Overview of Parallel Programming Concepts

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

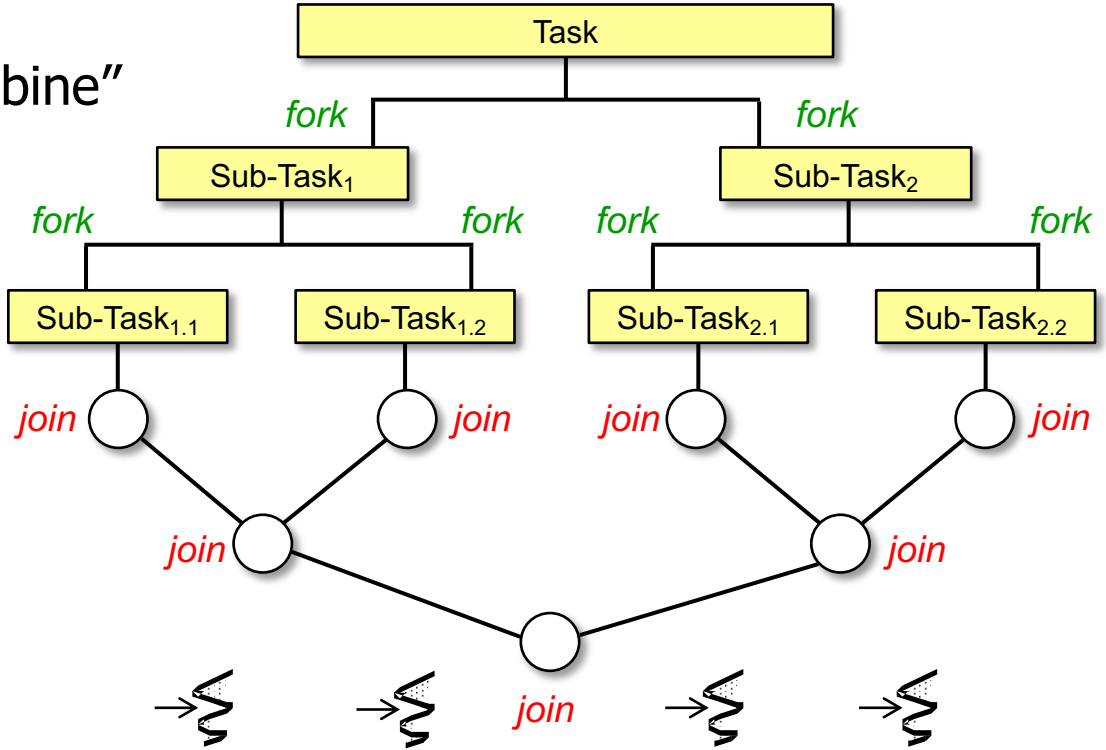
Professor of Computer Science

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
  - i.e., the "split, apply, combine" parallel processing model

