CPSC 531: System Modeling and Simulation

Carey Williamson
Department of Computer Science
University of Calgary
Fall 2017

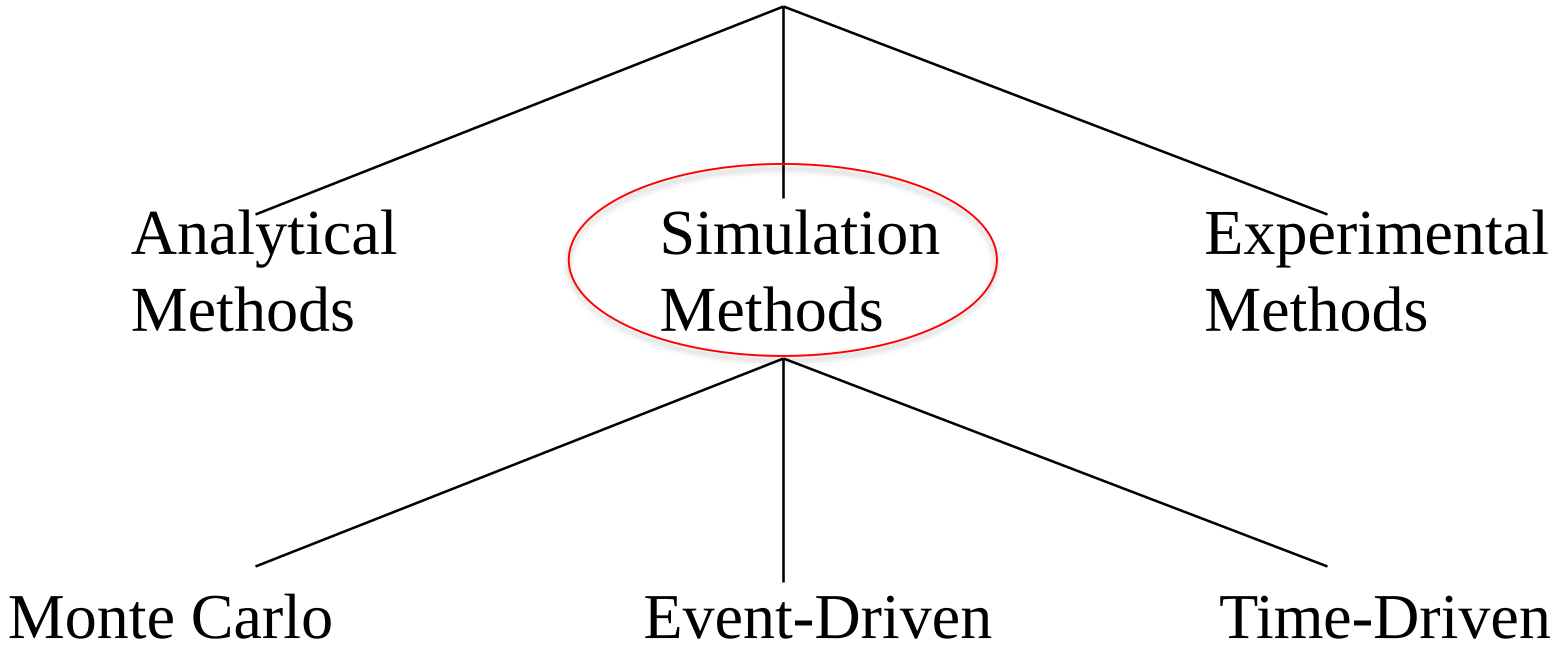
Simulation Methods

- Outline:
  - Basics of simulation modeling
  - Overview of simulation paradigms
    - Monte Carlo simulation
    - Time-driven simulation
    - Event-driven simulation
  - Simulation implementation approaches
    - Sequential vs parallel vs distributed
  - Technical issues for parallel simulation

