The Answer to the Ultimate Question of Distributed Systems

An Overview of Application Programming/Composition, Scheduling, Execution, and Performance Evaluation Models

Adel Nadjaran Toosi

Outline

- Execution models
- Programming models
- Scheduling
- Platform
- Objectives
- Evaluation Methods
- Constraints (Parameters)
- We conclude with an abstract!!

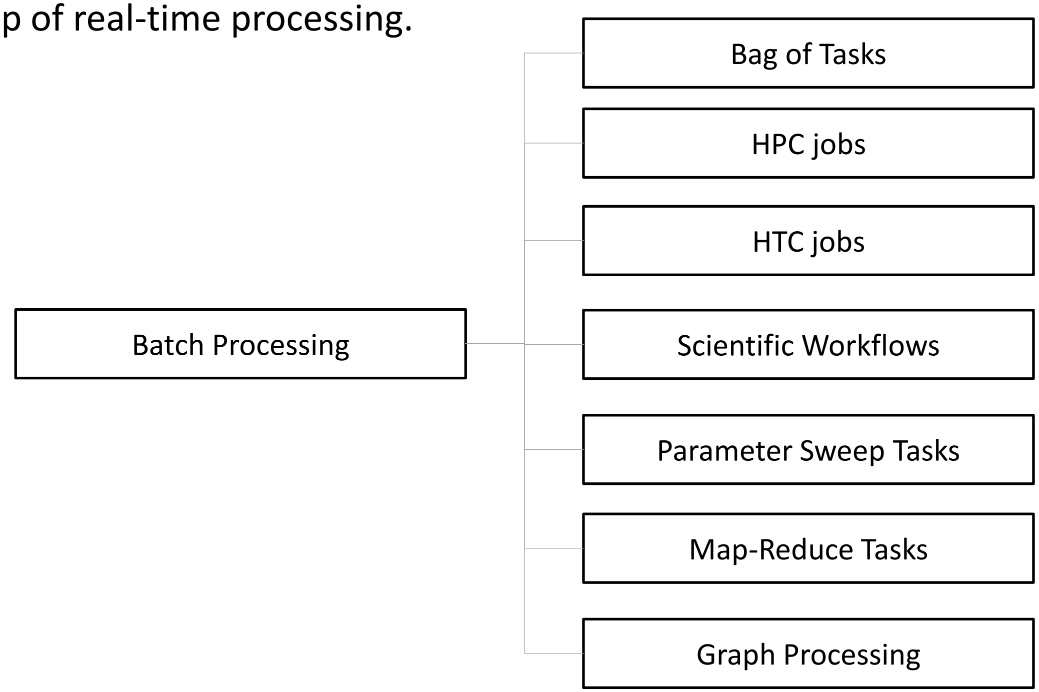
Execution Models In Distributed Systems

- Execution Models
  - Batch Processing
  - Interactive Processing (Online Processing)
  - Stream Processing
  - Real-time Processing
  - Parallel Processing

Adel Nadjaran Toosi

Batch Processing

- **Batch Processing**: allows users to submit series of programs (jobs) and they will be executed to completion without further user input and manual intervention.
  - Is Hadoop a batch processing framework?
    - In better words, Hadoop is an open source distributed processing framework.
    - Hadoop Map-Reduce is best suited for batch processing.
  - Spark and Storm can be used for real time and stream processing.
  - Strom is Hadoop of real-time processing.



Interactive (Online) Processing

- **Interactive Processing:** Interactive computing refers to application which accepts input from humans, e.g., Web applications, Massively Multiplayer Online (MMO) games.
- **Online Processing:** Another term for Interactive processing.
  - Interactive or online processing requires a user to supply an input.
  - Bar code scanning, online analytical processing (OLAP), online transaction processing (OLTP)

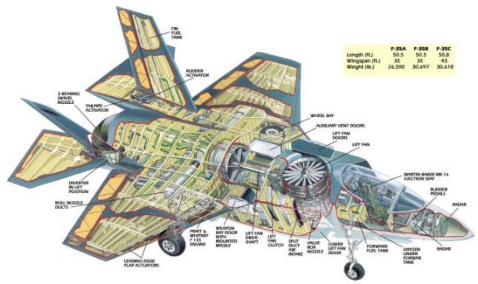
Adel Nadjaran Toosi

Stream Processing

- **Stream Processing:** record-by-record analysis of machine data in motion, e.g., Sensor Networks analytics, Internet of things applications, Online video processing.
- Characteristics
  - Compute Intensity
  - Data Parallelism
  - Data Locality
- Spark is a batch processing system at heart, but Spark Streaming is a stream processing system.

