling study that ran approximately 388 processor hours on 36 to 1,152 MPI ranks
- This case is selected for what it was able to quickly show about the application.
- Other cases we have used (not shown) have used much larger applications (over 2.5 million LOC) and run for over 24 hours on to 256 nodes.
- (LDPXI overhead does not increase with the number of nodes or the length of the run.)

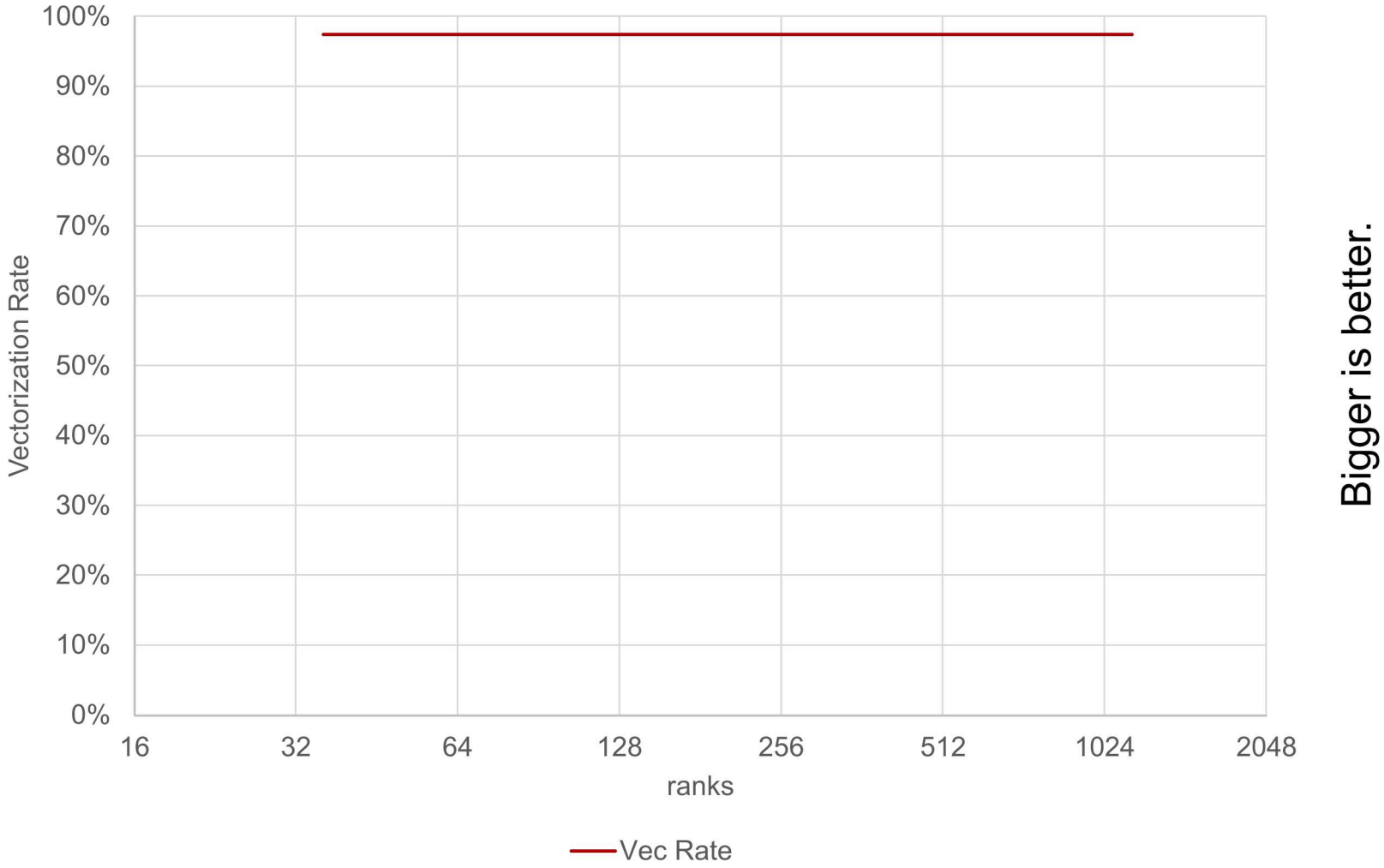
MPI Time Relative To Total Time



Smaller is better.

Is it dominated by CPU, communication, or I/O performance? (CPU.)
What is the MPI profile? (Nearly equal collective as point-to-point.)