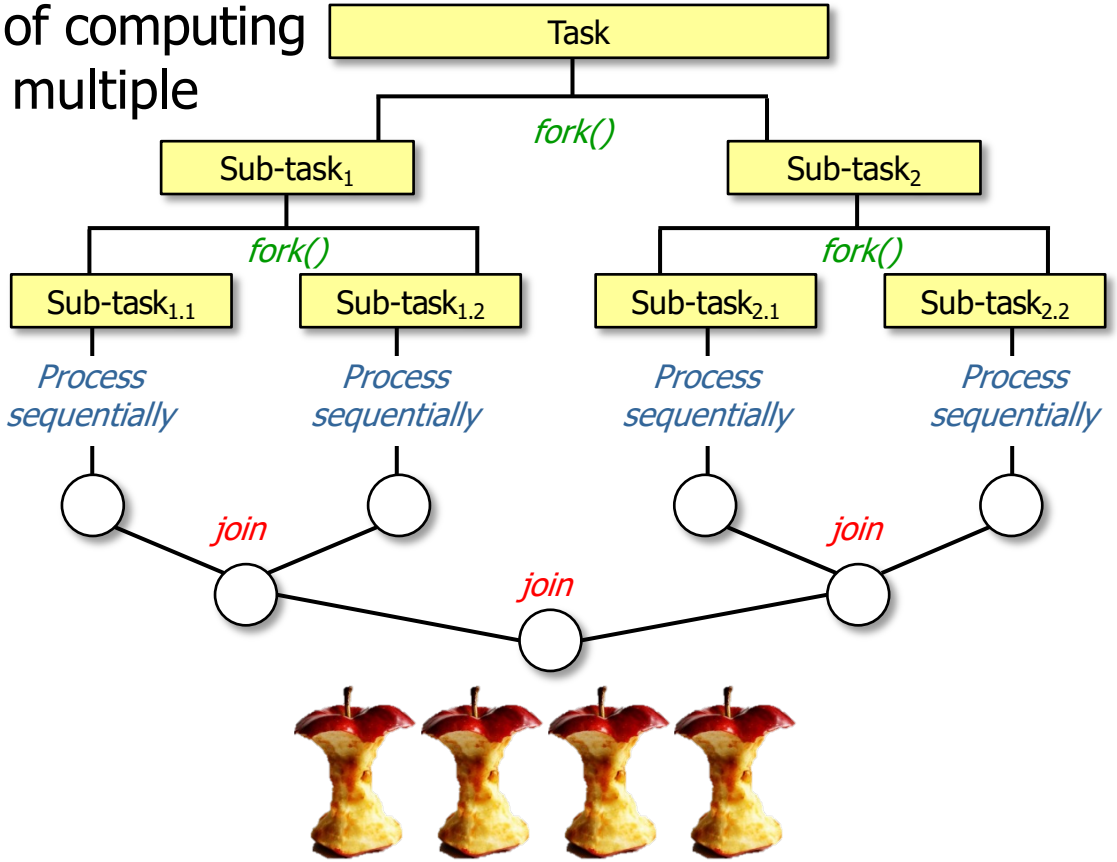


See en.wikipedia.org/wiki/Parallel_computing

An Overview of Parallel Programming

An Overview of Parallel Programming

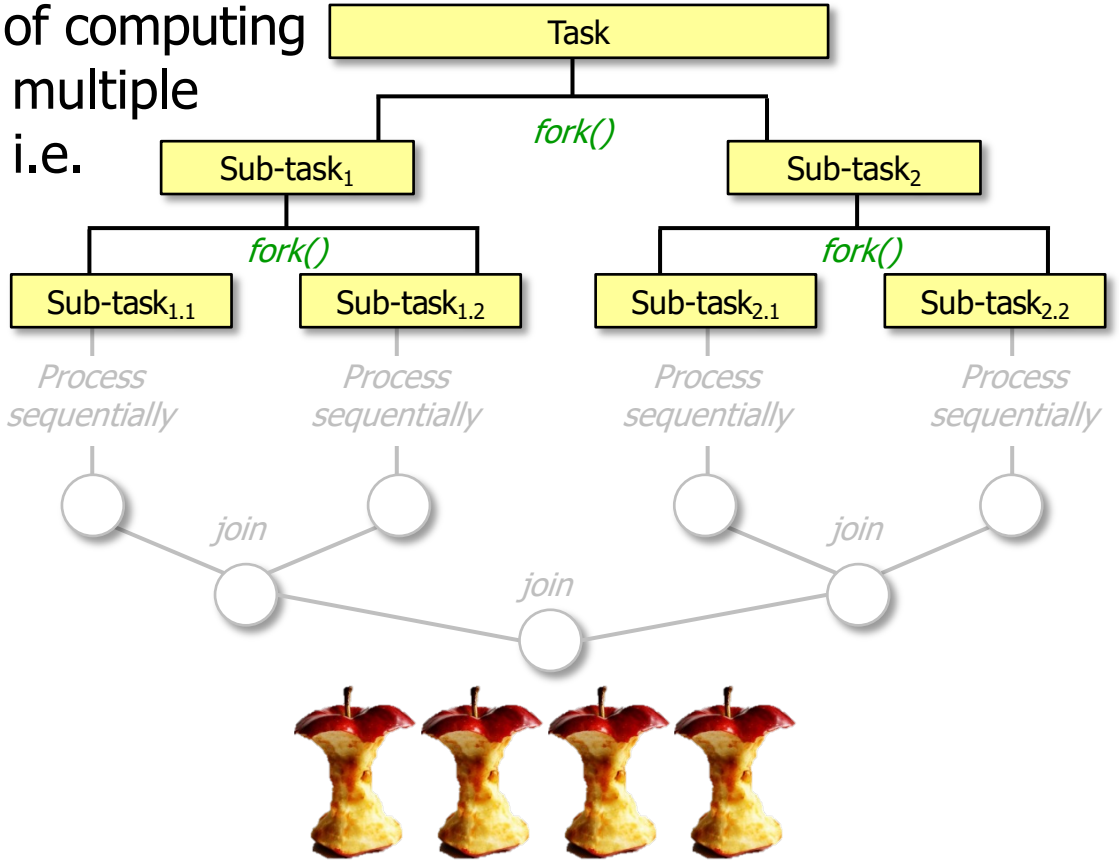
- Parallel programming is a form of computing that performs three phases on multiple processors or processor cores



