INSTABASE

Distributed Workflow Processing

Radhika Patwari

Indian Institute of Technology Kharagpur (IIT KGP)

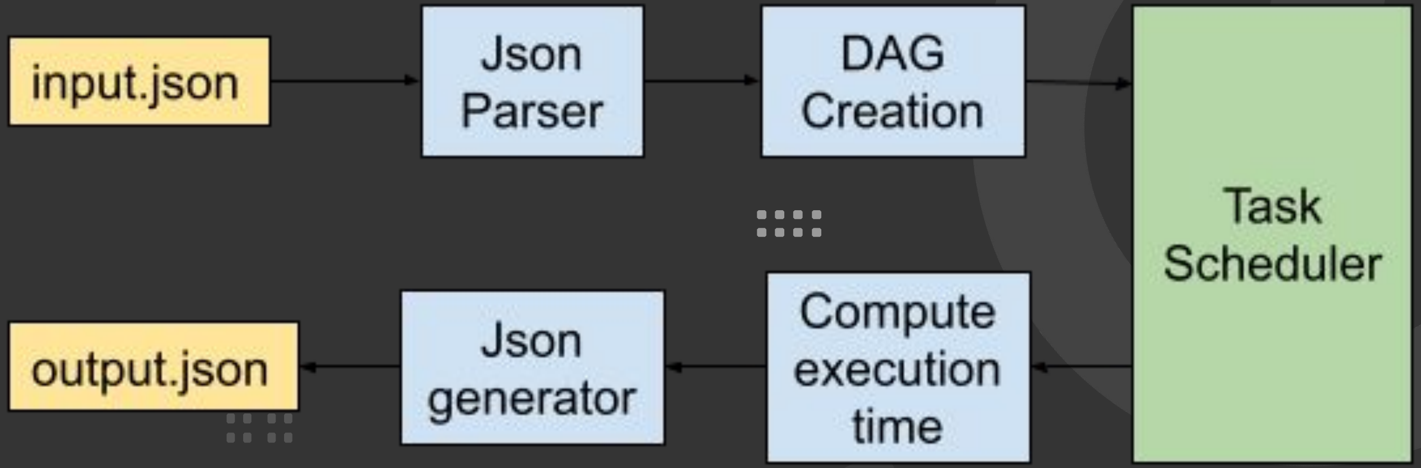
<Encipher/>

Coding the next generation of automation applications

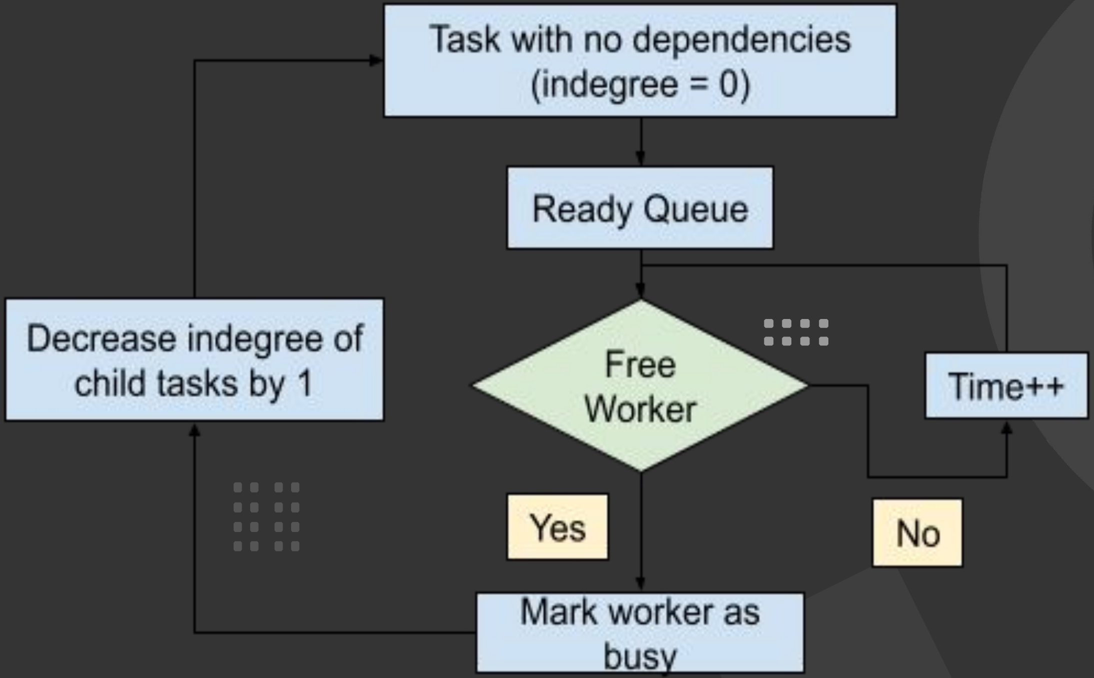
Solution Overview

- NP-complete problem
- Heuristic approach for tasks in ready queue
  - first-come first-serve basis
  - higher priority to low-cost tasks
  - ensuring directed acyclic graph
  - maintaining task dependencies using Kahn’s algorithm
  - minimising median of overall execution time
  - efficient in terms of space
  - minimising idle time of workers
  - idle time vs fairness trade-off

Orchestrator component - Design



Scheduling component - Design


Task with no dependencies (indegree = 0)
Ready Queue
Free Worker
Time++
Decrease indegree of child tasks by 1
Yes
No
Mark worker as busy

Parallel Execution

- Parallel tasks start simultaneously
- Available in ready queue
- Early coming task assigned to free worker
- Minimising median vs mean of execution time
- Different parallel tasks assigned to same/different workers
- Child tasks enter the ready queue once all dependent tasks executed

Task Dependency

- Kahn algorithm for topological sorting
- Adjacency list is used to generate dependency graph
- Task dependencies form a Directed Acyclic Graph
- No cycles/self-loops
- Indegree = 0 implies the task has no dependency left for execution
- Tasks with indegree = 0 reside in ready queue
- Ensuring task dependencies during every task-worker assignment

Fairness

- For overlapping tasks,
    - priority to first-come low-cost task
    - allotment of other task to a free worker / same worker after completion
- For non-overlapping tasks,
    - order always maintained
- Round Robin Scheme ?
    - Minimises mean of execution time , what about median ?
- Shortest Job First ?
    - Increase Idles time of workers, trade-off ?

Third party libraries / frameworks

- Jsoncpp
  - Reading/Writing json files in C++
- STL C++ library
  - Pre-implemented data structures
    - Priority queue
    - Set
    - Map
    - Vector
  - Algorithms