Vectorization Rate

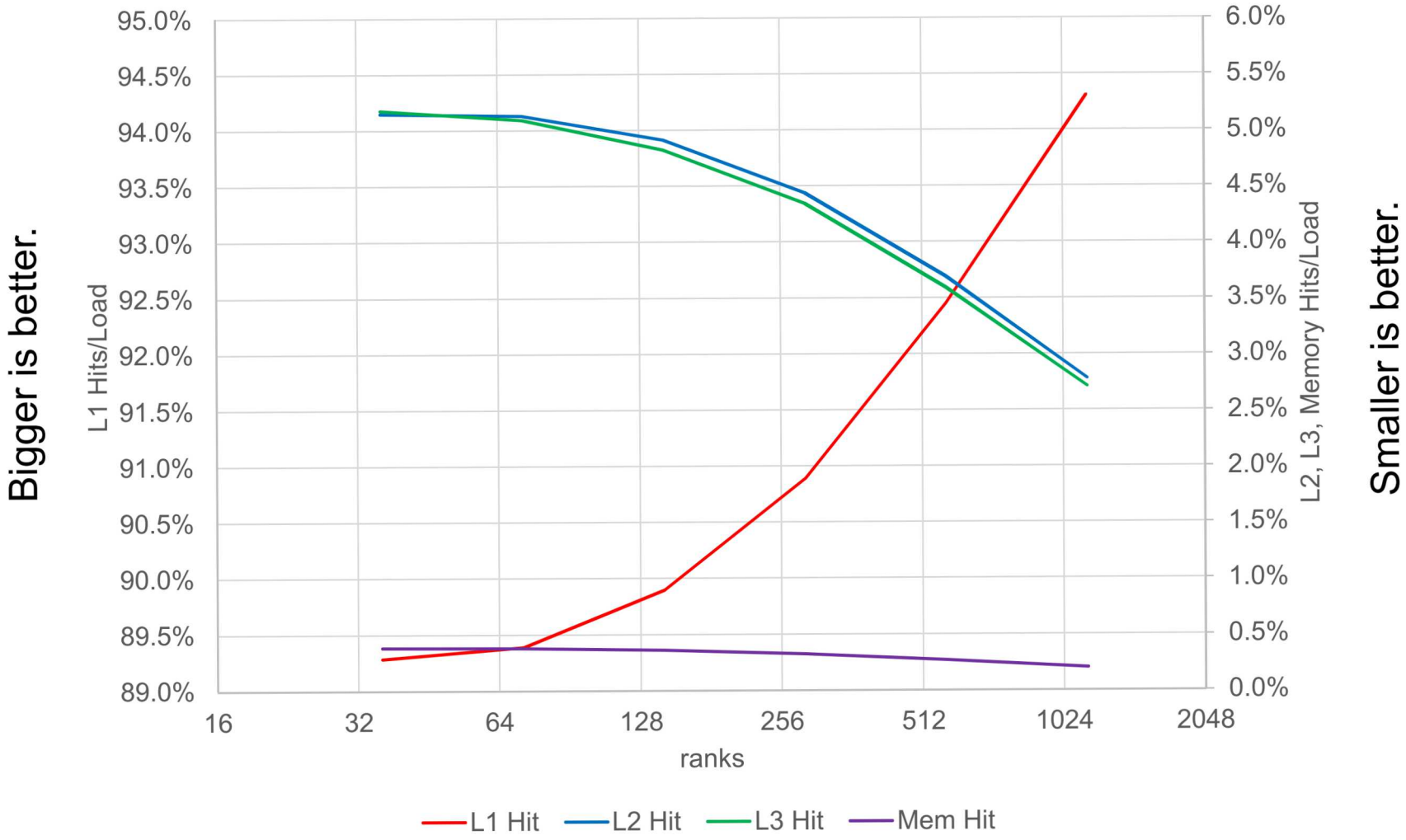


Bigger is better.

Does it vectorize well? (Yes.)