See www.jstatsoft.org/article/view/v040i01/v40i01.pdf

An Overview of Parallel Programming

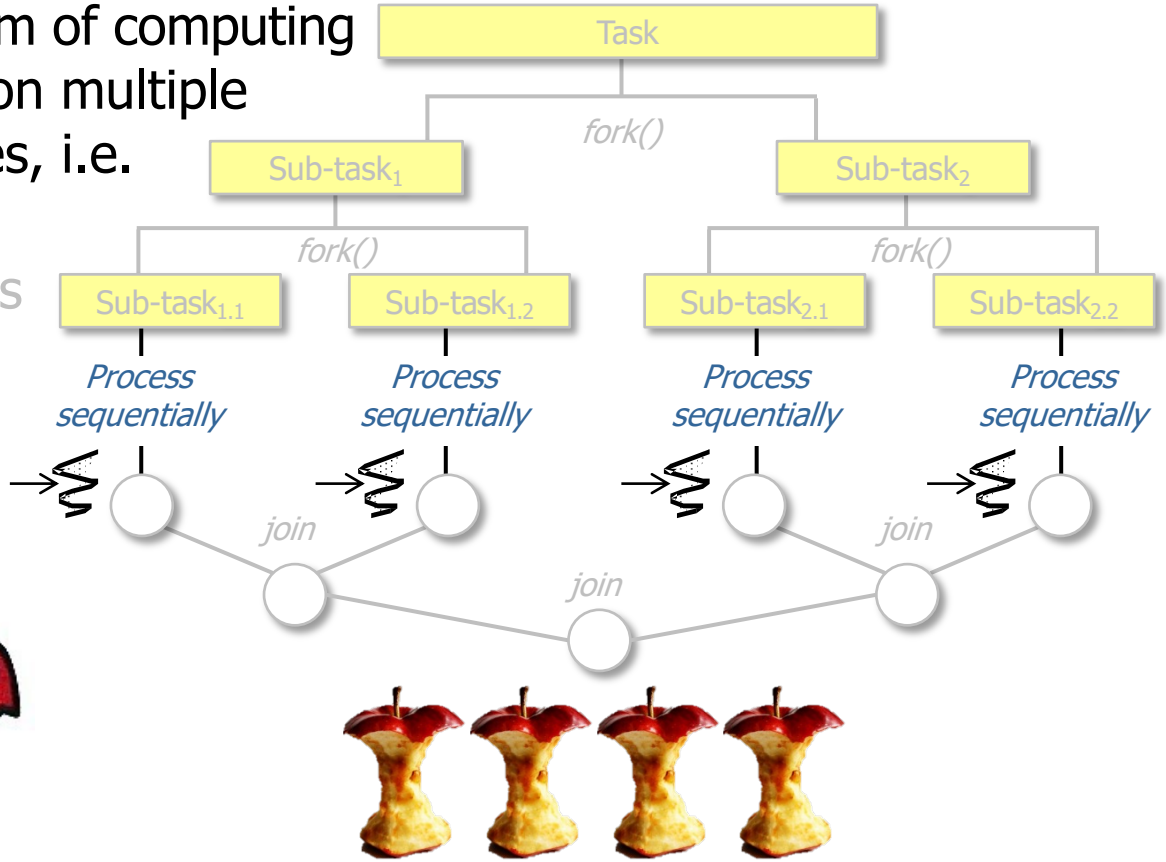
- Parallel programming is a form of computing that performs three phases on multiple processors or processor cores, i.e.
  - **Split** – partition an initial task into multiple sub-tasks

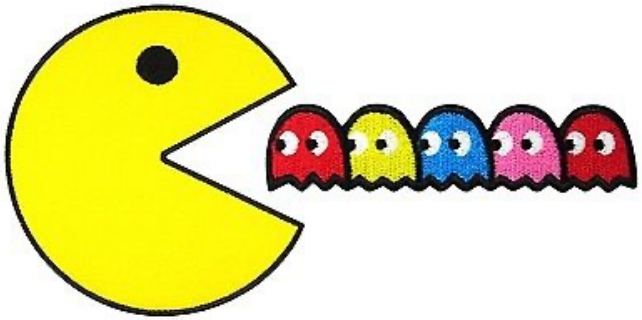


Ideally sub-tasks are split efficiently & evenly (& recursively until a threshold is met)

An Overview of Parallel Programming

- Parallel programming is a form of computing that performs three phases on multiple processors or processor cores, i.e.
  - **Split** – partition an initial task into multiple sub-tasks
  - **Apply** – Run independent sub-tasks in parallel

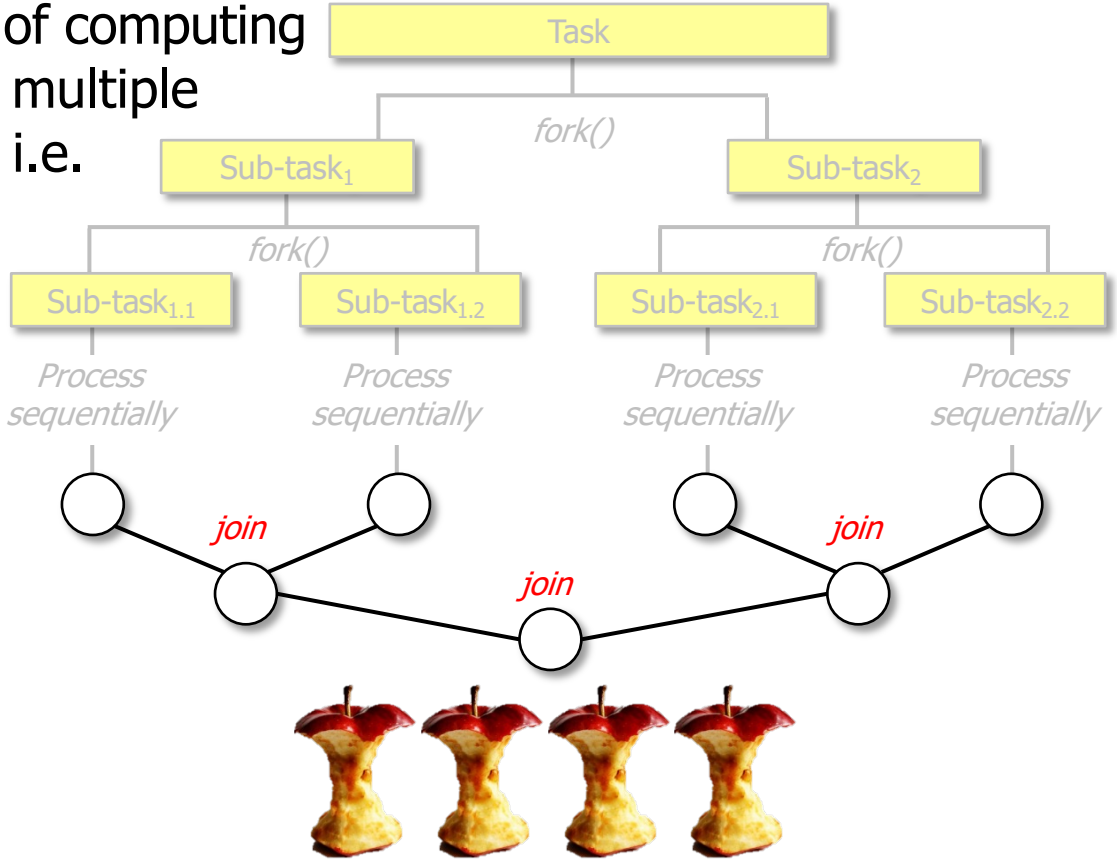


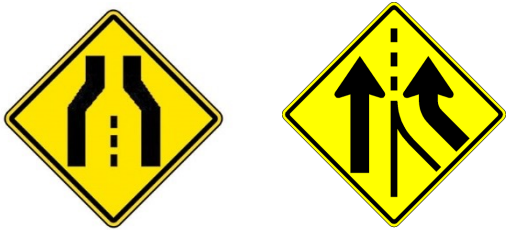


Each sub-task runs sequentially, but together they run in parallel

An Overview of Parallel Programming

- Parallel programming is a form of computing that performs three phases on multiple processors or processor cores, i.e.
  - **Split** – partition an initial task into multiple sub-tasks
  - **Apply** – Run independent sub-tasks in parallel
  - **Combine** – Merge the sub-results from sub-tasks into a single "reduced" result





The final reduced result can be a primitive value, an object, a collection, etc.

An Overview of Parallel Programming

- A key goal of parallel programming is to partition many tasks into sub-tasks & combine results *efficiently*

See developer.ibm.com/articles/j-java-streams-4-brian-goetz

An Overview of Parallel Programming

- A key goal of parallel programming is to partition many tasks into sub-tasks & combine results *efficiently*
  - Parallelism is thus an *optimization* of key performance characteristics

See en.wikipedia.org/wiki/Computer_performance

An Overview of Parallel Programming

- A key goal of parallel programming is to partition many tasks into sub-tasks & combine results *efficiently*
  - Parallelism is thus an *optimization* of key performance characteristics

See en.wikipedia.org/wiki/Up_to_eleven

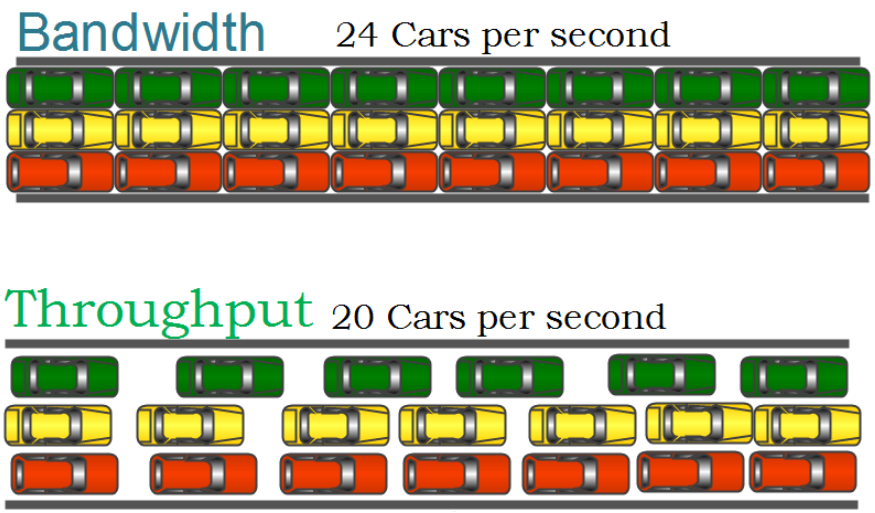
An Overview of Parallel Programming

- A key goal of parallel programming is to partition many tasks into sub-tasks & combine results *efficiently*
  - Parallelism is thus an *optimization* of key performance characteristics, e.g.,
    - *Throughput*
      - How many units of info a system can process within a given time

See en.wikipedia.org/wiki/Throughput

An Overview of Parallel Programming

- A key goal of parallel programming is to partition many tasks into sub-tasks & combine results *efficiently*
  - Parallelism is thus an *optimization* of key performance characteristics, e.g.,
    - *Throughput*
      - How many units of info a system can process within a given time
    - There's often a difference between max throughput vs. actual throughput



Peak performance is limited in practice by overheads like resource contention, software inefficiency, external dependencies, & interference

See www.comparitech.com/net-admin/throughput-vs-bandwidth

An Overview of Parallel Programming

- A key goal of parallel programming is to partition many tasks into sub-tasks & combine results *efficiently*
  - Parallelism is thus an *optimization* of key performance characteristics, e.g.,
    - *Throughput*
    - *Scalability*
      - A system's ability to handle a growing amount of workload