Real-time Processing

- **Real-time Processing:** real time data processing involves a continual input, process and output of data. Data must be processed in a small time period (or near real time).
  - Hard real-time
    - Nuclear systems, avionics
  - Firm real time
    - Sound system
  - Soft real-time
    - Weather stations



- **Real-time Processing vs. Stream Processing:** There are *no compulsory time limitations* in stream processing while small *guaranteed deadline* is compulsory in real-time processing
  - **Storm** is a stream or real-time processing system?
  - Other examples: airline ticket reservations, stock market, Fly-by-wire, antilock brakes, Videoconference applications, VoIP

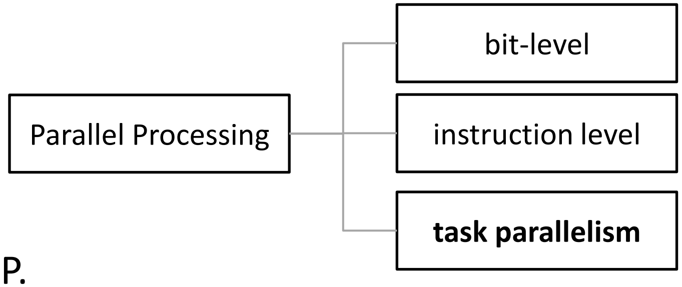
Adel Nadjaran Toosi

Parallel Processing

- **Parallel Processing:** is the processing of program instructions by dividing them among multiple processors with the objective of running a program in less time.
  - Concurrent computing vs. Parallel Processing
    - It is possible to have parallelism without concurrency (such as bit-level parallelism)
    - Concurrent and parallel programming are different. for instance, you can have two threads (or processes) executing concurrently on the same core through context switching. When the two threads (or processes) are executed on two different cores (or processors), you have parallelism.
  - **Speed-up and Amdahl's law :** If α is the fraction of running time a program spends on non-parallelizable parts, then the maximum speed-up with parallelization of the program is ,

$$\lim_{p \to \infty} \frac{1}{\frac{1-\alpha}{p}+\alpha} = \frac{1}{\alpha}$$

where P being the number of processors used.



  - Examples: Parallel programs in MPI and OpenMP.

Adel Nadjaran Toosi

Other types

- Data Warehouse
- Transaction Processing
- ?

Application Programming Models In Distributed Systems

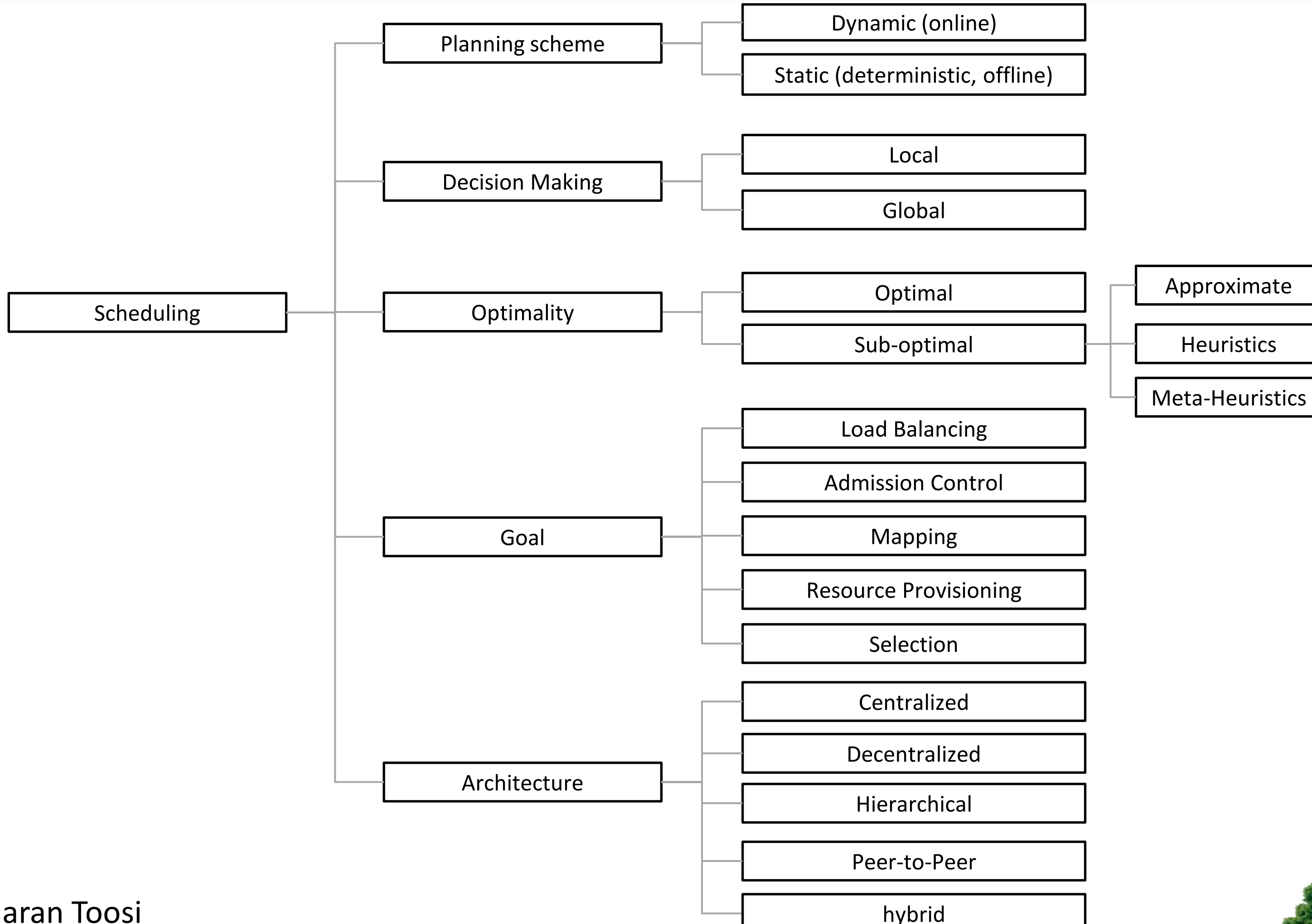
- Programming Models
  - Thread
  - Task
  - Map-Reduce
  - Message Passing
  - Data Flow
  - Workflow
  - Parameter Sweep
  - Bag of Tasks