Performance Evaluation

- Analytical Methods
- Simulation Methods
- Experimental Methods

Performance Evaluation



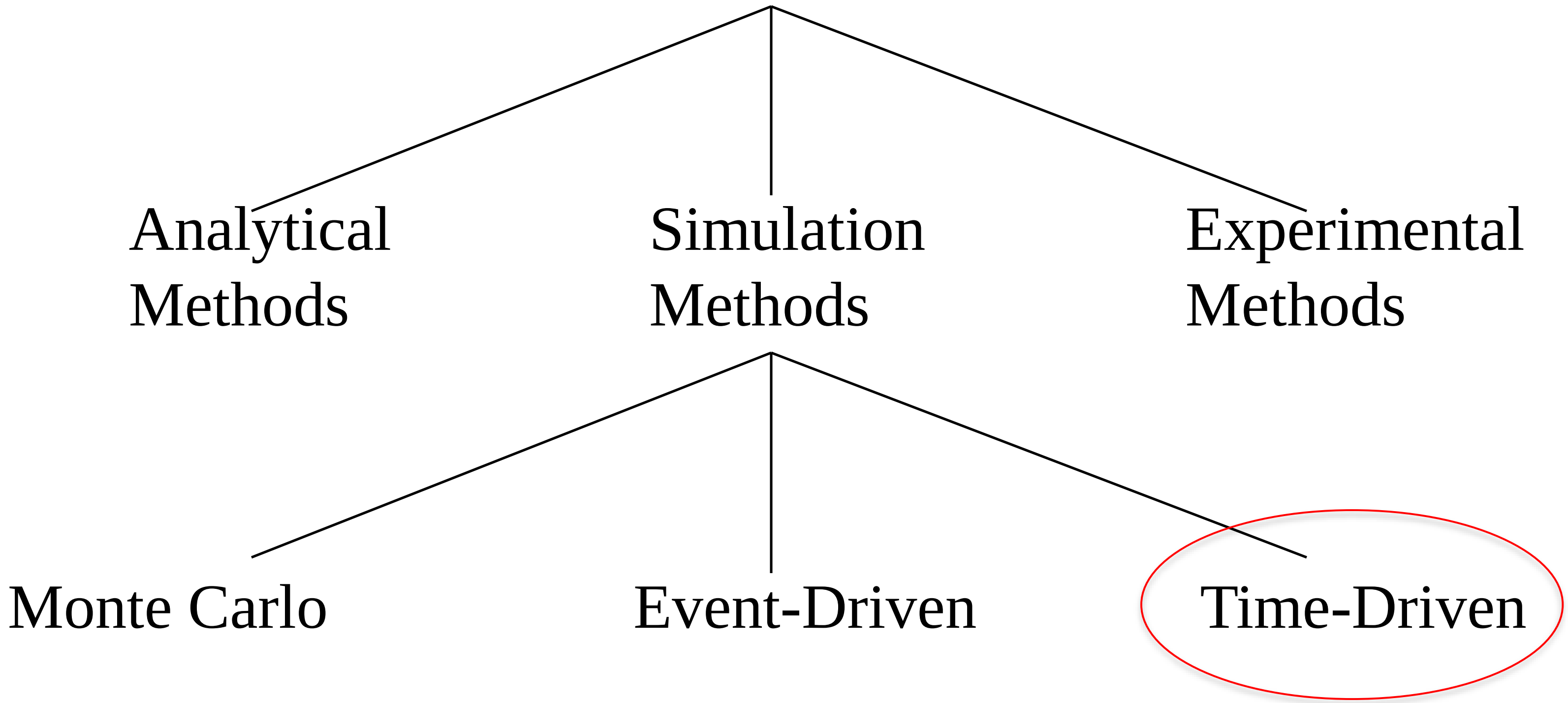
Performance Evaluation

- Analytical Methods
- Simulation Methods
  - Monte Carlo
  - Event-Driven
  - Time-Driven
- Experimental Methods

Monte Carlo Simulation

- Estimating an answer to some difficult problem using probabilistic approaches, based on (lots of) random numbers
- Examples: numerical integration, primality testing, WSN coverage
- Suited to stochastic problems in which probabilistic answers are acceptable
- Might be one-sided answers (primality)
- Can bound probability to some $\epsilon \ll 1$