See en.wikipedia.org/wiki/Scalability

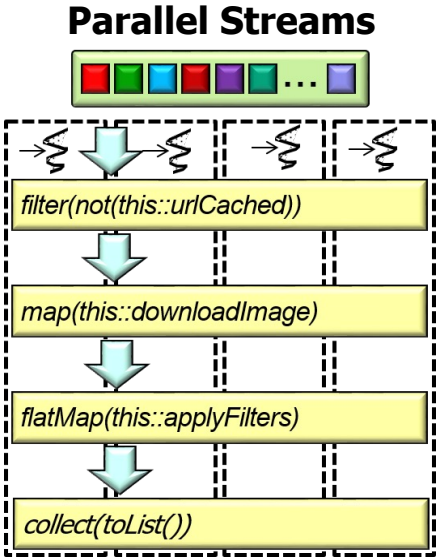
An Overview of Parallel Programming

- A key goal of parallel programming is to partition many tasks into sub-tasks & combine results *efficiently*
  - Parallelism is thus an *optimization* of key performance characteristics, e.g.,
    - *Throughput*
    - *Scalability*
      - A system's ability to handle a growing amount of workload
      - Scalability is often associated w/ cloud computing "autoscaling"

See en.wikipedia.org/wiki/Autoscaling

An Overview of Parallel Programming

- A key goal of parallel programming is to partition many tasks into sub-tasks & combine results *efficiently*
  - Parallelism is thus an *optimization* of key performance characteristics, e.g.,
    - *Throughput*
    - *Scalability*
      - A system's ability to handle a growing amount of workload
      - Scalability is often associated w/ cloud computing "autoscaling"
        - However, we focus on Java multi-core parallelism, not cloud parallelism



See reintech.io/blog/java-parallel-programming-utilizing-multiple-cores

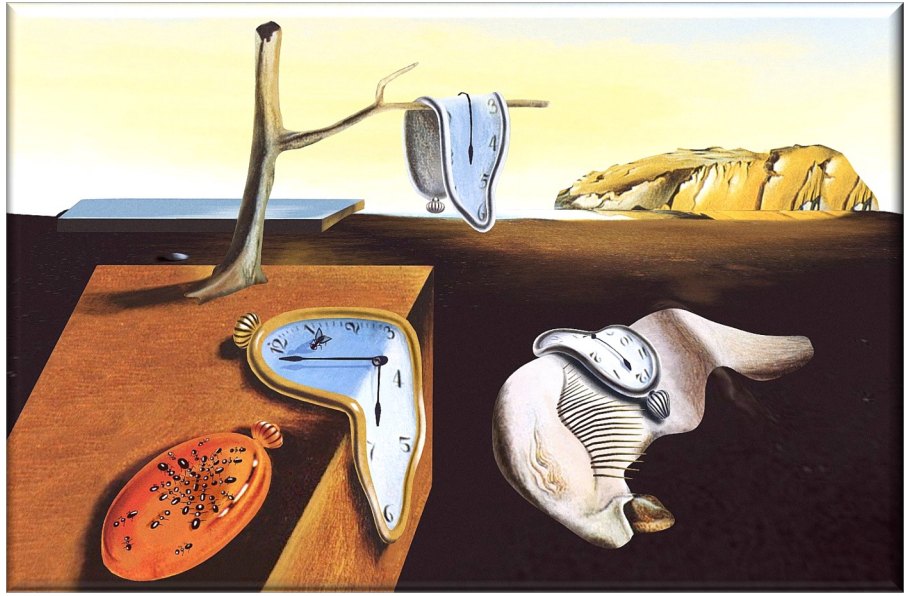
An Overview of Parallel Programming

- A key goal of parallel programming is to partition many tasks into sub-tasks & combine results *efficiently*
  - Parallelism is thus an *optimization* of key performance characteristics, e.g.,
    - *Throughput*
    - *Scalability*
    - *Latency*
      - The delay between a user's action & a system's response to that action

See en.wikipedia.org/wiki/Latency_(engineering)

An Overview of Parallel Programming

- A key goal of parallel programming is to partition many tasks into sub-tasks & combine results *efficiently*
  - Parallelism is thus an *optimization* of key performance characteristics, e.g.,
    - *Throughput*
    - *Scalability*
    - *Latency*
      - The delay between a user's action & a system's response to that action
      - Minimizing latency (& jitter) is often essential for mission- & safety-critical real-time systems



See en.wikipedia.org/wiki/Real-time_computing

End of Overview of Parallel Programming Concepts

Discussion Questions

1. Which of the following are key goals of parallelism?
   a. *Parallelism is used to offload work from a non-blocking user interface thread to background threads that can block*
   b. *Parallelism is used to efficiently partition tasks into sub-tasks & combine results*
   c. *Parallelism focuses on sharing resources safely/efficiently & avoid concurrency hazards*
   d. *Parallelism focuses on optimizing performance by avoiding resource sharing & not blocking*