Scheduling

- Scheduling is the process of **arranging**, **controlling** and **optimizing** work and workloads by assigning them to **resources**.
- Three main components of any scheduling problem:
  – *Consumer*, e.g. processes, threads, cloud clients.
  – *Resource*, e.g., CPU, I/O, VMs
  – *Policy*
- Allocation vs. Scheduling!!!
  – Often implicit distinction between the terms in the literature, but, in general, it can be said that:
    - Allocation is from resources' point of view, while Scheduling is form consumers' point of view.

Adel Nadjaran Toosi

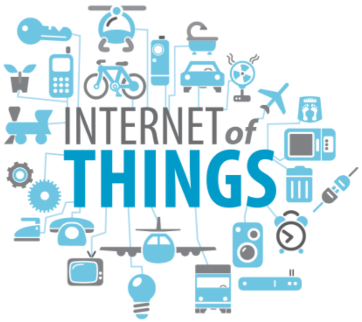
Scheduling



Adel Nadjaran Toosi

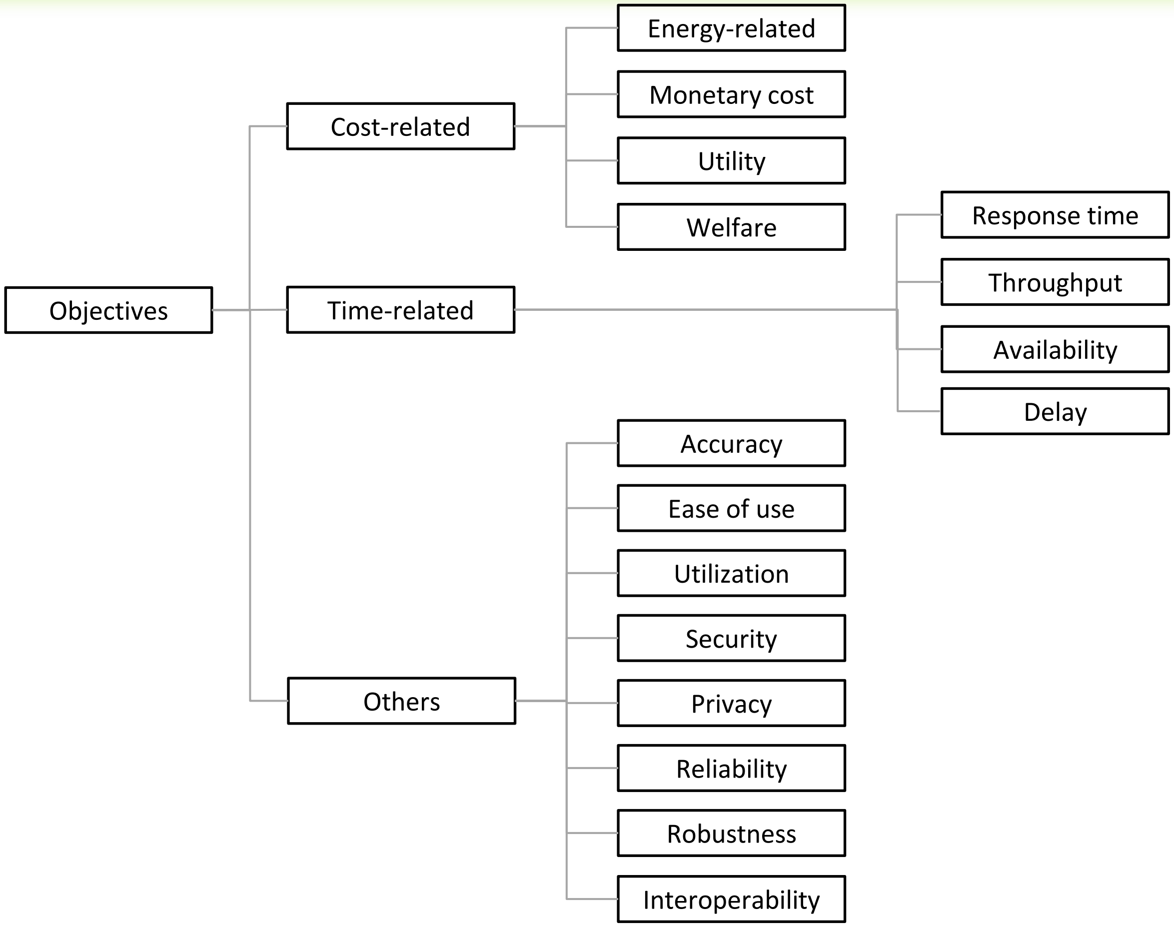
Platforms

- Cluster
- Grid
- Cloud
- Peer-to-Peer Systems
- Super Computers
- Mobile Computing
- Sensor Networks
- Internet of things
- Content delivery networks (CDN)
- Software Defined Networks (SDN)
- …



Objectives

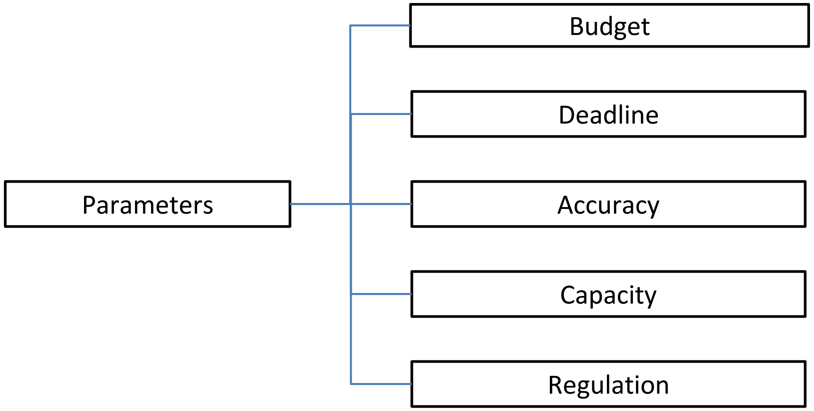


Objectives
Cost-related
Energy-related
Monetary cost
Utility
Welfare
Time-related
Response time
Throughput
Availability
Delay
Others
Accuracy
Ease of use
Utilization
Security
Privacy
Reliability
Robustness
Interoperability

Evaluation Methods

- Evaluation Methods
  - Analytical Modeling
  - Measurement
  - Empirical Analysis
  - Emulation
  - Simulation

Constraints (Parameters)



How to write your abstract!!

- Problem
  - Short Background (If necessary)
  - Scope
    - Application model, e.g., Map-reduce
    - Platform, e.g., Cluster
  - Objective, e.g., Cost and Energy Consumption
  - Constraints, e.g., Capacity and Available Renewable Energy
- Methodology
  - E.g., Online scheduling using meta-heuristics
  - Evaluation Method
    - Analytical proofs, Simulation, Emulation, Real Implementation
- Results/Findings
  - Conclusion/Implications

Adel Nadjaran Toosi

BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data

Problem:

In this paper, we present BlinkDB, a massively parallel, approximate query engine for running interactive SQL queries on large volumes of data. BlinkDB allows users to trade-off query **accuracy** for **response time**, enabling **interactive queries** over massive data by running queries on data samples and presenting results annotated with meaningful error bars.

Methodology:

To achieve this, BlinkDB uses two key ideas: 1) an adaptive optimization framework that builds and maintains a set of multi-dimensional stratified samples from original data over time, and 2) a dynamic sample **selection strategy** that selects an appropriately sized sample based on a query's **accuracy** or **response time** requirements.

Evaluation:

We **evaluate** BlinkDB against the well-known TPC-H benchmarks and a real-world analytic workload derived from Conviva Inc., a company that manages video distribution over the Internet. Our experiments on a node **cluster** show

Results and Conclusions:

that BlinkDB can answer queries on upto 17TBs of data in less than seconds(over 200× faster than Hive),with in an error of 2-10%.

Adel Nadjaran Toosi

The answer is 41.

Any other questions?

Adel Nadjaran Toosi