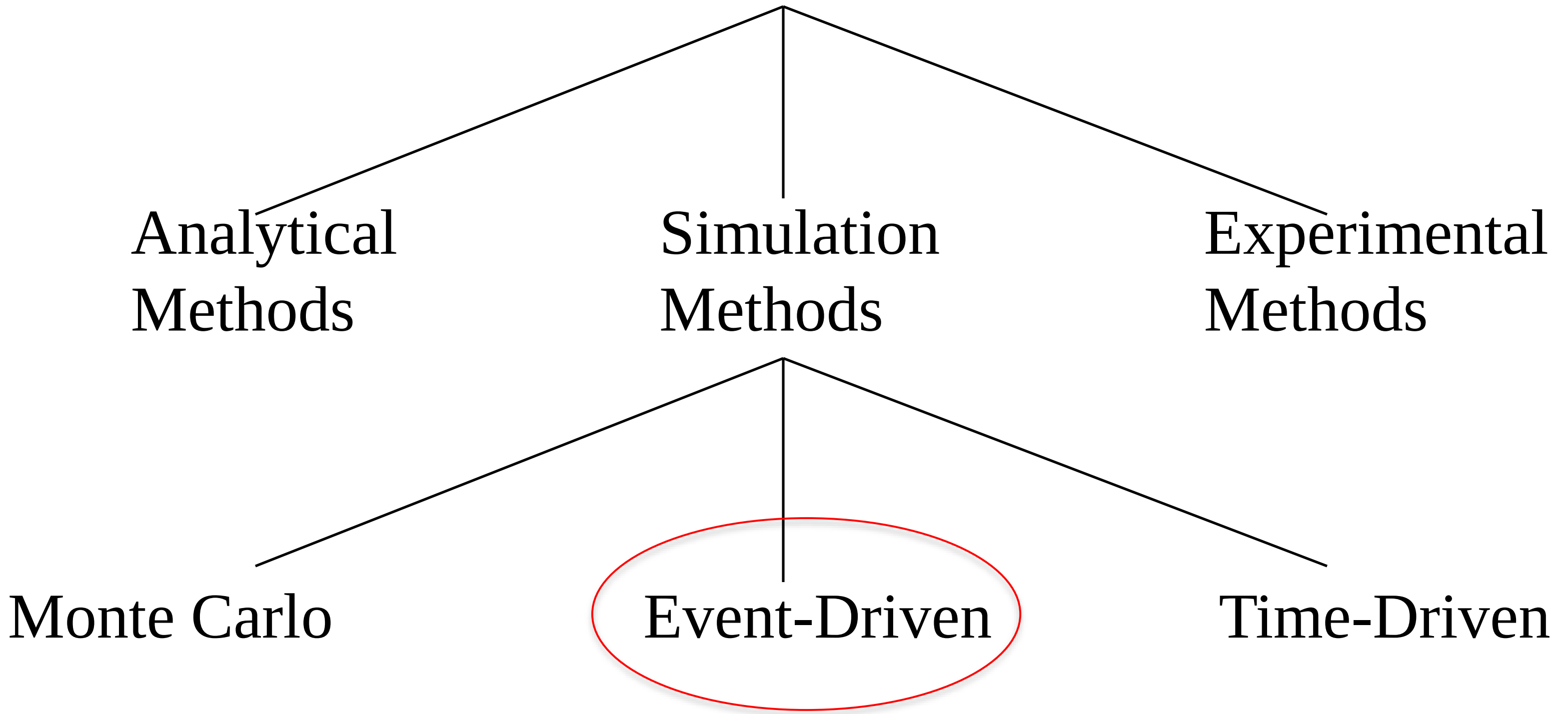
Performance Evaluation



Time-Driven Simulation

- Time advances in fixed size steps
- Time step = smallest unit in model
- Check each entity to see if state changes
- Well-suited to continuous systems
  - e.g., river flow, factory floor automation
- Granularity issue:
  - Too small: slow execution for model
  - Too large: miss important state changes

Performance Evaluation



UNIVERSITY OF CALGARY

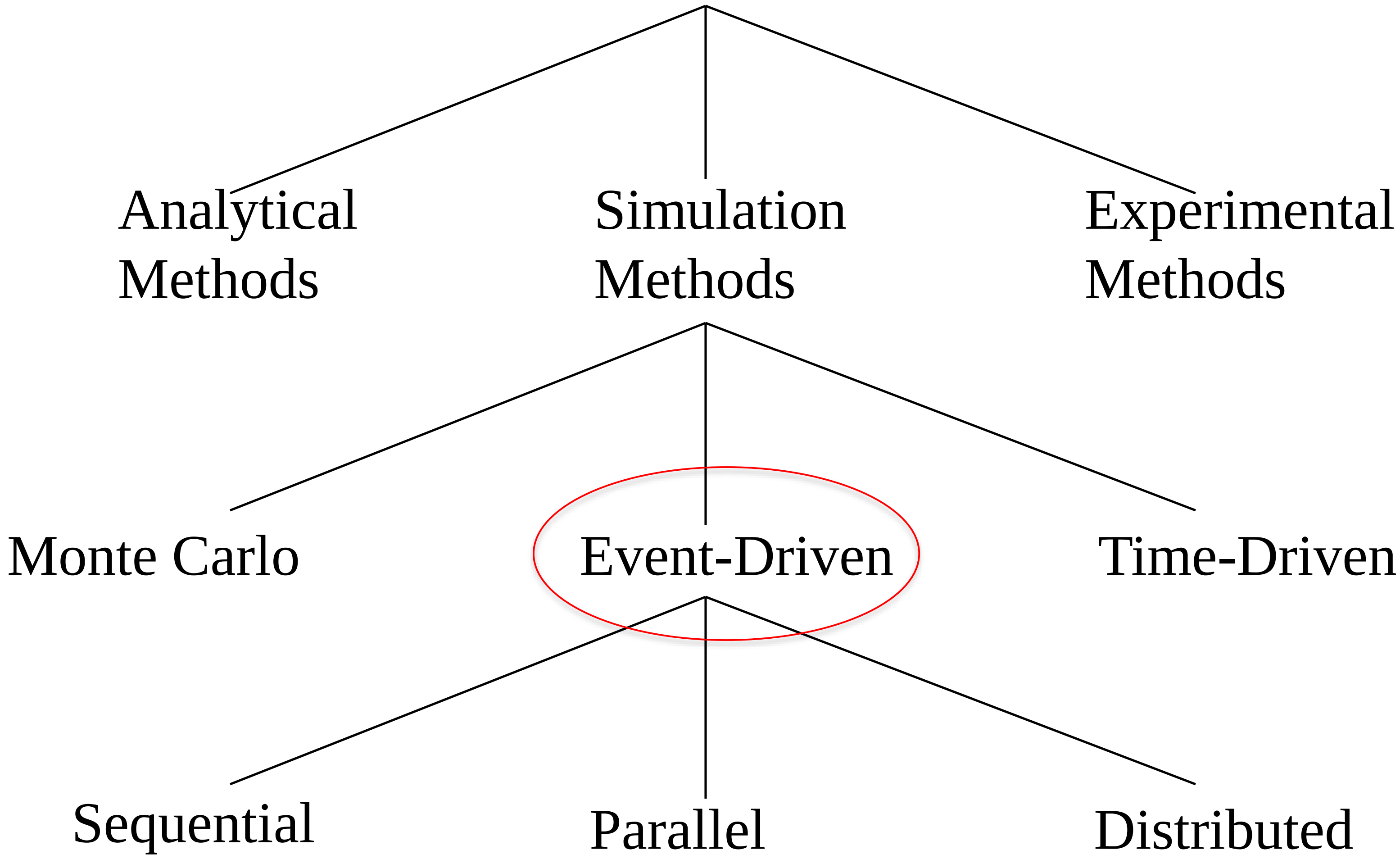
Event-Driven Simulation (1 of 2)

- Discrete-event simulation (DES)
- System is modeled as a set of entities that affect each other via events (messages)
- Each entity can have a set of states
- Events happen at specific points in time (continuous or discrete), and trigger discrete state changes in the system
- Very general technique, well-suited to modeling discrete systems (e.g., queues)

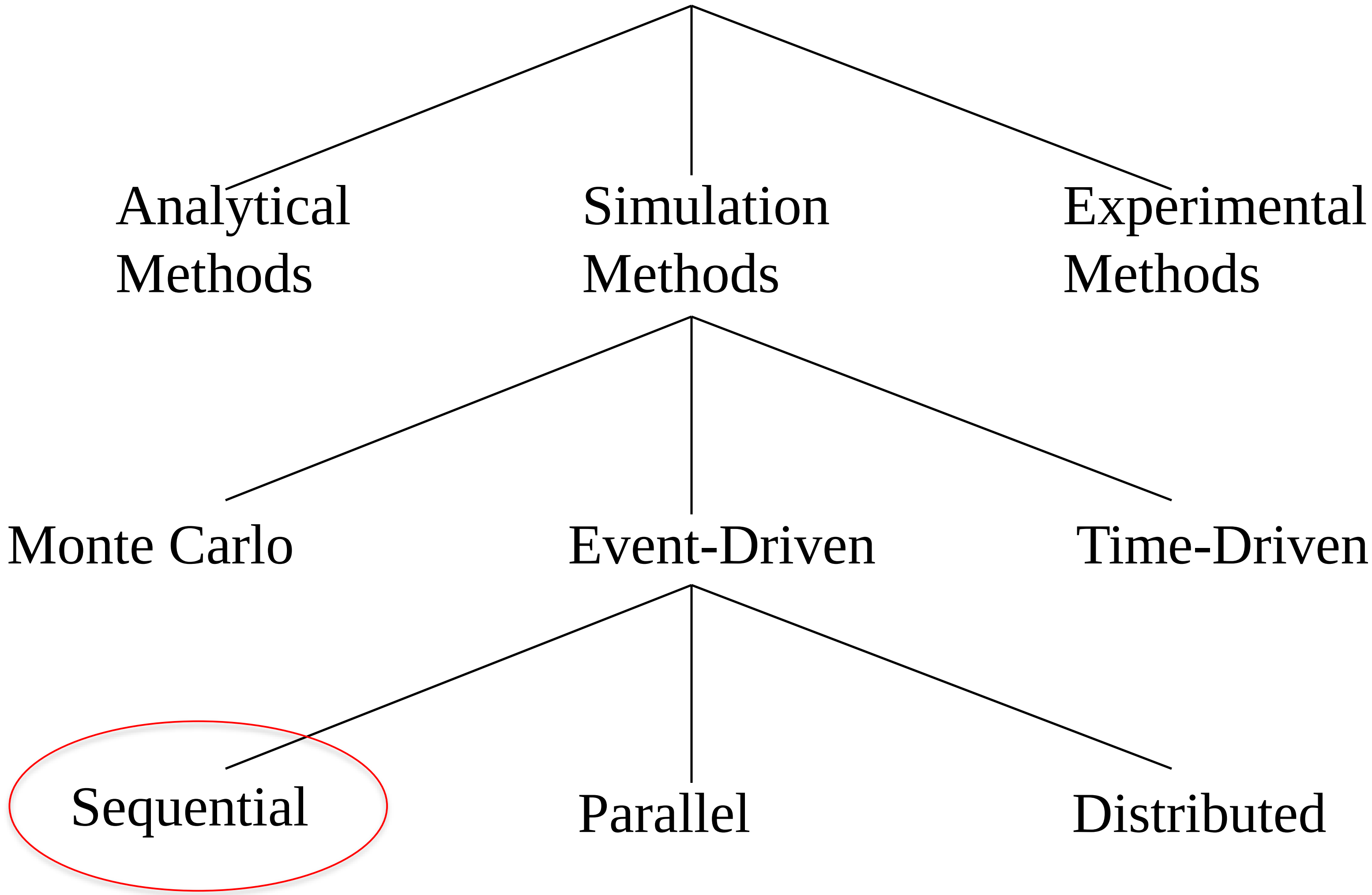
Event-Driven Simulation (2 of 2)

- Typical implementation involves an event list, ordered by time
- Process events in (non-decreasing) timestamp order, with seed event at t=0
- Each event can trigger 0 or more events
  - Zero: “dead end” event
  - One: “sustaining” event
  - More than one: “triggering” event
- Simulation ends when event list is null, or desired time duration has elapsed

Performance Evaluation



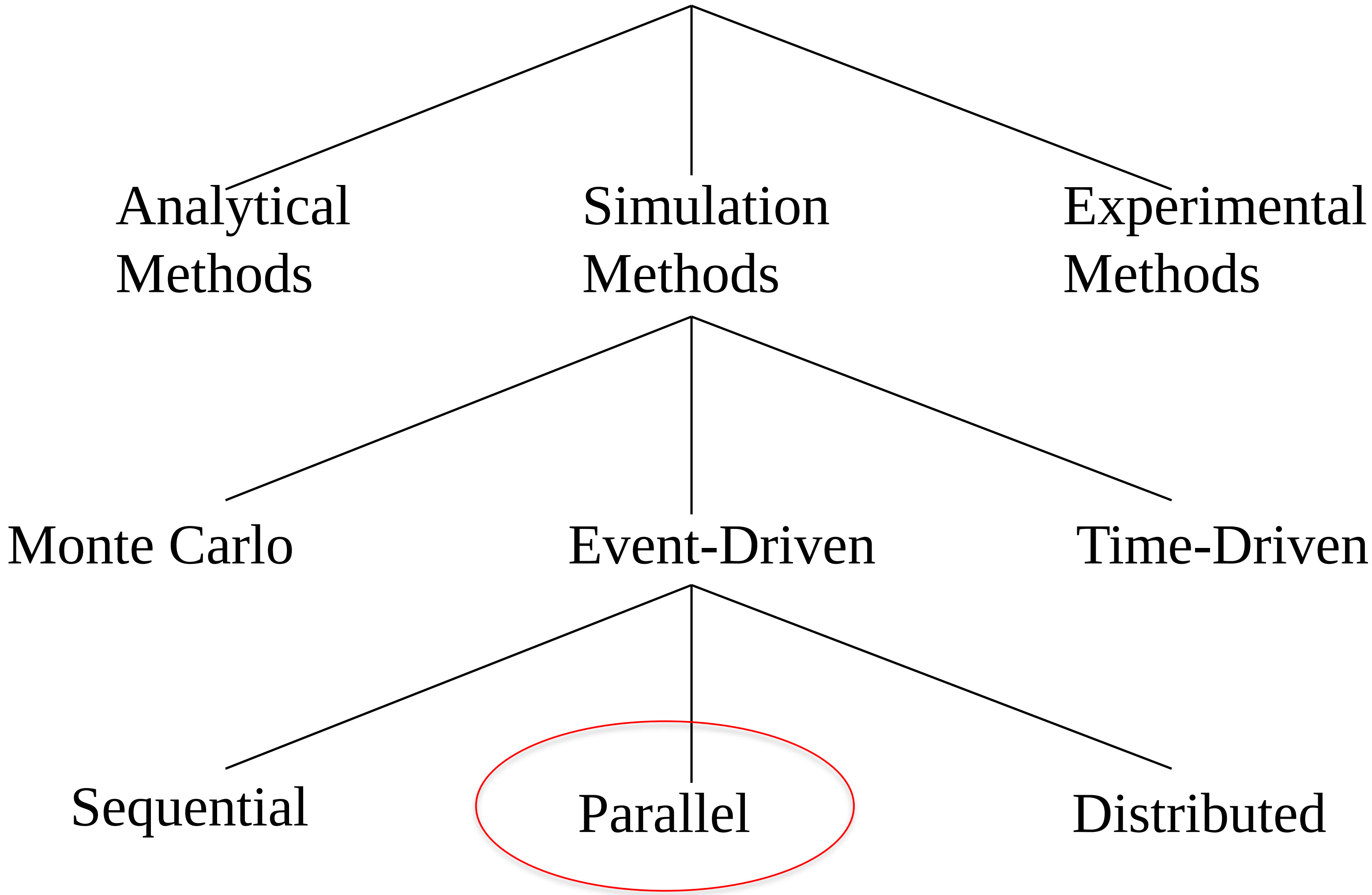
Performance Evaluation



Sequential Simulation

- Assumes a single processor system
- Uses central event list (ordered by time)
- Global state information available
- Single, well-defined notion of time
- Many clever implementation techniques and data structures for optimizing event list management
  - Linked list; doubly-linked list; priority queue; heap; calendar queue; trie structure

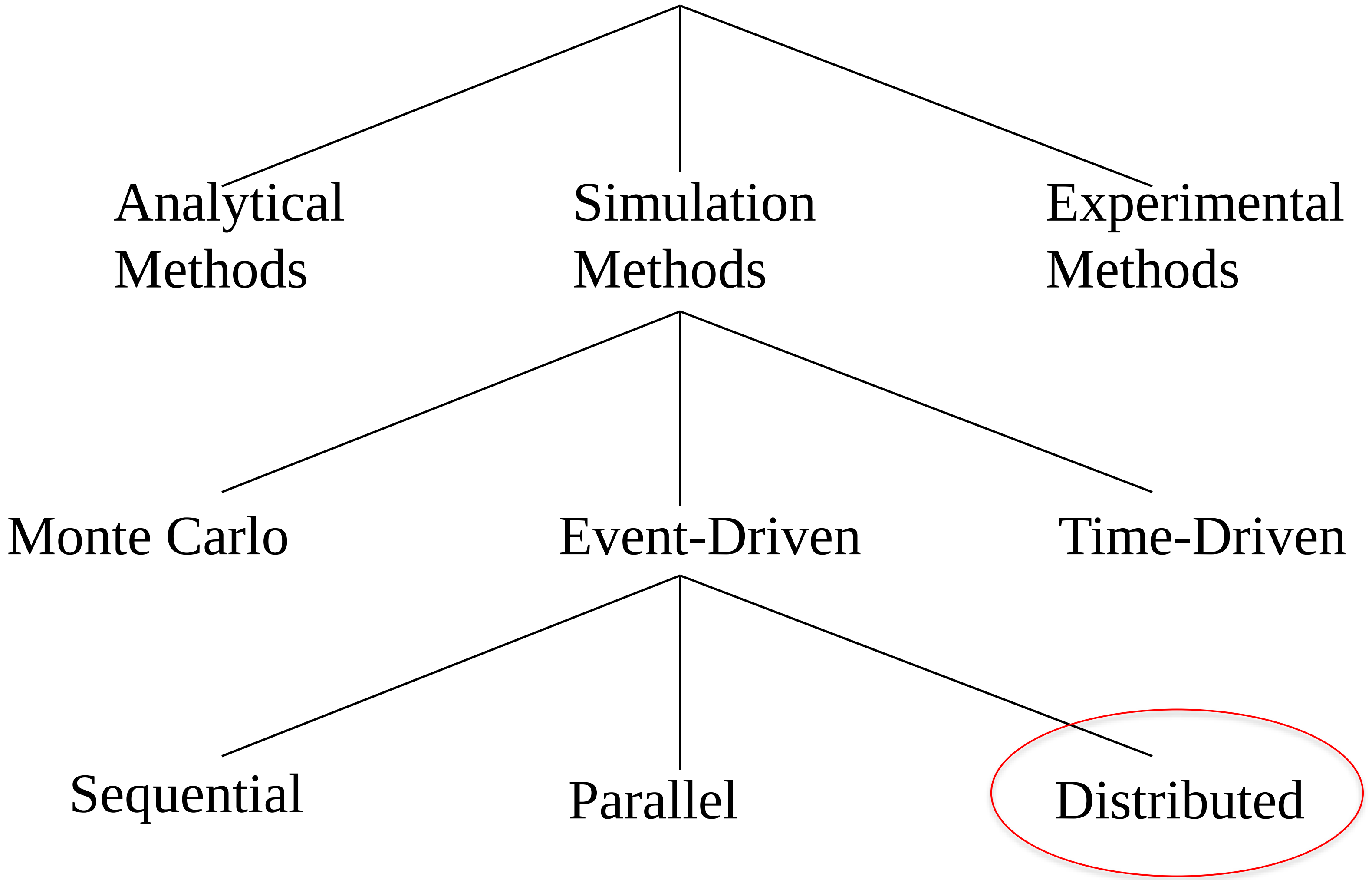
Performance Evaluation



Parallel Simulation

- Assumes multiple processors or cores, often tightly coupled, with shared memory
- Same sim results as sequential, only faster
- Need fast inter-process communication
- Shared state vs. no shared state
- Event list: centralized or not?
  - Central event list can be a bottleneck
  - Decentralized requires careful coordination
- Potentially different views of time
- Conservative versus optimistic execution

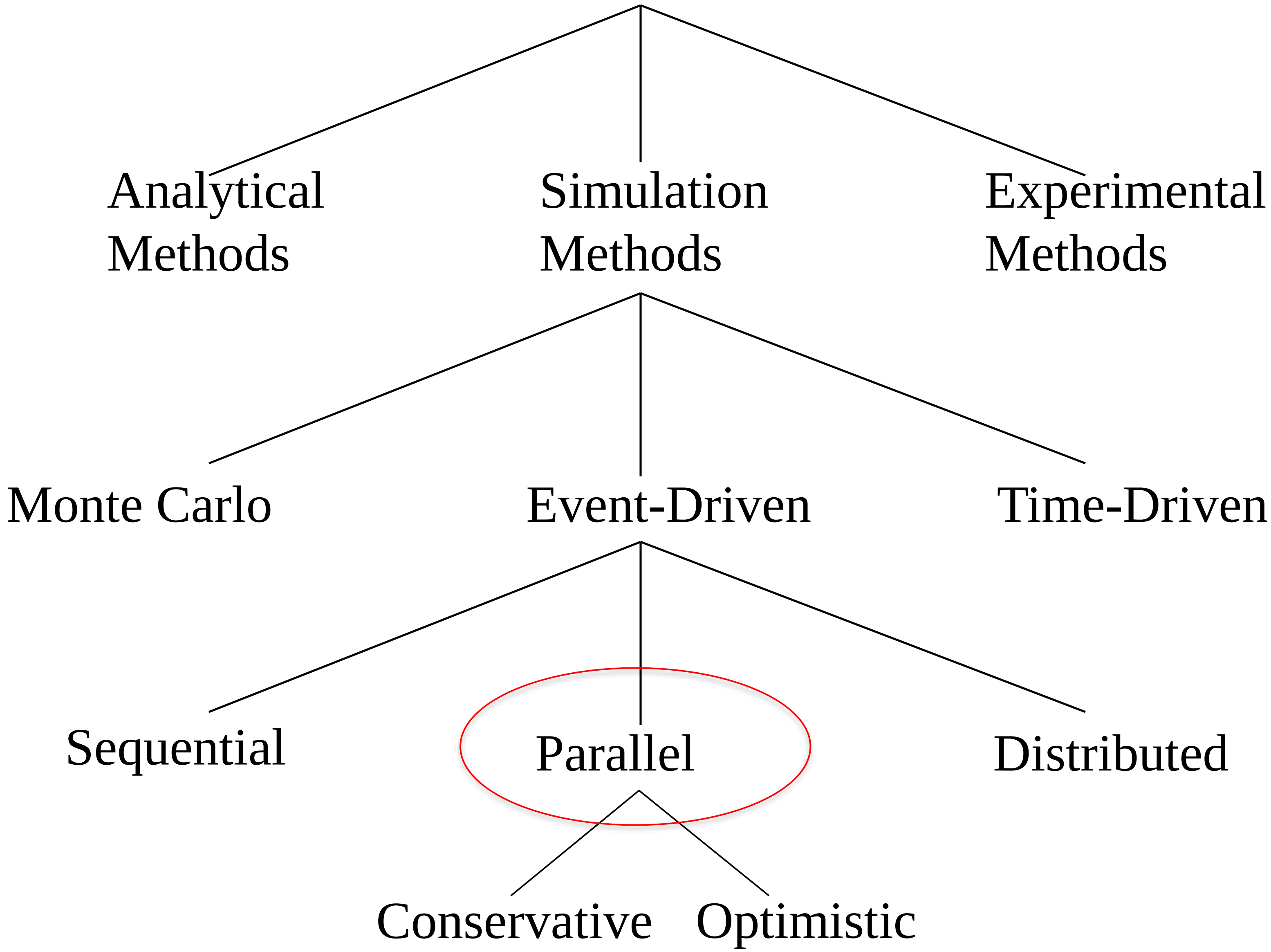
Performance Evaluation



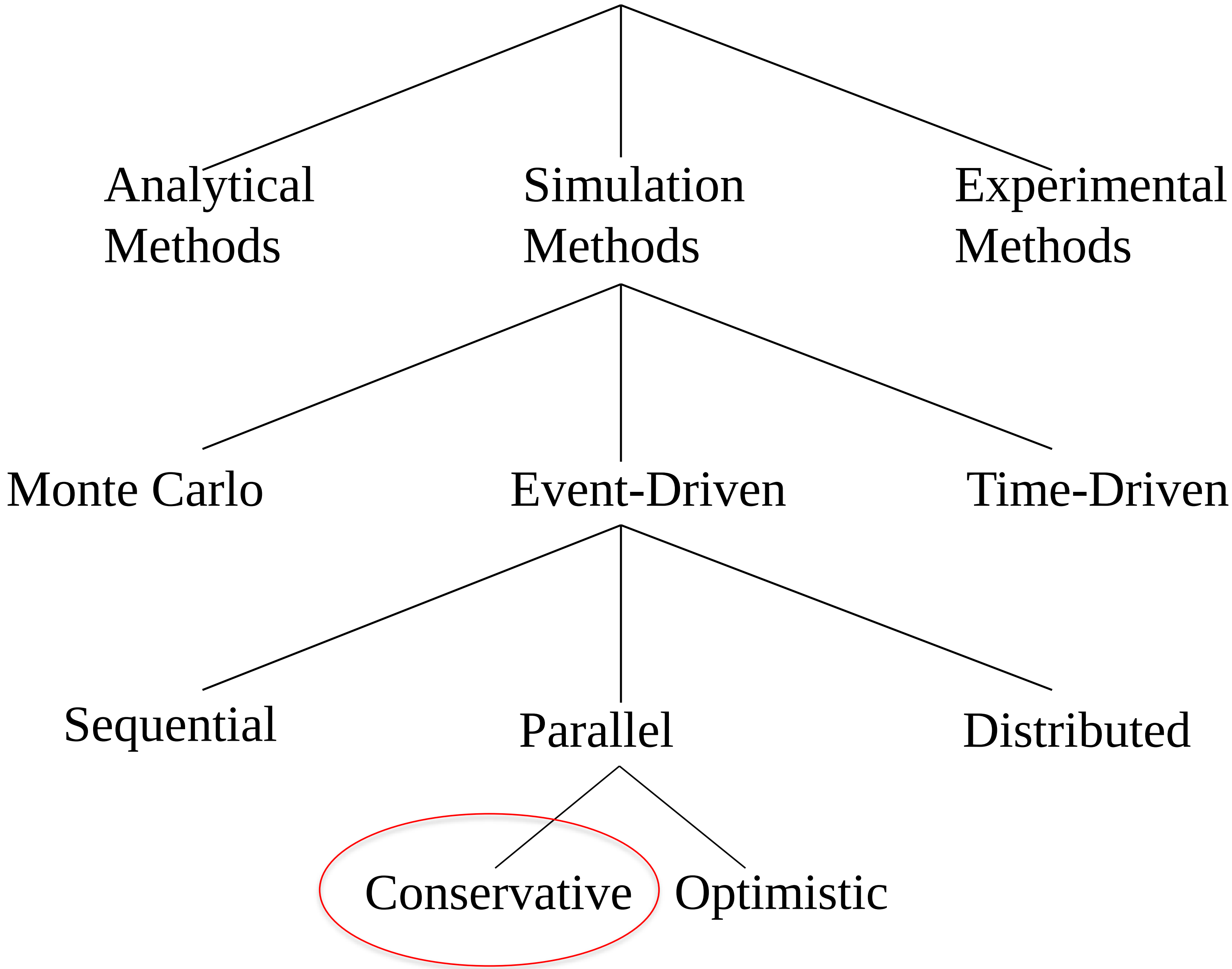
Distributed Simulation

- Assumes multiple processors, but geographically distributed (LAN/WAN)
- Inter-process communication becomes expensive because of large latencies
- Need to find right balance between computation and communication
- Granularity of task scheduling
- Similar technical issues to parallel simulation with respect to concurrency

Performance Evaluation



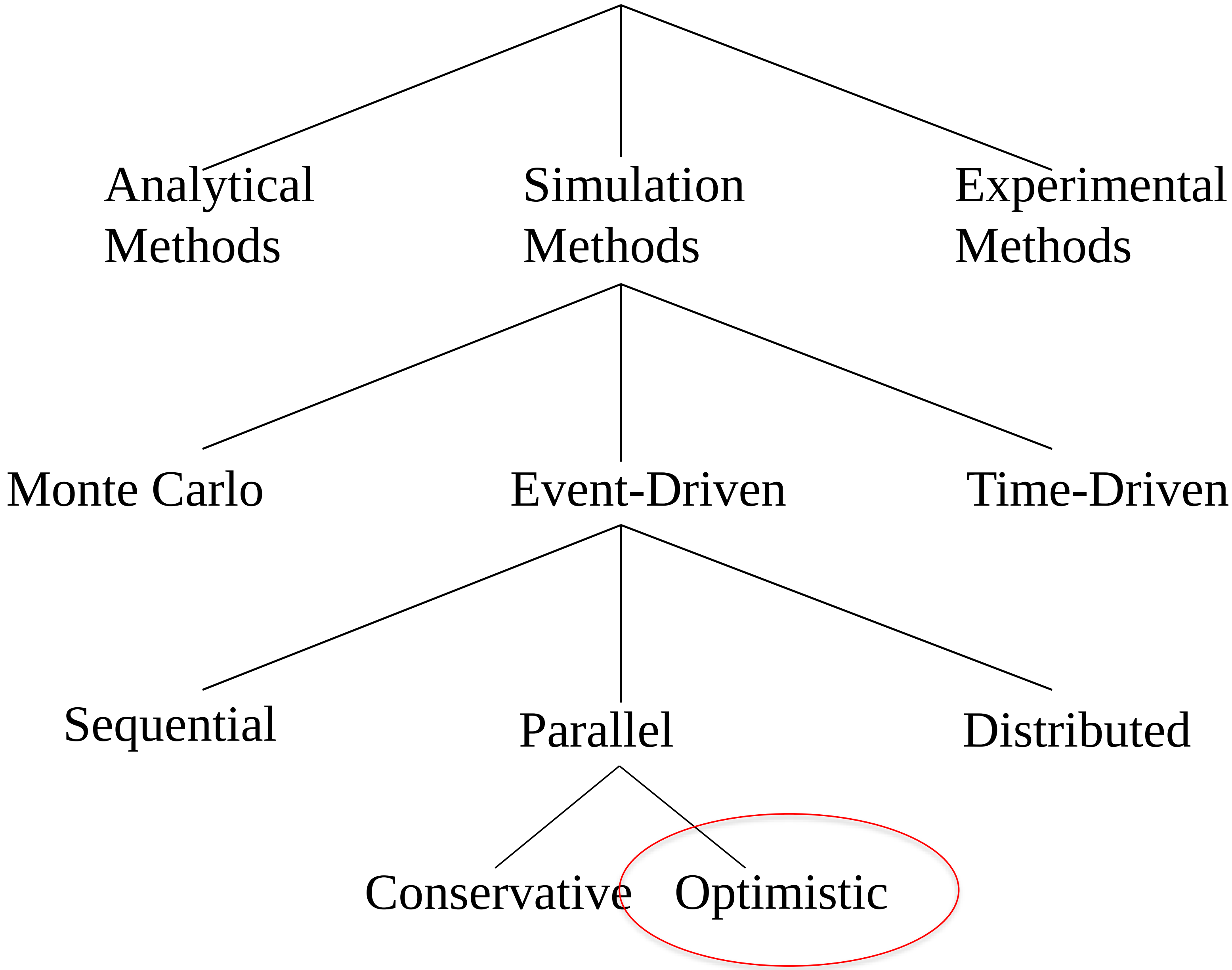
Performance Evaluation



Conservative Parallel Simulation

- Simulation tasks are divided into Logical Processes (LPs)
- LPs are mapped onto physical processors or cores for execution
- Sim events = messages between LPs
- LPs are carefully coordinated to track the Global Virtual Time (GVT), and only execute events when safe to do so
- Advantage: Always correct execution
- Disadvantage: Can sometimes be slow

Performance Evaluation



Optimistic Parallel Simulation

- Simulation tasks are divided into Logical Processes (LPs)
- LPs are mapped onto physical processors or cores for execution
- Sim events = messages between LPs
- LPs execute events based on local VT, moving forward in time at their own pace, but might sometimes get messages from the past
- Advantage: Often much faster execution
- Disadvantage: Occasional rollbacks required to restore correct simulation state

Parallel Simulation: Research Issues

- Correct synchronization (Chandy-Misra)
- Achieving effective speedups (lookahead)
- Granularity of simulation models (Nicol)
- Simulation languages/environments (UCLA)
- Event list management (scalability issues)
- Partitioning into LPs (i.e., load balancing, locality, minimize inter-LP communication)
- Global Virtual Time (GVT) algorithm
- TimeWarp (Jefferson)
- State-saving approaches (Gomes/Unger)
- Hybrid models (Kiddle/Simmonds/Unger)
- HLA: High-Level Architecture (Fujimoto)

Summary

- Simulation methods offer a range of general-purpose approaches for performance evaluation
- Simulation modeler must determine the appropriate aspects of system to model
- "The hardest part about simulation is deciding what not to model." - M. Lavigne
- Many technical issues: RNG, validation, statistical inference, comp efficiency