When to Apply Parallelism in Practice

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

• Understand the meaning of key concepts associated with parallel programming

• Know when to apply parallelism in practice
When to Apply Parallelism in Practice
When to Apply Parallelism in Practice

- Parallelism is not a panacea!!
When to Apply Parallelism in Practice

- Instead, parallelism works best under certain conditions
When to Apply Parallelism in Practice

- Instead, parallelism works best under certain conditions, e.g.
- When tasks are independent

![Diagram showing parallel execution flow]

When to Apply Parallelism in Practice
When to Apply Parallelism in Practice

• Instead, parallelism works best under certain conditions, e.g.
  • When tasks are independent

“Embarrassingly parallel” tasks have little/no dependency or need for communication between tasks or for sharing results between them

See en.wikipedia.org/wiki/Embarrassingly_parallel
When to Apply Parallelism in Practice

- Instead, parallelism works best under certain conditions, e.g.
  - When tasks are independent

"Embarrassing" in this context means "over-abundance" or "too much of a good thing"!

See en.wikipedia.org/wiki/Embarrassment_of_riches
When to Apply Parallelism in Practice

- Instead, parallelism works best under certain conditions, e.g.
  - When tasks are independent
  - When there’s lots of data & processing to perform

See [en.wikipedia.org/wiki/Terracotta_Army](en.wikipedia.org/wiki/Terracotta_Army)
When to Apply Parallelism in Practice

- Instead, parallelism works best under certain conditions, e.g.
  - When tasks are independent
  - When there’s lots of data & processing to perform

- \( N \) is the # of data elements to process
- \( Q \) quantifies CPU processing intensity

See: on-sw-integration.epischel.de/2016/08/05/parallel-stream-processing-with-java-8-stream-api
When to Apply Parallelism in Practice

• Instead, parallelism works best under certain conditions, e.g.
  • When tasks are independent
  • When there’s lots of data & processing to perform
  • When tasks neither block nor share mutable state

See henrikeichenhardt.blogspot.com/2013/06/why-shared-mutable-state-is-root-of-all.html
When to Apply Parallelism in Practice

- Instead, parallelism works best under certain conditions, e.g.
  - When tasks are independent
  - When there’s lots of data & processing to perform
  - When tasks neither block nor share mutable state
  - Hence Java’s focus on “Work-stealing”
  - To avoid blocking

See [en.wikipedia.org/wiki/Work_stealing](en.wikipedia.org/wiki/Work_stealing)
When to Apply Parallelism in Practice

• Instead, parallelism works best under certain conditions, e.g.
  • When tasks are independent
  • When there’s lots of data & processing to perform
  • When tasks neither block nor share mutable state
    • Hence Java’s focus on
      • “Work-stealing”
      • The “fork-join” paradigm
    • To avoid sharing mutable state

See [en.wikipedia.org/wiki/Fork-join_model](en.wikipedia.org/wiki/Fork-join_model)
When to Apply Parallelism in Practice

- Instead, parallelism works best under certain conditions, e.g.
  - When tasks are independent
  - When there’s lots of data & processing to perform
  - When tasks neither block nor share mutable state
  - When there are many cores and/or processors

![Diagram showing parallel tasks and their synchronization]

THE MORE, THE MERRIER!

See [en.wikipedia.org/wiki/Multi-core_processor](en.wikipedia.org/wiki/Multi-core_processor) & [en.wikipedia.org/wiki/Multiprocessing](en.wikipedia.org/wiki/Multiprocessing)
End of When to Apply Parallel Programming in Practice