Outline

- What is Interactive Parallelization Tool (IPT)?

- Parallelizing applications using IPT

- Survey Link

IPT – How can it help you?

- IPT is a tool that assists users in learning parallel programming concepts without feeling burdened with the information on the syntax of MPI/OpenMP/CUDA

- If you know **what** to parallelize and **where**, IPT can help you with the syntax (of MPI/OpenMP/CUDA) and typical code reengineering for parallelization
  - IPT is not a 100% automatic tool for parallelization
  - Familiarize yourself with the IPT user-guide

- C and C++ languages are supported as of now, Fortran will be supported in future

Discussion of High-Level Concepts

General Concepts Related to Parallel Programming:

- Data distribution/collection/reduction

- Synchronization

- Loop dependence analysis (exercise # 2)

Must know before using IPT

Specific to OpenMP:

- A **structured block** having a single entry and exit point

- Threads communicate with each other by reading/writing from/to a shared memory region

- Compiler directives for creating teams of threads, sharing the work among threads, and synchronizing the threads

- Library routines for setting and getting thread attributes

IPT can help with most of these

Additional Concepts Related to OpenMP:

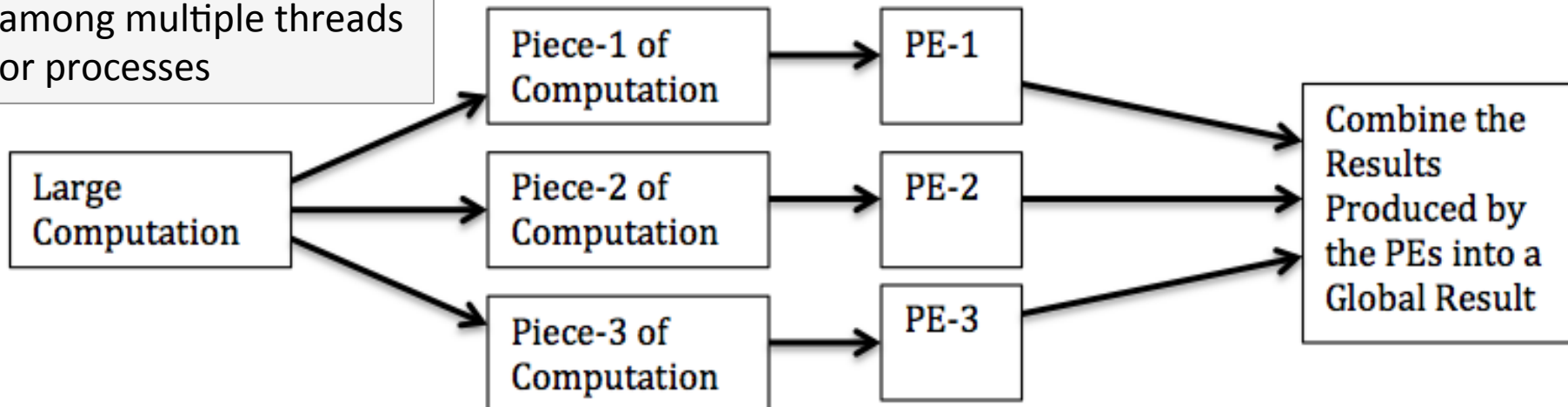- Environment variables to control run-time behavior

Programmer needs to decide at run-time

TACC

Process of Parallelizing a Large Number of Computations in a Loop (Conceptual)

- Loops can consume a lot of processing time when executed in serial mode
- Their total execution time can be reduced by sharing the computation-load among multiple threads or processes
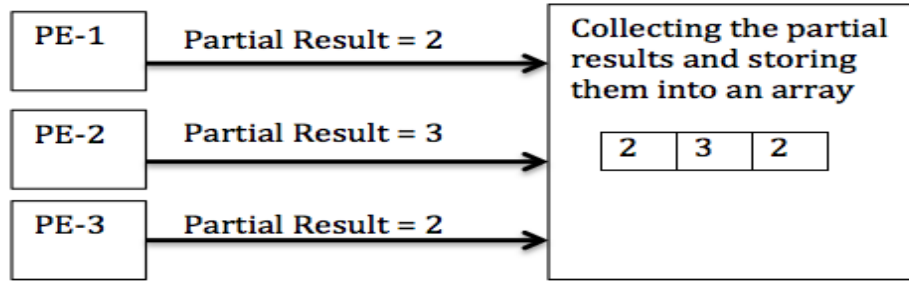
Large Computation Decomposed into Smaller Pieces

Each Piece of the Decomposed Computation is Mapped to a Processing Element (PE)

Large Computation

Piece-1 of Computation

Piece-2 of Computation

Piece-3 of Computation

PE-1

PE-2

PE-3

Combine the Results Produced by the PEs into a Global Result

Data Distribution/Collection/Reduction (Conceptual)

A Processing Element (PE) is a thread in OpenMP
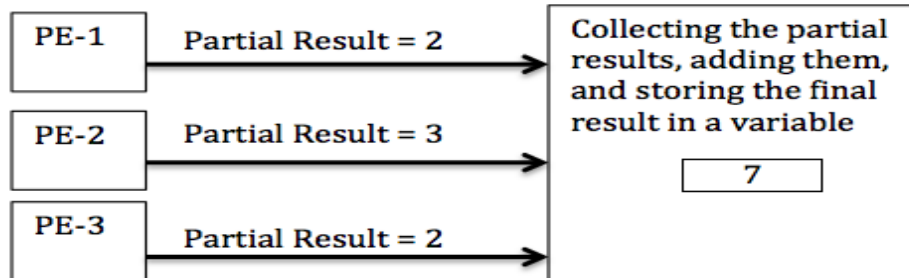
Synchronization (Conceptual)

- Synchronization helps in controlling the execution of threads relative to other threads in a team

- Synchronization constructs in OpenMP are as follows but you do not need to know them for using IPT

  master, single, atomic, critical, barrier, taskwait, flush, parallel {…}, ordered

Loop/Data Dependency (Analytical)

- Loop dependence implies that there are dependencies between the iterations of a loop that prevent its parallel processing
  - Analyze the code in the loop to determine the relationships between statements

- Analyze the order in which different statements access memory locations (data dependency)

- On the basis of the analysis, it may be possible to restructure the loop to allow multiple threads or processes to work on different portions of the loop in parallel

- For applications that have hotspots containing anti-dependency between the statements in a loop (leading to incorrect results upon parallelization), code refactoring should be done to remove the ante-dependency prior to parallelization.

Ready to try IPT?

IPT Demo

TACC | IPT   ⟶ Logi

INTERACTIVE PARALLELIZATION TOOL

Terminal   Compile   Run   Job History   Help

Welcome to IPT

You may use your
existing TACC
portal account or
create a new one.

Sign In to IPT

Enter Username

Enter Password

Sign In

Forgot your password? Click here

Create Account

Source Code Used for the Demo

https://tinyurl.com/y9auqfbu

Steps for the Demo

1. Download the code "circuit.c" from the Google drive
2. Login to the IPT web portal
3. Go to the "Terminal" tab
4. Scroll down to the "File Upload" section, and click on the browse button to select the downloaded file "circuit.c" from your computer
5. After the file is uploaded to the IPT web, you can start using IPT as follows:

```
ipt@43be716e9e1d:~$ IPT circuit.c
```

Link to the Survey – 5 questions

https://tinyurl.com/yc7mxwfb

Thanks a lot for you attention!

Questions? Email: rauta@tacc.utexas.edu

Looking for internship opportunities? Please contact us.

IPT: High-Level Overview