CONCURRENT AND PARALLEL PROGRAMMING

Dr. Emmanuel S. Pilli

MNIT Jaipur

Syllabus - CST 303

- Concurrent versus sequential programming.
- Concurrent programming constructs and race condition.
- Synchronisation primitives.
- Processes and threads. Interprocess communication.
- Livelock and deadlocks, starvation, and deadlock prevention.
- Issues and challenges in concurrent programming paradigm and current trends.

Syllabus - CST 303

- Parallel algorithms – sorting, ranking, searching, traversals, prefix sum etc.
- Parallel programming paradigms – Data parallel, Task parallel, Shared memory and message passing,
- Parallel Architectures, GPGPU, pthreads, STM, OpenMP
- OpenCL, Cilk++, Intel TBB, CUDA
- Heterogeneous Computing: C++AMP, OpenCL

References

- Principles of Concurrent and Distributed Programming by Ben-Ari (Prentice-Hall International)
- Concurrent Programming: Principles and Practice by Greg Andrews (Addison Wesley)
- Synchronization Algorithms and Concurrent Programming by Gadi Taubenfeld (Pearson)

References

- Introduction to Parallel Computing by Ananth Grama, et al (Pearson)
- Programming Massively Parallel Processors - A Hands-on Approach by David B. Kirk (Morgan Kaufmann)
- Parallel Algorithms by Joseph JaJa (Addison Wesley)
- CUDA Programming by Shane Cook (Morgan Kaufmann)
- Heterogeneous Computing with OpenCL by Benedict Gaster, et al (Morgan Kaufmann)

Approach

- Concurrent Programming (POSIX Threads)
- Synchronization Primitives
- Interprocess Communication
- Livelock and Deadlocks
- Parallel Programming Paradigms
  - Message Passing (MPI), Shared Memory (OpenMP)
- Parallel Architectures GPGPU
- Nvidia CUDA, AMD OpenCL, Cilk++, Intel TBB
- Parallel Algorithms
- Heterogeneous Computing

Weightage – Theory

- Mid Term 1 20
- Mid Term 2 20
- End Term 40
- Quizzes 10
- Assignments 10

Weightage – Practical

- Continuous Evaluation 30
- Mid Term 1 (Quiz / Test) 10
- Mid Term 2 (Quiz / Test) 10

Schedule

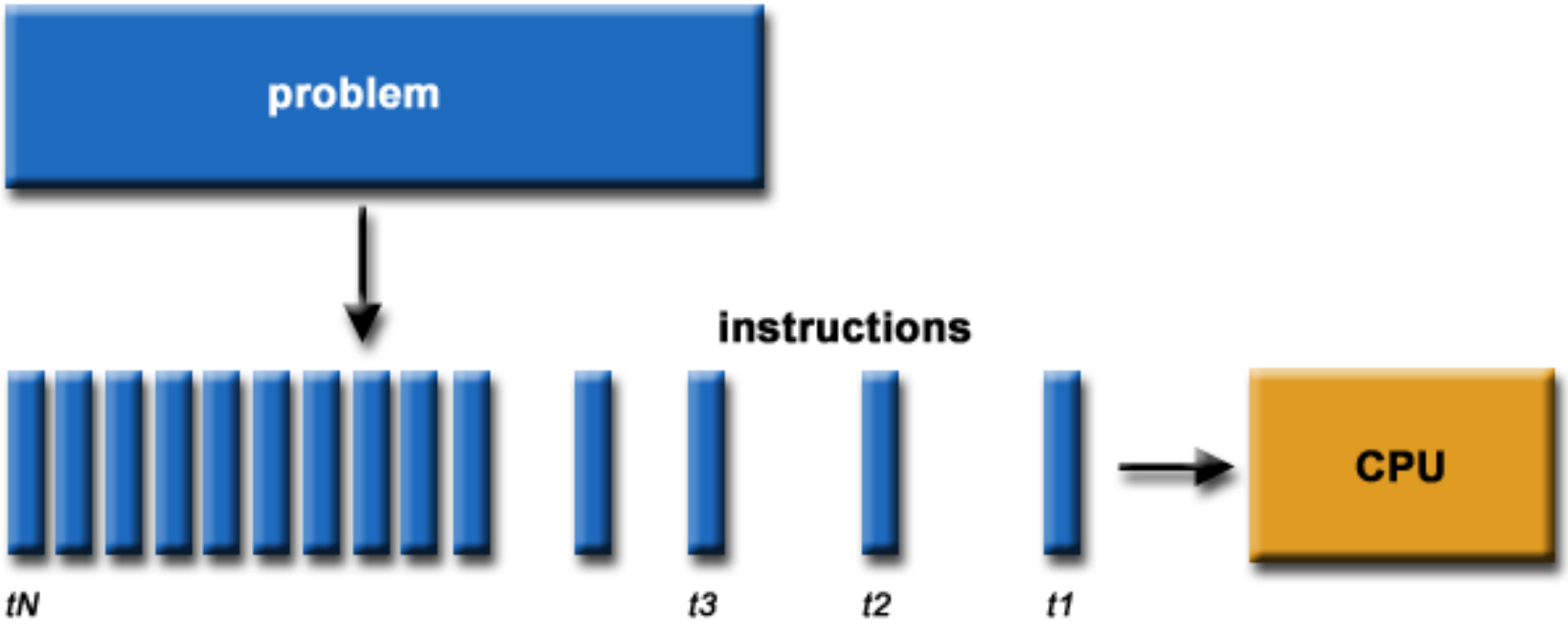
- Concurrent Programming
- Synchronisation Primitives
- Interprocess Communication
- