From job submission support to advanced performance tuning of parallel applications.

A case study from a university with an open access policy to high performance computing.

Robert Henschel
Director, Science Community Tools
Research Technologies, UITS
Indiana University
June 22nd, 2017

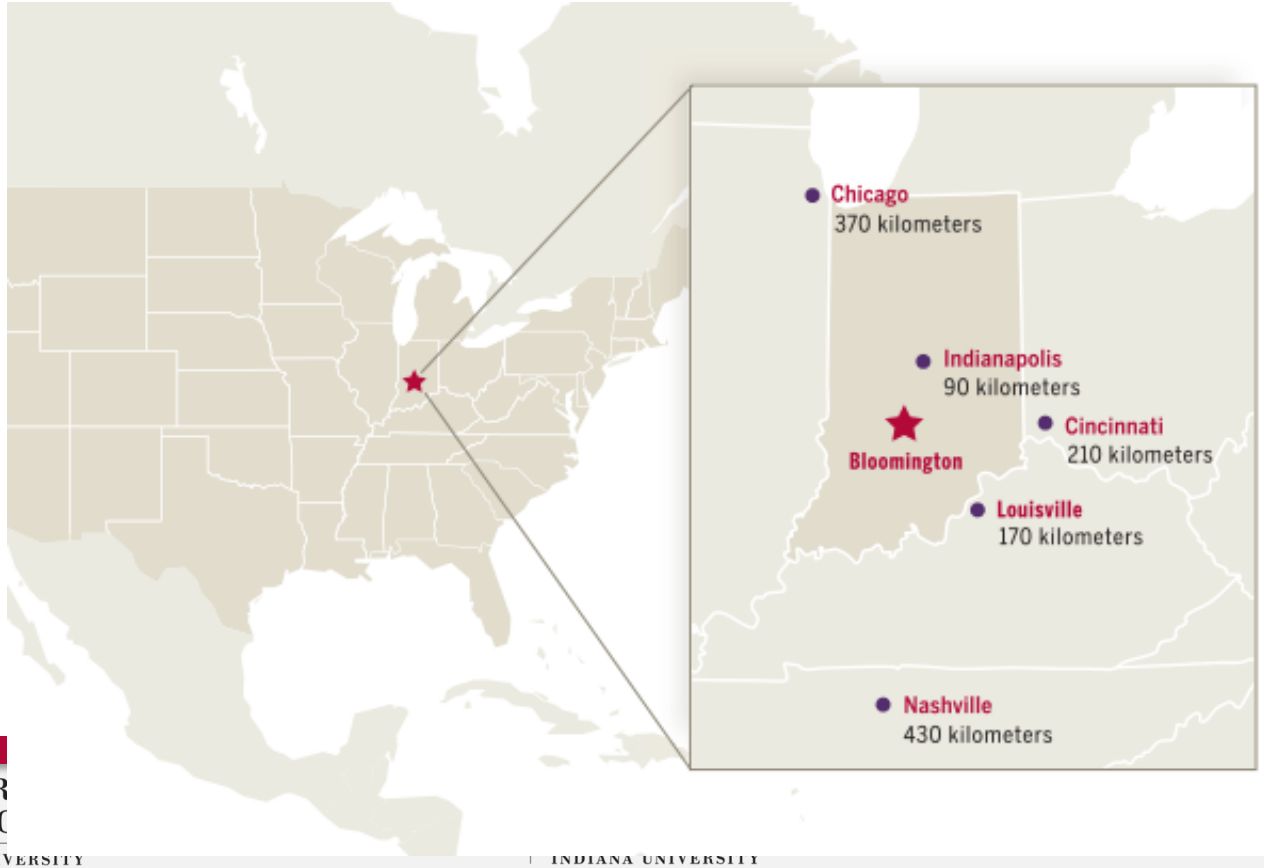
RESEARCH TECHNOLOGIES
INDIANA UNIVERSITY
University Information Technology Services

PERVASIVE TECHNOLOGY INSTITUTE
INDIANA UNIVERSITY

Contents

- Indiana University and HPC@IU
- What is Performance Tuning
- Examples of Performance Engineering
- SPEC High Performance Group

Indiana University



IU – Campuses and Medical School Centers

IU Campuses

IU School of Medicine campuses and clinics

IU Overview

Fall 2016	Number
Undergraduate	93,740
Graduate	12,397
Doctoral - Research	4,323
Doctoral - Practice	3,700
Total Students	114,160
Staff	11,498
Faculty	9,005
Grand Total	134,633

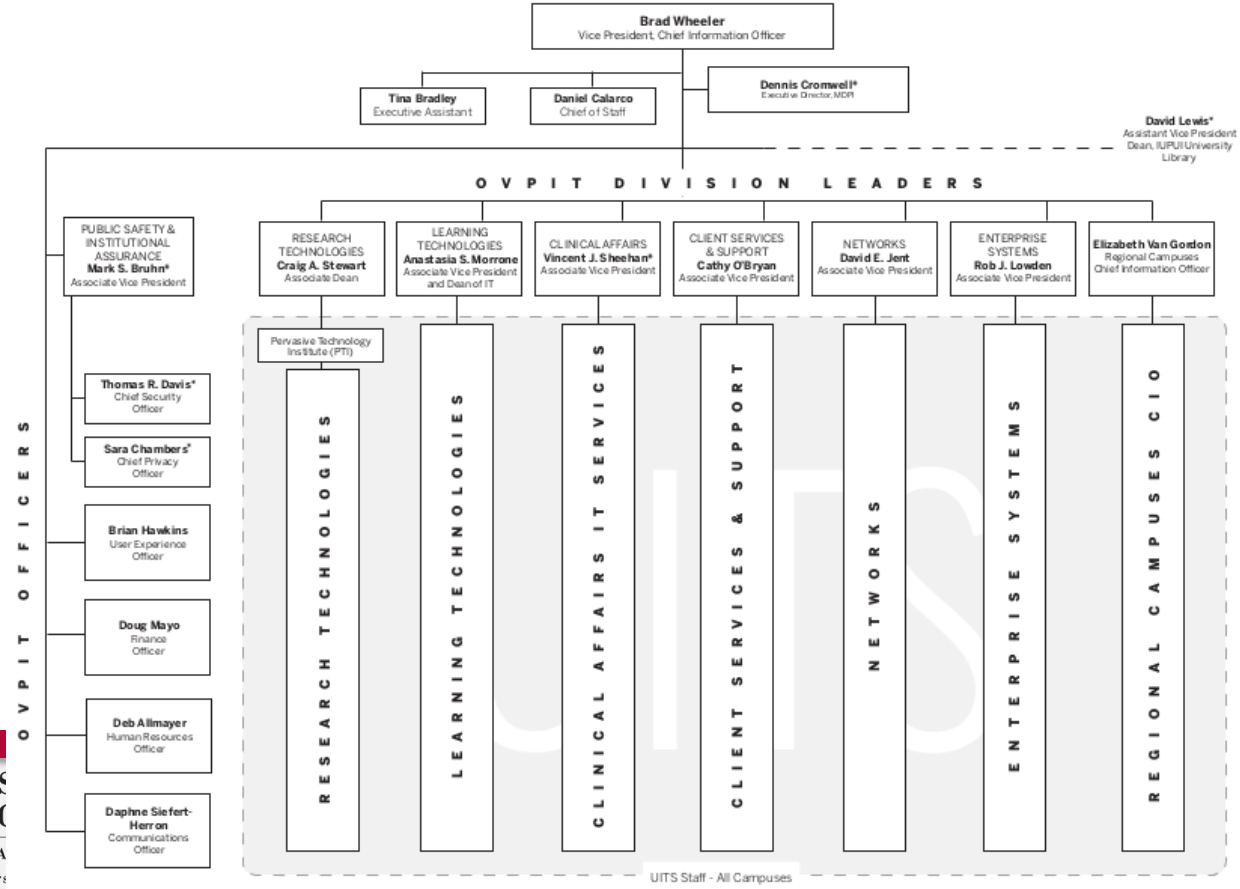
Overall

- Operating budget - $3.5B
- Grant Awards of $614M in 2016

Centralized IT Org - UITS

- 700+ professional staff
    - 130 Research Technologies
- 500+ part time staff

High Performance Computing at IU



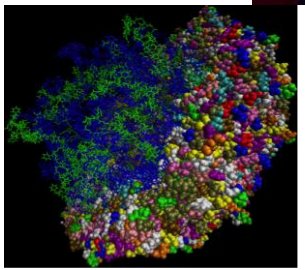
Research Technologies

Associate Dean, RT, and Executive Director, PTI Craig A. Stewart

Systems Matt Link	Visualization and Analytics Eric Wernert	Science Community Tools Robert Henschel	Community Engagement and Interoperability Therese Miller	Advanced Cyberinfrastructure Dave Hancock
High Performance File Systems Stephen Simms	Research Analytics Scott Michael	Scientific Applications and Performance Tuning Abhinav Thota	Campus Bridging and Research Infrastructure Joe Buttler	High Performance Systems Peg Lindenlaub
Research Storage Charles McClary	Advanced Visualization Lab Michael Boyles	National Center for Genome Analysis Support Tom Doak	Grant Support and Outreach Winona Snapp-Childs	Jetstream Cyberinfrastructure Georg Turner
Application Desktop Virtualization Stephanie Cox	Research Data Services Esen Tuna	Advanced Parallel Applications Ray Sheppard	Education, Outreach, Training Robert Ping	
High Throughput Computing Robert Quick	Digital Humanities Cyberinfrastructure Tassie Gniady	Scalable Compute Archive Arvind Gopu	Jetstream Project Managemnt and Outreach Jeremy Fischer	
		Advanced Biomedical IT Core Richard Meraz		

HPC @ IU - Compute

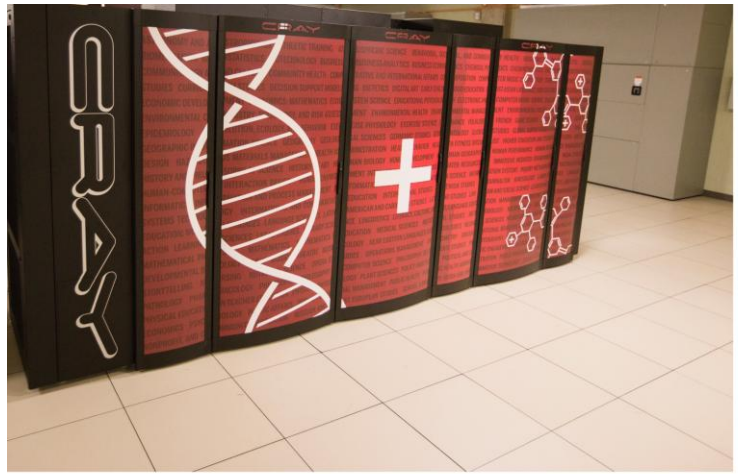
- **Big Red II** – Cray XE6/XK7
  - 1020 nodes, 1 PFLOPS
  - CPUs/GPUs
  - CLE 5 up 02
  - Torque/Moab
  - 22 LNET Routers (QDR)
  - 4 DVS nodes (10Gb)

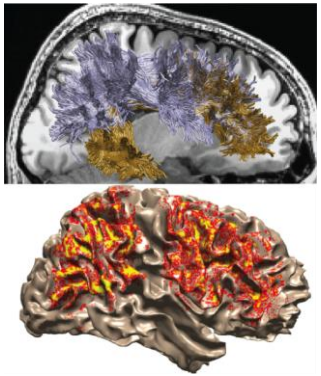


- Available to all Faculty, Staff, and Graduate Students
- Support/consulting available

HPC @ IU - Compute

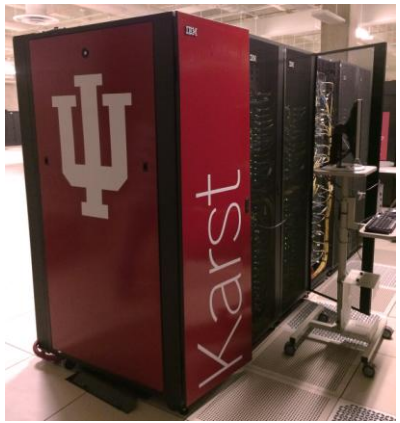
- **Big Red II+** – Cray XC30
  - 560 nodes, 286 TFLOPS
  - Only CPUs
  - CLE 5, soon CLE 6
  - SLURM
  - 6 LNET Routers (2x FDR)
  - 2 DVS Nodes (40Gb)
  - Available to Grand Challenge Projects
  - Jobs >= 256 node desired



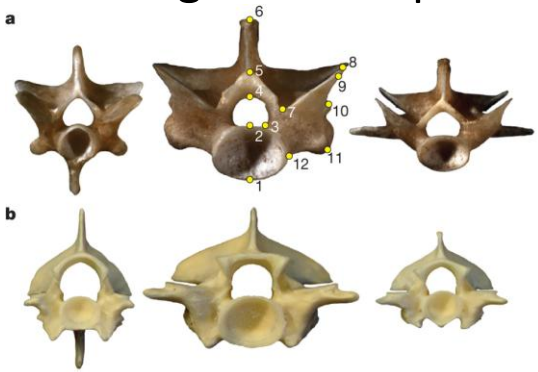


HPC @ IU - Compute

- **Karst** – standard cluster available for expansion
  - General purpose Intel Linux cluster
  - Condo nodes may be purchased for special needs or greater response



- Started at ~275 nodes -> ~400
- Upgrade in progress
- First nodes installed in Fall 2014
- NextScale nx360 M4 & M5
- 10/40Gb networking
- Memory profiles from 32GB -> 1024GB
- Using xCAT
- RHEL6 soon with some RHEL7



HPC @ IU – Interactive Compute

- 13 “fat” nodes with ThinLinc remote desktop
- Serving users with interactive needs and users new to HPC
- Test bed for HPC convenience features

HPC @ IU – Cloud Compute

- **Jetstream** – NSF production cloud
- NSF's first cloud dedicated to science and engineering research across all areas of activity supported by the NSF
- Interactive/On-Demand System
- User-selectable library of VMs
- Supporting 9 science gateways currently
  - Galaxy, CyVerse, SEAGrid, others
- >1,500 users in 1st year
- 20% new to XSEDE

Award #1445604

HPC @ IU – Storage

Data Capacitor, DC-WAN, DC-RAM

- Data storage on disk, not backed up (scratch & projects)
- Temporary storage of research data – purged regularly
- 5.3 PB DCII / 1.1 PB DC-WAN / 35 TB DC-RAM

- Wrangler (dual-site 20 PB environment with TACC)

- Lustre-based file systems
- In the midst of storage procurement
- Will add 1-2 file systems and ~2x capacity

HPC @ IU – Storage

Scholarly Data Archive (SDA)

- Distributed tape storage for large-scale archival/near-line storage
- Mirrored– 2 copies (IUB and IUPUI)
- Open to IU community – undergrads/non-IU must have sponsor
- Supports collaborative activities

- 43 PB of tape storage capacity
- Supports SFTP, HSI, HPSS API
- HIPAA-aligned

Contents

- Indiana University and HPC@IU
- What is Performance Tuning
- Examples of Performance Engineering
- SPEC High Performance Group

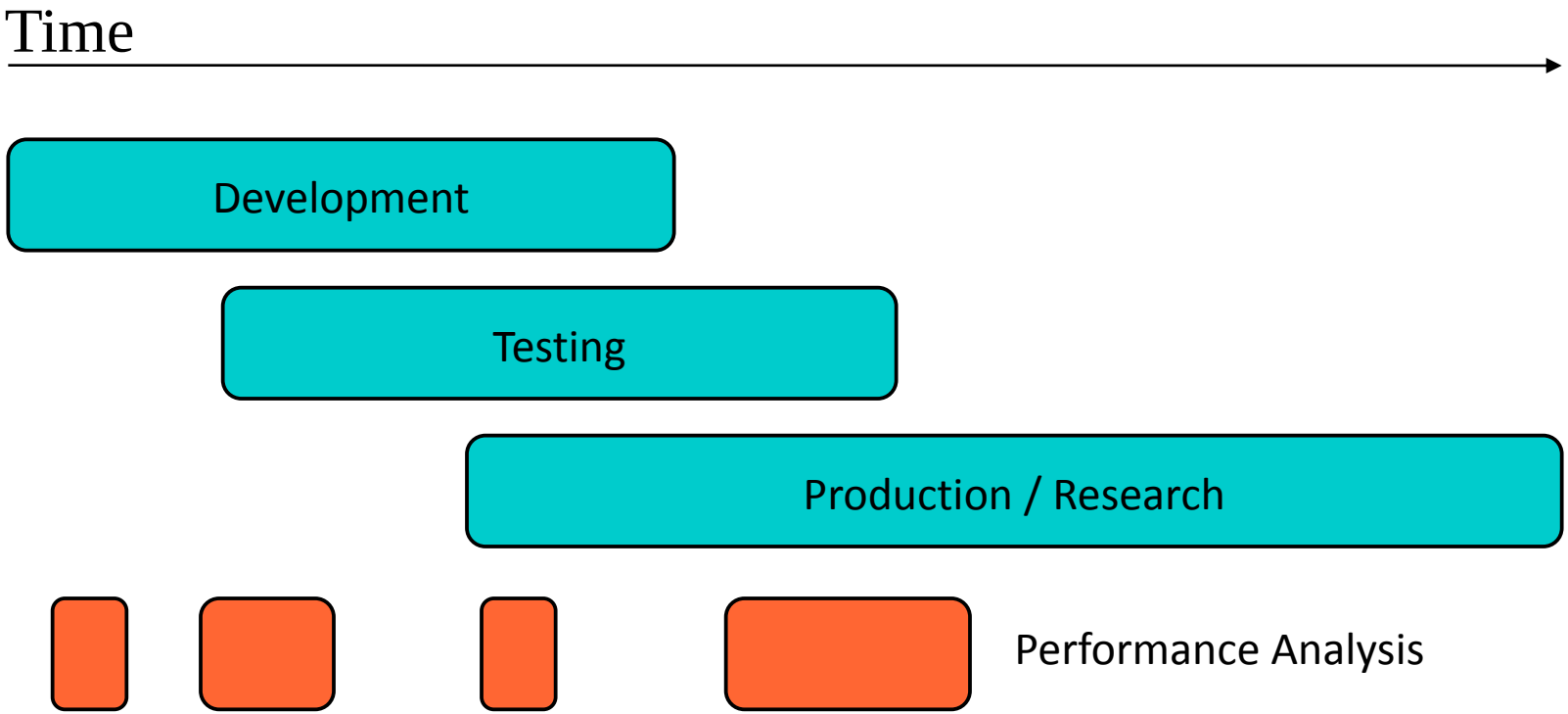
What is Performance Tuning?

- Decrease the resource need or increase the output of the application/workflow.
- … of scientific applications.
  - Make them run faster.
  - Make them run at all.
  - Run problem sizes impossible without tuning.

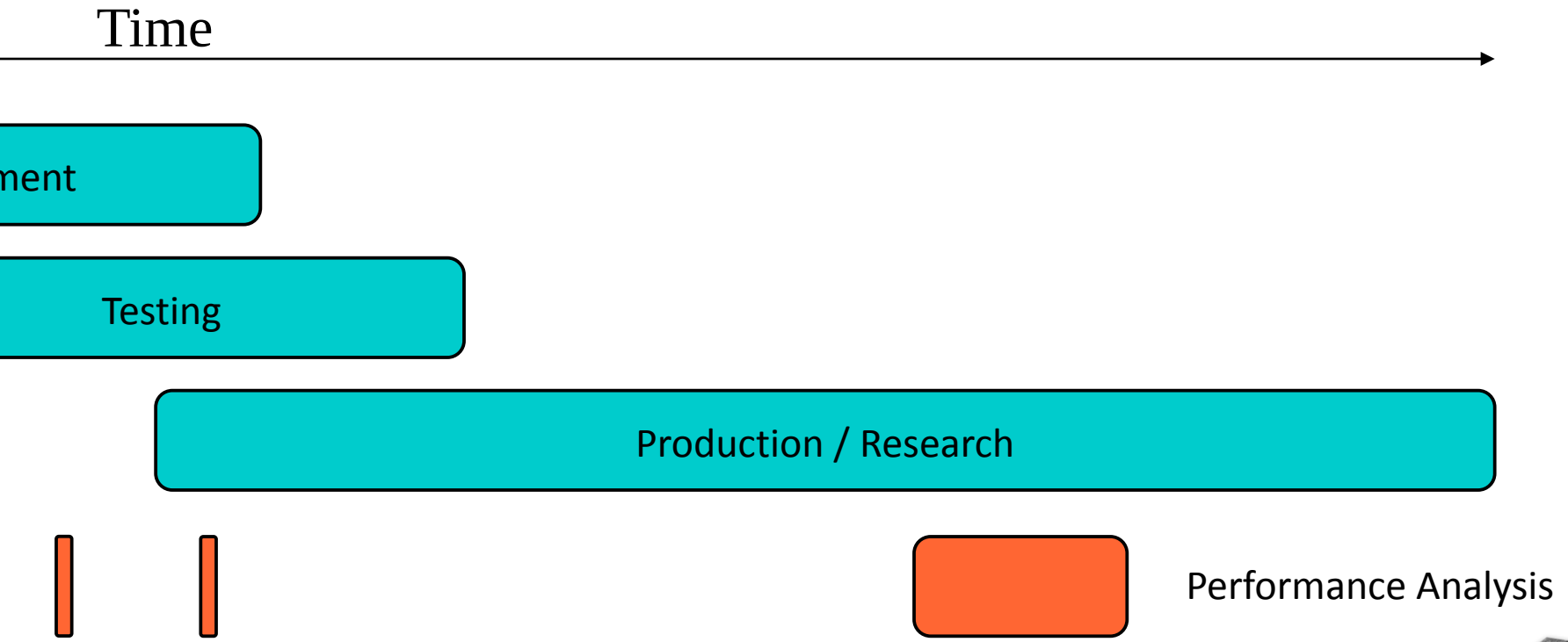
What is Performance Tuning?

- … of scientific workflows.
  - Make the whole computational workflow run faster.
  - Work with a research group to enable research otherwise impossible.

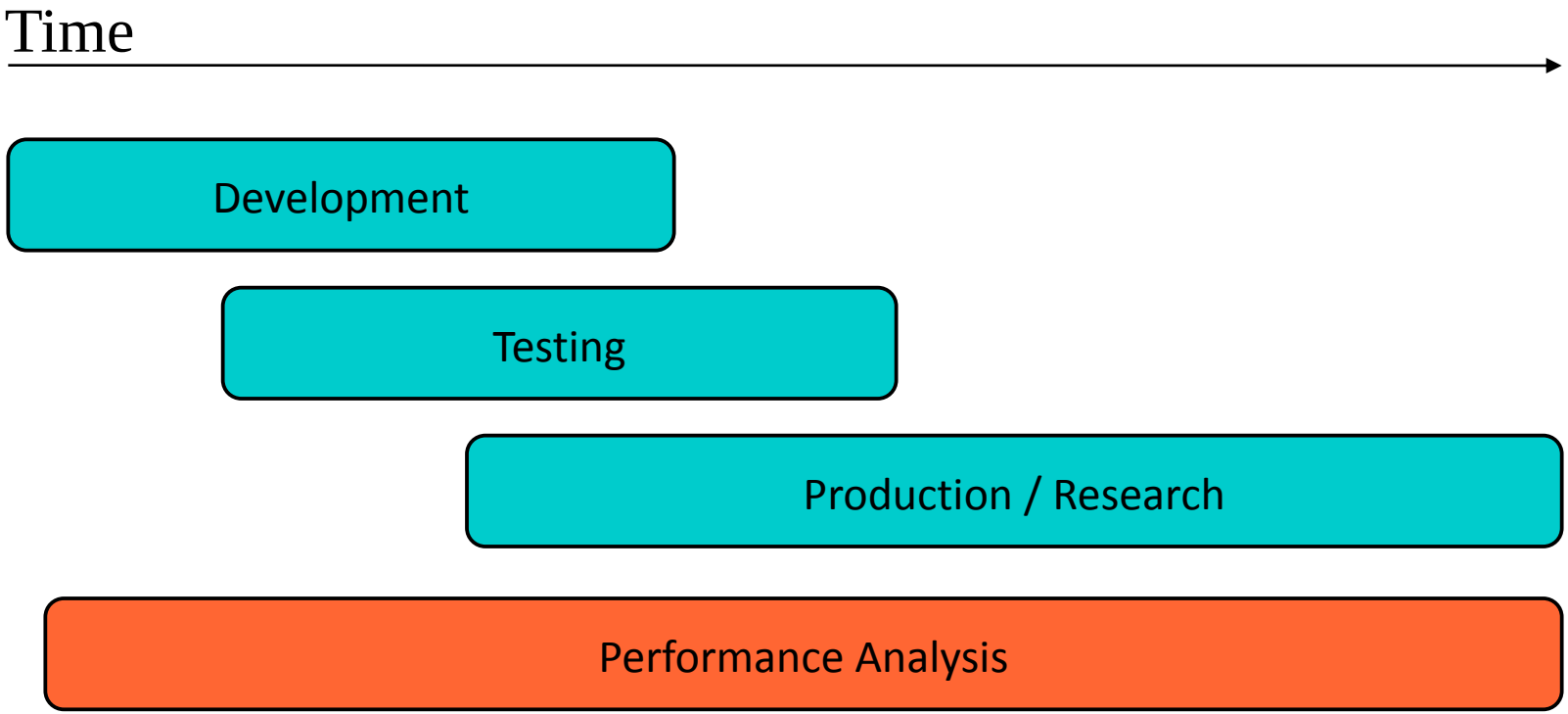
What is Performance Tuning?



What is Performance Tuning?



What is Performance Tuning?



Contents

- Indiana University and HPC@IU
- What is Performance Tuning
- Examples of Performance Engineering
- SPEC High Performance Group

Examples of Performance Engineering

- Karst Desktop – Entry Level HPC
- Workflow Tuning for WGS
- Trinity Performance Tuning
- Agro-IBIS Performance Tuning

Entry Level HPC

- A way to make supercomputing more user friendly
- A new way to login and interact with the Karst cluster
- A GUI/desktop instead of a terminal
- Based on ThinLinc, a Linux remote desktop solution using VNC and SSH

Entry Level HPC

User Desktop/Laptop

SSH

VNC through SSH

Karst Login Nodes (3)

Karst Desktop Nodes (13)

SSH

SSH

Karst Compute Nodes (256)

Karst Desktop - Features

- More user friendly interface than a terminal
  - A new front end to Karst, with new capabilities
  - Filesystem browser and file editors/viewers
- Graphical access to compute nodes (indirectly)
- Works more seamlessly compared to X forwarding
  - Addresses latency issues
  - Really great for GUI based applications
- Convenient data transfer/share options
- Supports long running tasks (disconnect / reconnect)
- Supports ssh keys

Karst Desktop – Use Cases

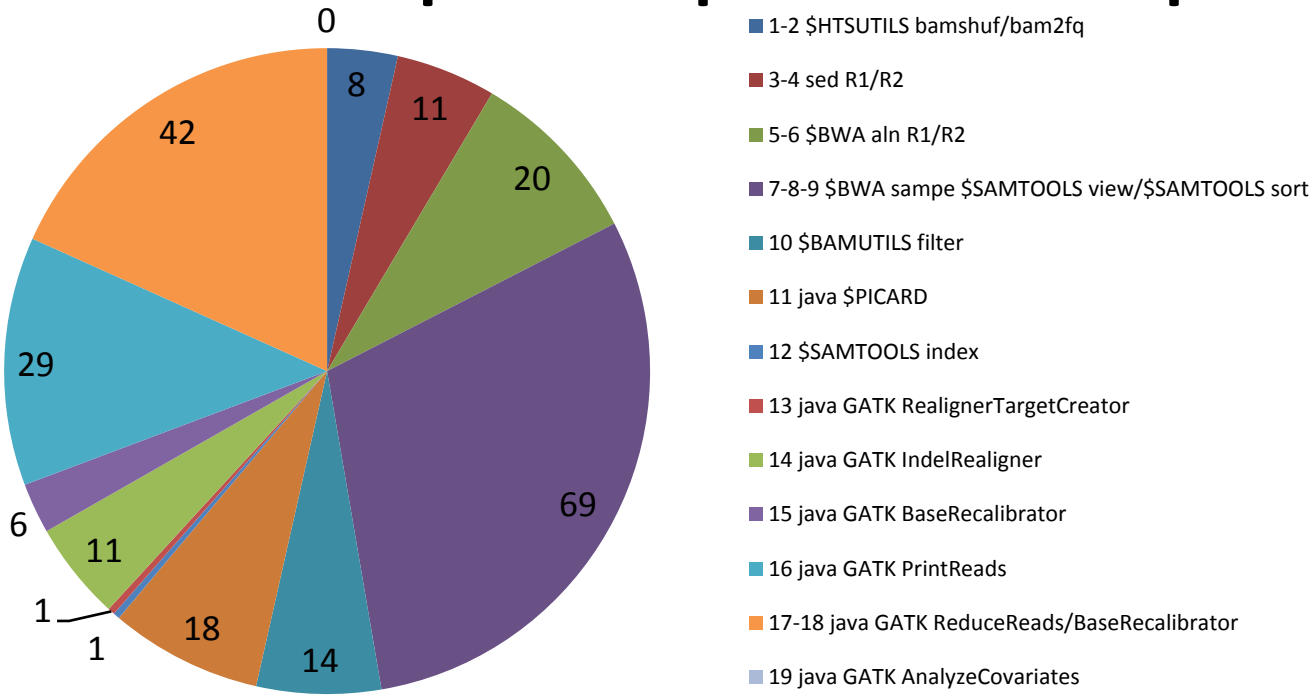
- Running mathematical and statistical applications
- GUIs of HPC applications such as Vampir, Allinea MAP, TotalView
- Visualization
- COMSOL Multiphysics Client/Server
- Data Enclave
- Desktop environment for crystallography tool suite
- Easy access to compilers for classes
- Long running data movement jobs
- Facilitates collaboration

Karst Desktop – Use Cases

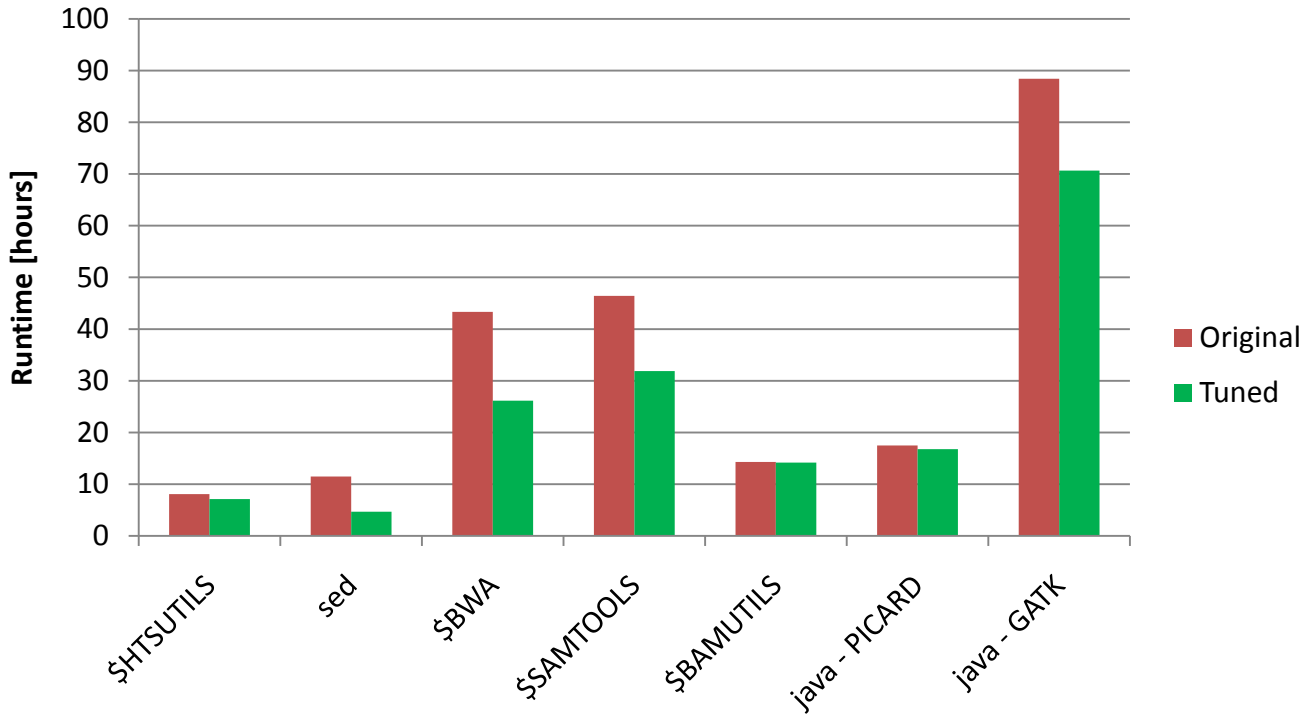
Workflow Tuning for WGS

- Broad Reference Pipeline (with very minor modifications)
  - 19 stages, 10 days of runtime
  - Going from 200 GB to 1 TByte per subject
- 818 Alzheimers patients
  - 150 TByte of total data
  - 100x coverage
- Final result:
  - Reduced pipeline runtime by 30% and output volume by 20%

Runtime per Pipeline Step



Runtime Comparision by Application



Trinity Performance Tuning

- Work done in 2012, together with ZIH and the BROAD Institute.
  - Matthias Lieber, Richard LeDuc, Brian Haas
- Resulted in a successful NIH grant proposal with BROAD and ZIH.

Trinity

- A bioinformatics code
  - Actually… a Perl script that calls a whole bunch of binaries – a workflow.
- Runtime can be hours, days, or even weeks, depending on input data and compute resource
- Open source with 3rd party dependencies

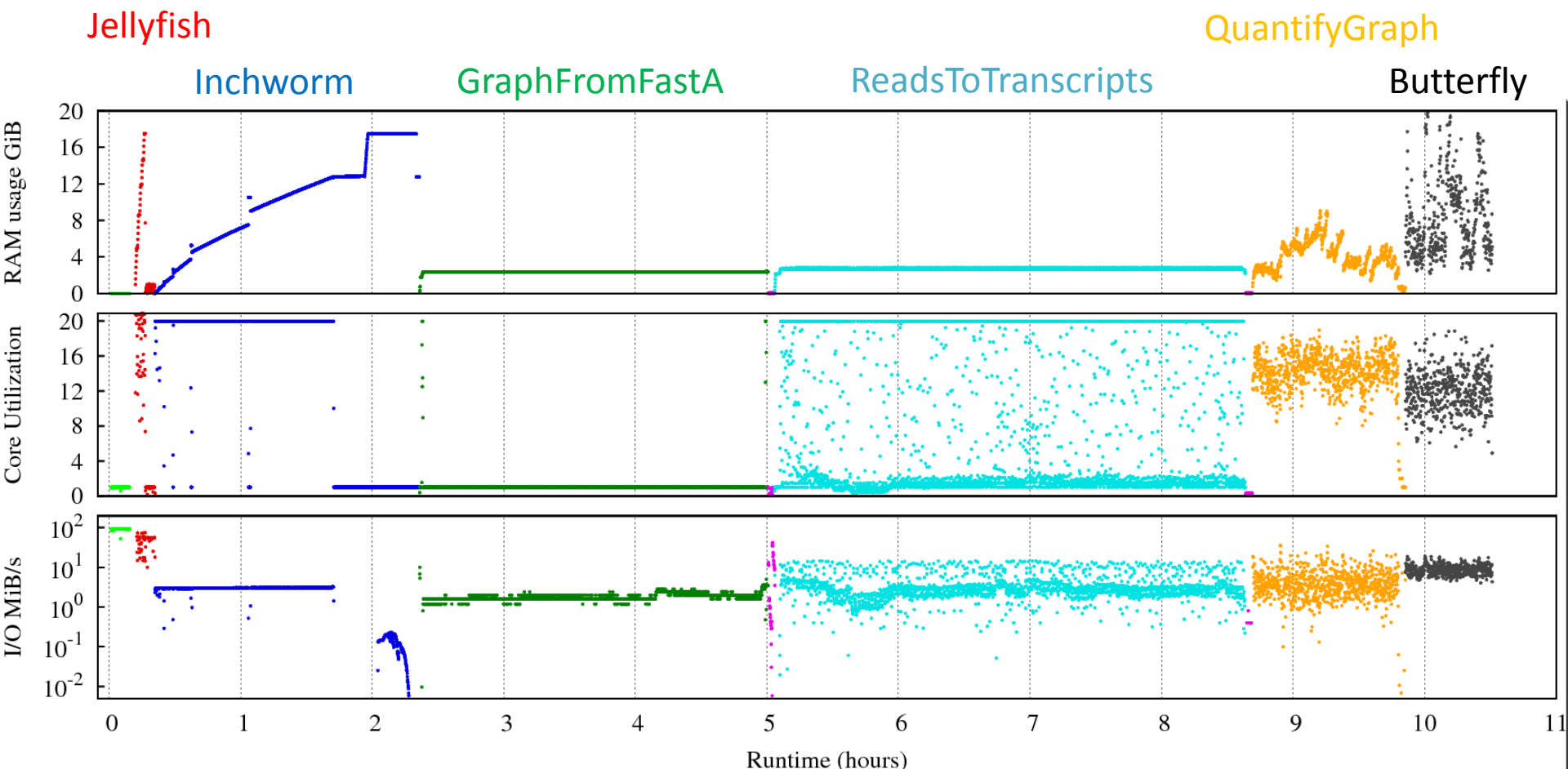
Our Plan

- Reproduce results from previous performance paper

- Perform general optimizations
- Optimize components

- Publish results
- Push modified source code into official repository

Performance Visualization - CollectL

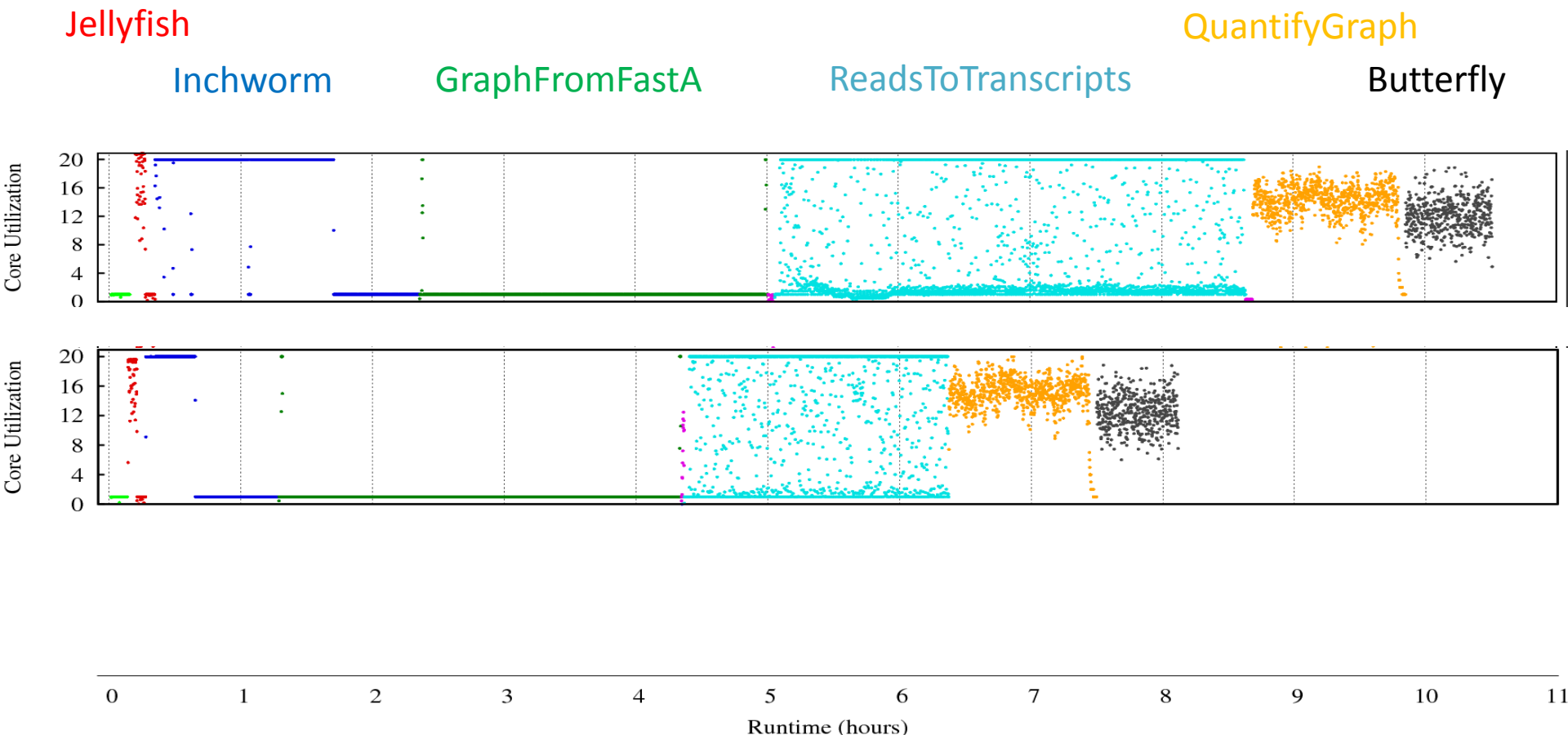


General Optimizations

- Only global optimizations that can be applied by end users
  - Compiler and runtime options

- Building with the Intel Compiler where possible
  - Using "-fast" compiler flags:
    -ipo -O3 -no-prec-div -static -xHost
- Thread placement and pinning using KMP_AFFINITY and "numactl"
- Input/Output and temporary files on "/dev/shm"

General Optimizations

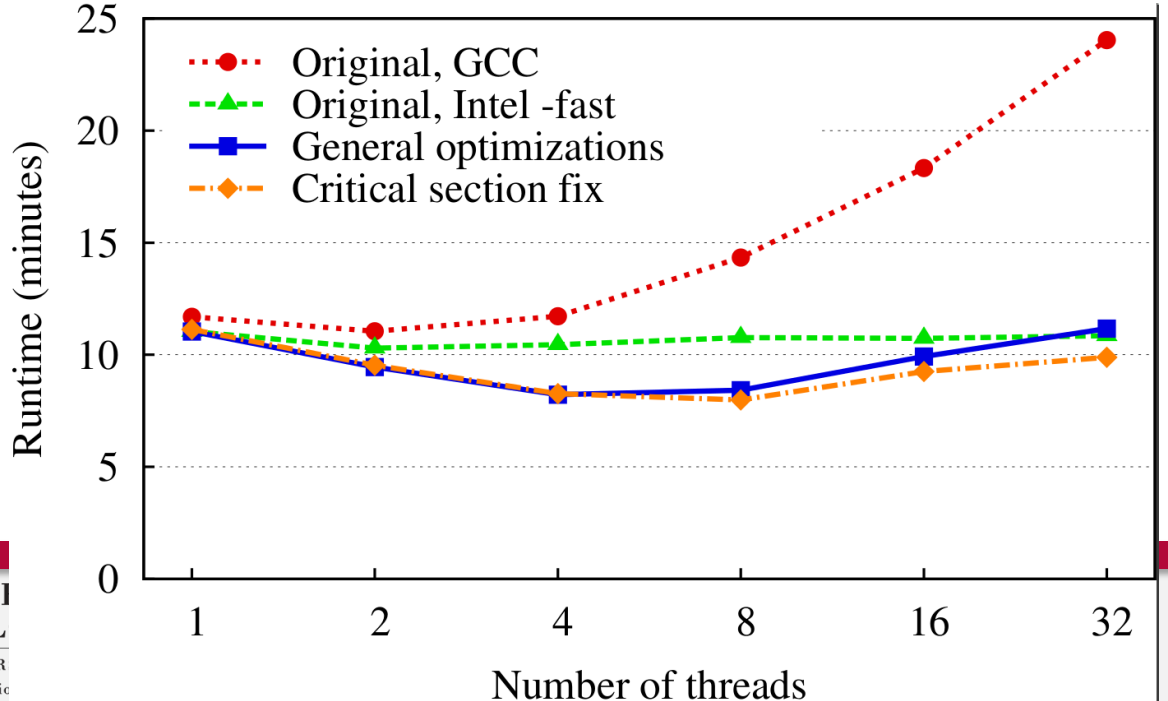


Optimizing Components

- Inchworm
- GraphFromFastA
- QuantifyGraph
- Other components

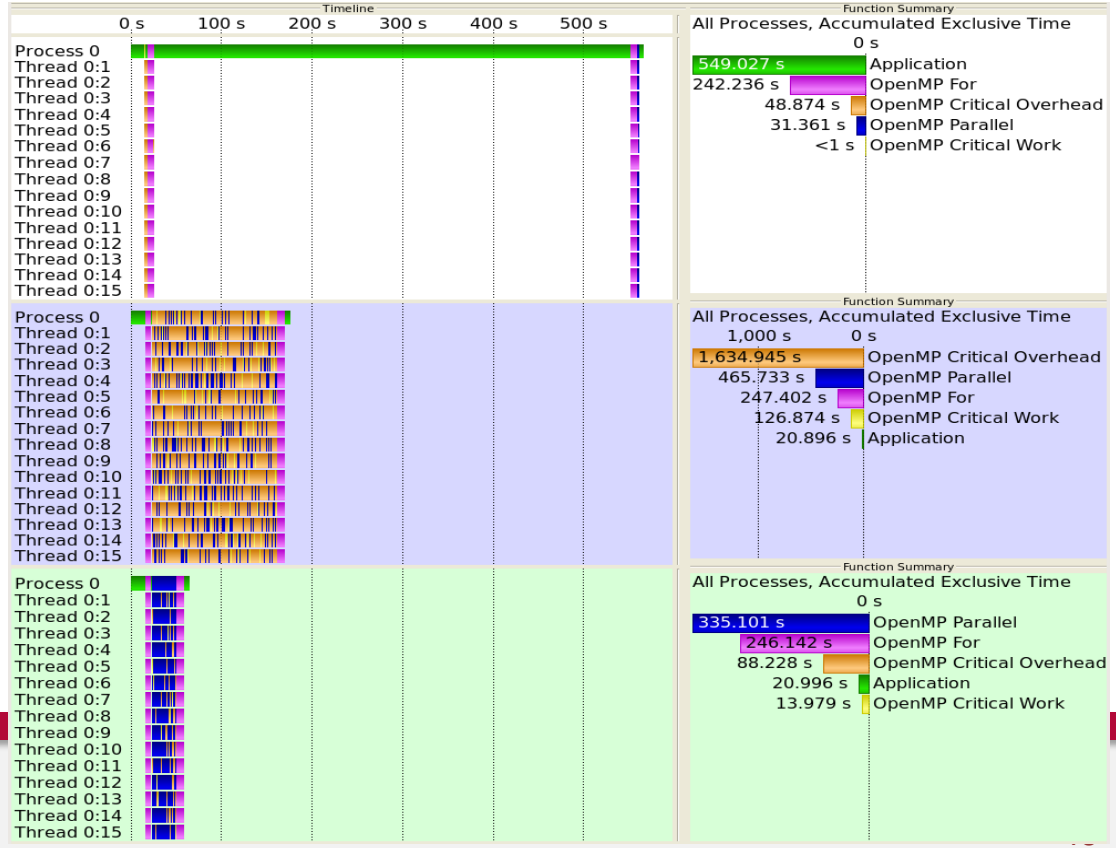
Optimizing Inchworm

- Intel's OpenMP runtime seems superior to GCC's, for this workload



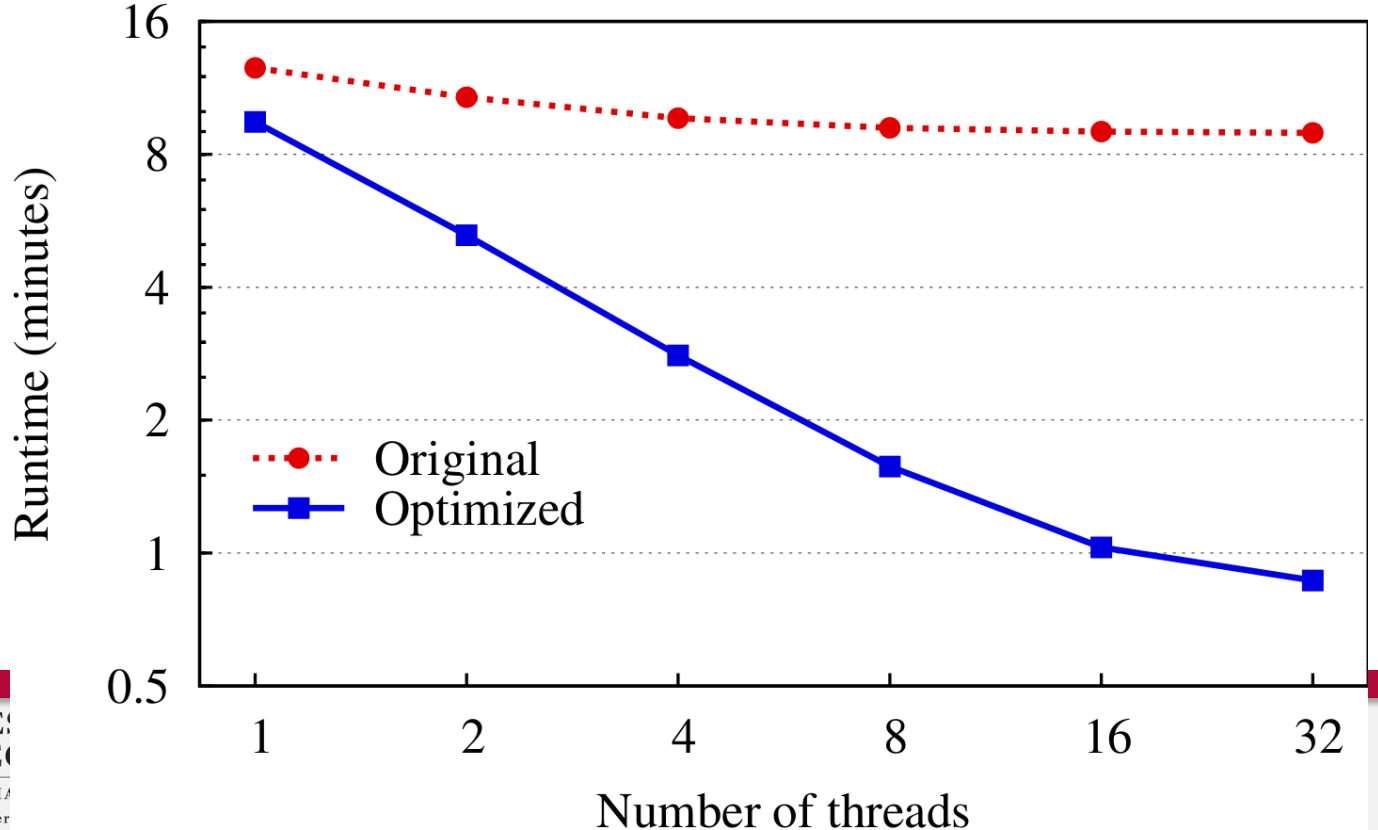
Optimizing GraphFromFastA

- Parallelizing read counting phase
- Optimized file input to reduce OpenMP critical section
- 10x faster on 32 cores



Optimizing GraphFromFastA

- Improved scalability.

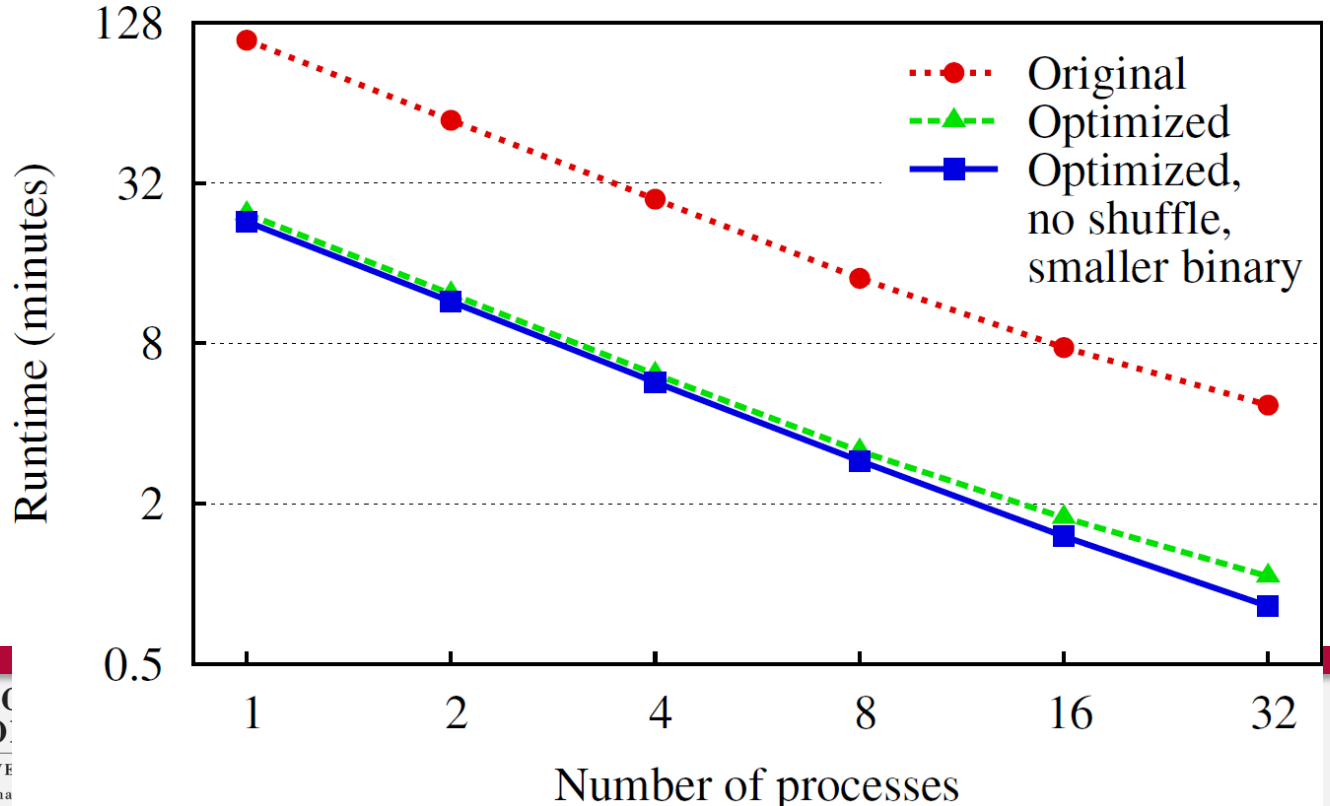


Optimizing QuantifyGraph

- Thousands of embarrassingly parallel tasks, with runtimes of 160 ms to 25 min
- Optimized relational operator “<”
- Reducing “system()” calls
- Reducing the read buffer from 200MB to 1kB
- 5x faster on 32 cores

Optimizing QuantifyGraph

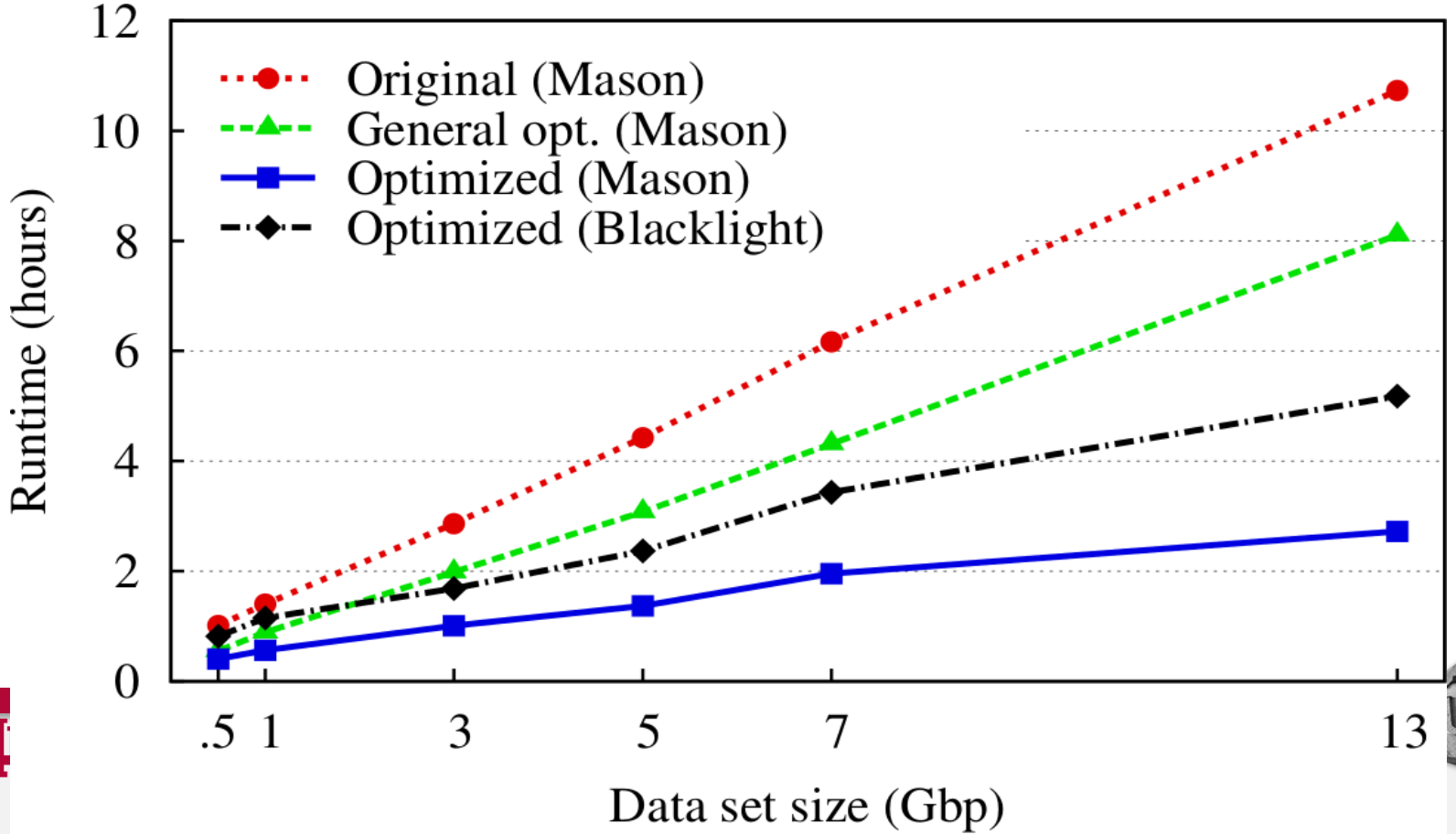
- Improved scalability.



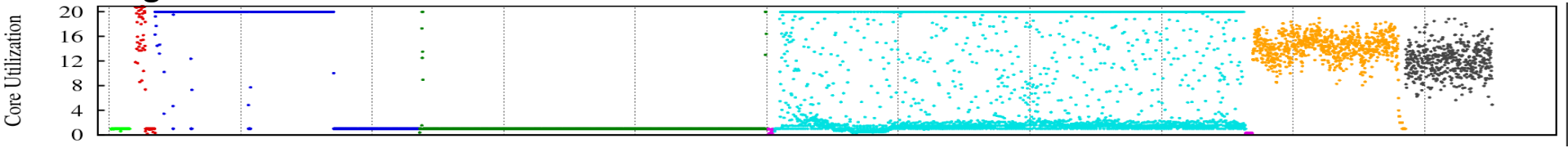
Optimizing Other Components

- Increasing the maximum for the "--CPU" parameter from 22 to 64
- Converting input files in parallel
- Setting "--max_memory" for Jellyfish to 20G, which reduces the number of times it flushes data
- Reduce Java GC threads for Butterfly to 4 per JVM

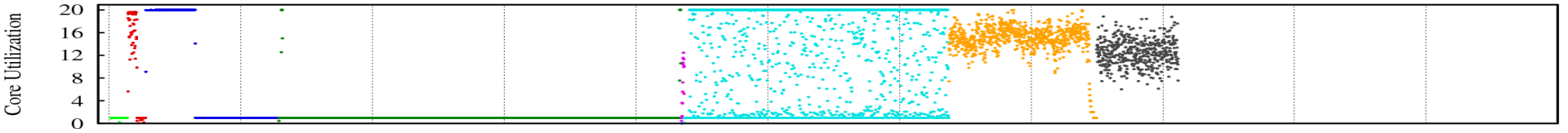
Final Results



Original

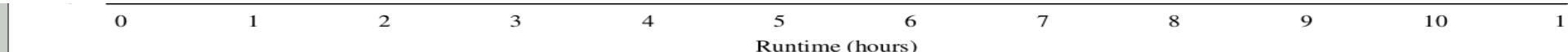


General Optimizations



Component Tuning

Jellyfish Inchworm GraphFromFastA ReadsToTranscripts QuantifyGraph Butterfly



Agro-IBIS

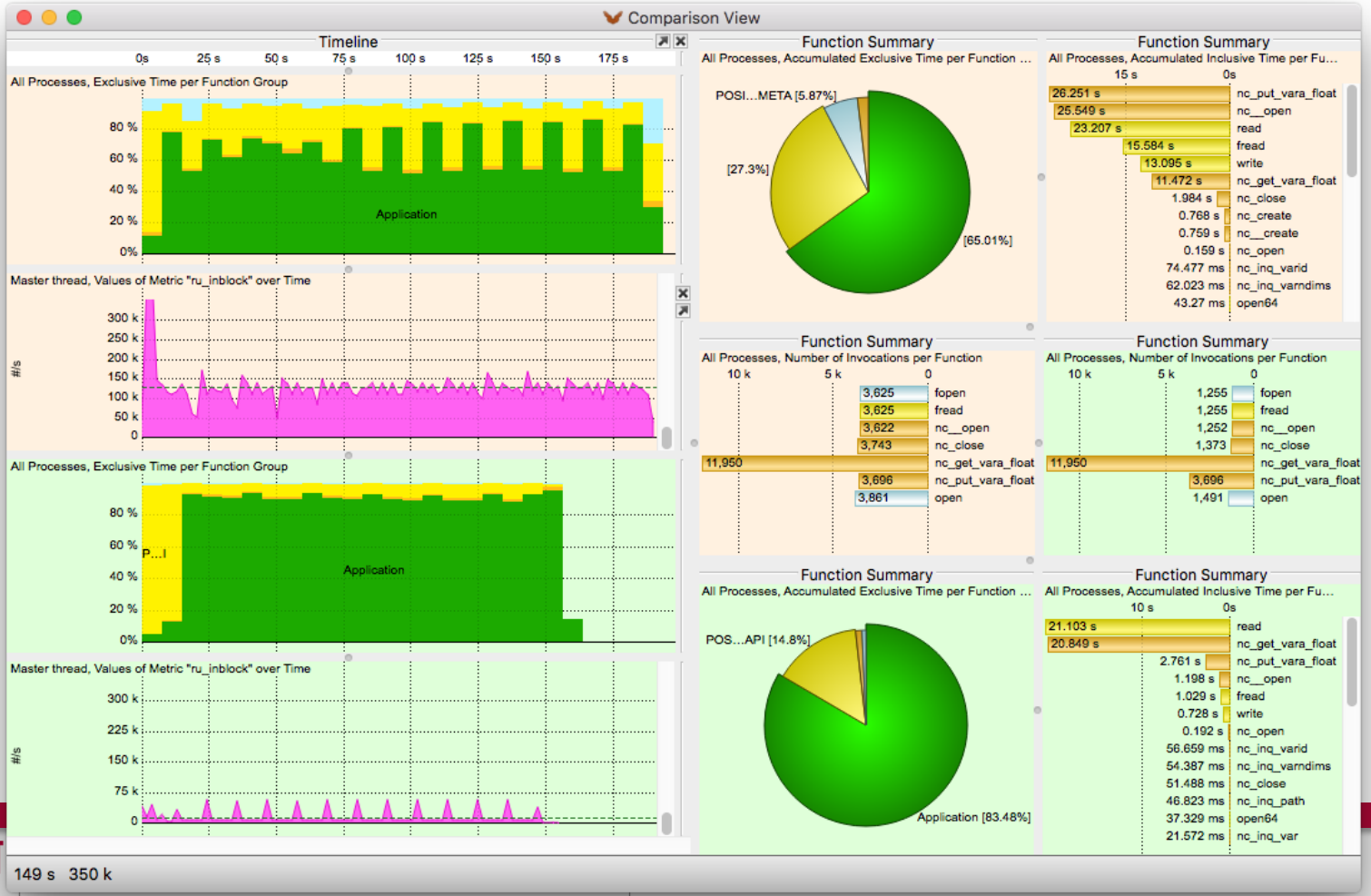
- Simulates agricultural ecosystems
  - Inputs include climate and weather data, farming decisions, and landscape properties
  - Outputs include physical state variables, fluxes, and agricultural parameters
  - Widely validated results for Midwestern US
- Serial, Fortran code
- Data is available to simulate at much larger scale
- Need to develop an HPC implementation of Agro-IBIS to solve large-scale models

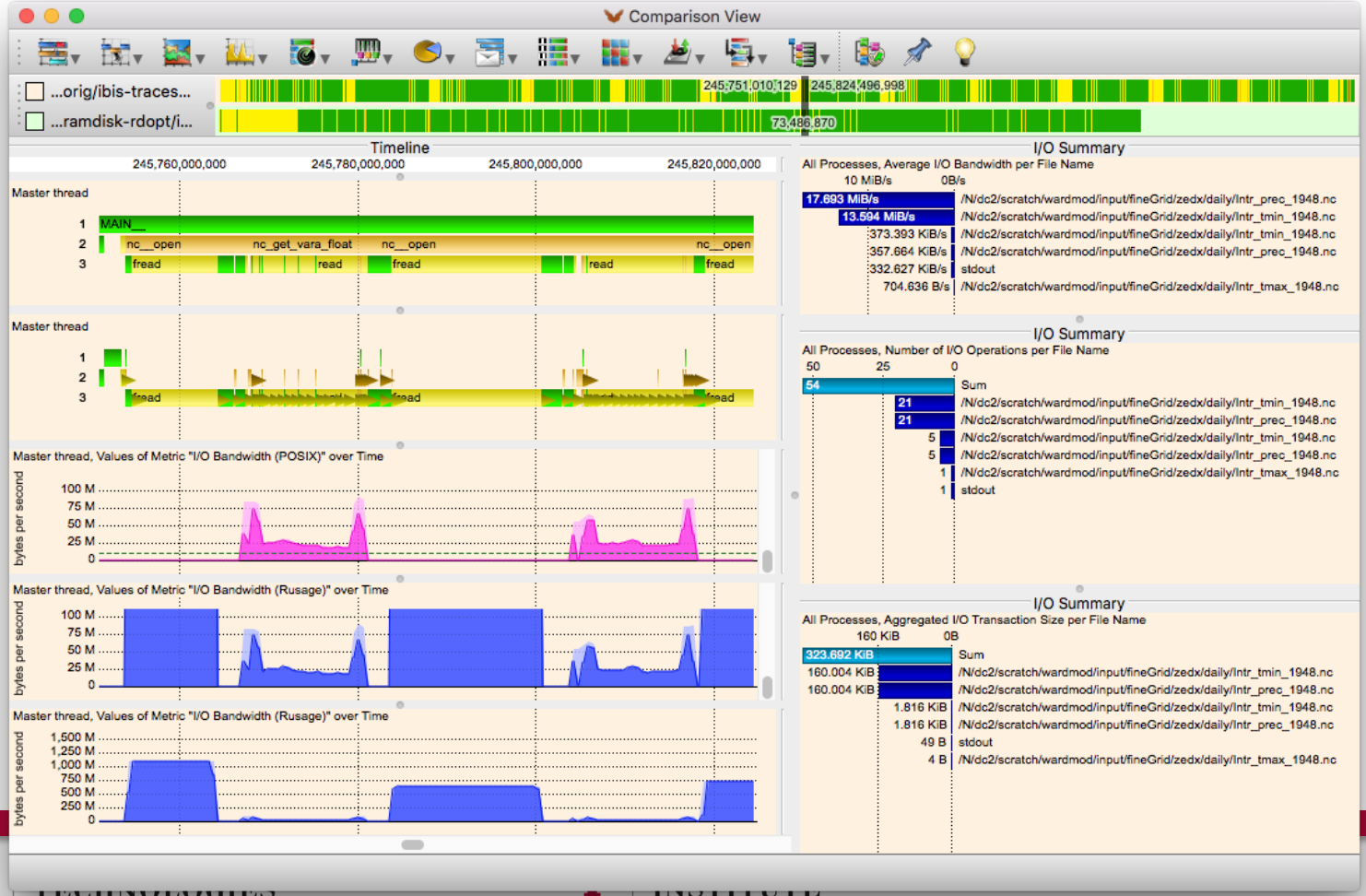
Agro-IBIS

- Development: Gains and Constraints
  - Strong desire to maintain consistency with community code
  - Optimizations desired to be drop-in or easily integrated with downloaded code
  - Implementation of netCDF library for standardized, optimized data storage
  - Parallel MPI wrapper written in C++ to manage domain decomposition and job launching
  - Previously unrecognized I/O bottleneck waiting to show up in parallel runs

Agro-IBIS

- Method for running IBIS puts a lot of strain on the filesystems used
  - Inputs and outputs for each IBIS run are separate file trees
  - IBIS instances scale perfectly – if you could ignore I/O cost
  - Very easy to tax the MDS without realizing it
  - This is just an example of what any conventional, serial app would do when domain decomposition doesn't take I/O into account.


Comparison View
Timeline
0s 25 s 50 s 75 s 100 s 125 s 150 s 175 s
All Processes, Exclusive Time per Function Group
80 %
60 %
40 %
20 %
0%
Application
Master thread, Values of Metric "ru_inblock" over Time
300 k
250 k
200 k
150 k
100 k
50 k
0
#/s
All Processes, Exclusive Time per Function Group
P...I
Application
Master thread, Values of Metric "ru_inblock" over Time
300 k
225 k
150 k
75 k
0
#/s
149 s 350 k
Function Summary
All Processes, Accumulated Exclusive Time per Function ...
POSI...META [5.87%]
[27.3%]
[65.01%]
Function Summary
All Processes, Accumulated Inclusive Time per Fu...
15 s 0s
26.251 s nc_put_vara_float
25.549 s nc__open
23.207 s read
15.584 s fread
13.095 s write
11.472 s nc_get_vara_float
1.984 s nc_close
0.768 s nc_create
0.759 s nc__create
0.159 s nc_open
74.477 ms nc_inq_varid
62.023 ms nc_inq_varndims
43.27 ms open64
Function Summary
All Processes, Number of Invocations per Function
10 k 5 k 0
3,625 fopen
3,625 fread
3,622 nc__open
3,743 nc_close
11,950 nc_get_vara_float
3,696 nc_put_vara_float
3,861 open
Function Summary
All Processes, Number of Invocations per Function
10 k 5 k 0
1,255 fopen
1,255 fread
1,252 nc__open
1,373 nc_close
11,950 nc_get_vara_float
3,696 nc_put_vara_float
1,491 open
Function Summary
All Processes, Accumulated Exclusive Time per Function ...
POS...API [14.8%]
Application [83.48%]
Function Summary
All Processes, Accumulated Inclusive Time per Fu...
10 s 0s
21.103 s read
20.849 s nc_get_vara_float
2.761 s nc_put_vara_float
1.198 s nc__open
1.029 s fread
0.728 s write
0.192 s nc_open
56.659 ms nc_inq_varid
54.387 ms nc_inq_varndims
51.488 ms nc_close
46.823 ms nc_inq_path
37.329 ms open64
21.572 ms nc_inq_var


Comparison View
...orig/ibis-traces...
...ramdisk-rdopt/i...
245,751,010,129
245,824,496,998
73,486,870
Timeline
245,760,000,000
245,780,000,000
245,800,000,000
245,820,000,000
Master thread
MAIN__
nc__open
nc_get_vara_float
nc__open
nc__open
fread
read
fread
read
fread
Master thread, Values of Metric "I/O Bandwidth (POSIX)" over Time
Master thread, Values of Metric "I/O Bandwidth (Rusage)" over Time
Master thread, Values of Metric "I/O Bandwidth (Rusage)" over Time
bytes per second
100 M
75 M
50 M
25 M
0
1,500 M
1,250 M
1,000 M
750 M
500 M
250 M
I/O Summary
All Processes, Average I/O Bandwidth per File Name
10 MiB/s
0B/s
17.693 MiB/s
13.594 MiB/s
373.393 KiB/s
357.664 KiB/s
332.627 KiB/s
704.636 B/s
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_prec_1948.nc
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_tmin_1948.nc
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_tmin_1948.nc
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_prec_1948.nc
stdout
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_tmax_1948.nc
I/O Summary
All Processes, Number of I/O Operations per File Name
50
25
0
54
Sum
21
21
5
5
1
1
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_tmin_1948.nc
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_prec_1948.nc
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_tmin_1948.nc
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_prec_1948.nc
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_tmax_1948.nc
stdout
I/O Summary
All Processes, Aggregated I/O Transaction Size per File Name
160 KiB
0B
323.692 KiB
160.004 KiB
160.004 KiB
1.816 KiB
1.816 KiB
49 B
4 B
Sum
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_tmin_1948.nc
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_prec_1948.nc
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_tmin_1948.nc
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_prec_1948.nc
stdout
/N/dc2/scratch/wardmod/input/fineGrid/zedx/daily/Intr_tmax_1948.nc

Contents

- Indiana University and HPC@IU
- What is Performance Tuning
- Examples of Performance Engineering
- SPEC HPG

Standard Performance Evaluation Corpor.

- SPEC is a non-profit corporation formed to "establish, maintain and endorse a standardized set of relevant benchmarks that can be applied to the newest generation of high-performance computers"
- Composed of four groups
  - Graphics and Workstation Performance Group (GWPG)
  - High Performance Group (HPG)
  - Open Systems Group (OSG)
  - Research Group (RG)
- https://www.spec.org

SPEC High Performance Group

- Develops benchmarks to represent high-performance computing applications for standardized, cross-platform performance evaluation.
- Benchmarks
  - SPEC OMP2012
  - SPEC MPI2007
  - SPEC ACCEL

SPEC ACCEL 1.2 – OpenMP Target

- Version 1.2 of the SPEC ACCEL benchmark was released this week.
- Addition of OpenMP suite with target directives

SPEC HPG Search Program

- We are building a new benchmark.
- MPI+X; Where X can be:
  - Nothing
  - Accelerator paradigms: CUDA, OpenACC, OpenMP4, …
  - Parallel paradigms: OpenMP, Threads, …
  - Libraries like Kokkos, TBB, MKL, …
- https://www.spec.org/hpg/search/

Acknowledgement

- Tom Doak, Arvind Gopu, Abhinav Thota, Dave Hancock, Richard Meraz for providing material for this presentations.
- Matthias Lieber for his work on Trinity.
- Huian Li for his work on the Alzheimers project.
- Holger Brunst, Shawn Slaving, Steven Simms, Cicada Dennis for their work on Agro-IBIS.
- RT leadership for all the support over the last couple of years.

This material is based in part upon work supported by the National Science Foundation under Grant Numbers ACI-1445604, DBI-1458641 and ABI-1062432. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

This research was supported by the National Cancer Institute Information Technology in Cancer Research program of the National Institutes of Health under award number 5U24CA180922-03.

Thank you!

Questions?