Cache Hit Rates Per Load Instruction

Sandia National Laboratories

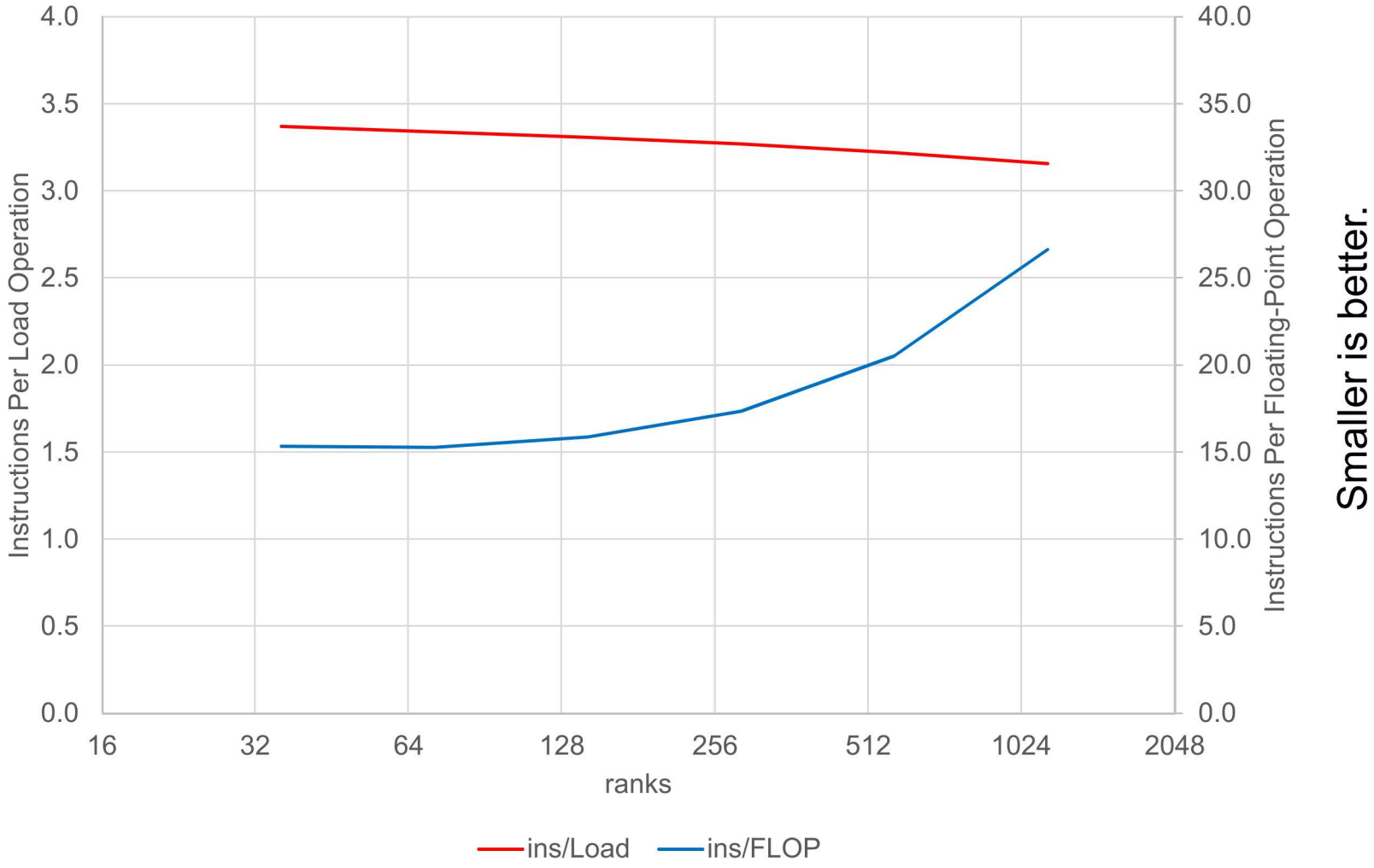


Does it use cache effectively? (Yes.)

Instructions Per FLOP, Instructions Per Load

Sandia National Laboratories

Bigger is better.



Smaller is better.

Does it have a high proportion of floating-point operations? (No!)

Conclusions

- LDPXI is a simple, light-weight tool that provides a highly useful set of features.
- LDPXI works well with large scale, complex, production applications where other tools sometimes have difficulty.
- LDPXI provides a high-level summary of application performance useful for rapid triage (shown here) as well as scaling, regression, resource usage and architecture studies.

Questions?

Sandia National Laboratories

What LDPXI Does

Sandia National Laboratories

- LDPXI (Load Pixie) – **LD_P**reload e**X**ecutable **I**nstrumentation
- Inserts high-level performance measures into an application
  - Initialization code for hardware performance counters is executed before main()
  - Instrumentation is placed as wrappers around C and MPI library routines of interest
  - Final code for reading counters and writing data to files is executed after exit()
- Measures each of the following for the whole application
  - Resource usage (total memory, user and system time)
  - MPI usage (number of messages, message size, time in communication)
  - Libc I/O and memory calls (number of calls, read/write sizes, time in I/O)
  - Hardware performance counters and CPU events
- Incurs very low overhead (much less than 1% of execution time, very low memory use)
- Works for large, complex, multi-language, MPI production applications at large scale
- Instrumentation takes place at link time (as a shared library) or load time (using LD_PRELOAD)
  - No changes to the application source, object or binary are required

What LDPXI Does Not Do

- LDPXI does not measure anything smaller than the whole program
- LDPXI does not use statistical sampling (like Allinea MAP and GNU gprof do)
- LDPXI is not a program debugger – it will not find logic errors
- LDPXI is not a memory debugger – it will not find memory leaks or bad pointers

What's Available

- LDPXI Version 1.0 (Available Now!)
  - Gathers libc, MPI and hardware event counters for all MPI ranks
  - Reduces the data to max, min and sum across all ranks
  - Piggy-backs the reductions on the MPI messaging system
  - Writes the reduced data as a CSV file
- LDPXI Version 1.1 (Coming Soon!)
  - Gathers the same libc, MPI and hardware event counter data per MPI rank as version 1.0
  - Raw data is written to individual binary files, one for each MPI rank
  - Data is extracted post-run using a flexible data calculator
  - Supports a wider range of analyses, including regression, scaling and architecture studies
  - Analysis can be written to a file (CSV, JSON, etc.) or directly to gnuplot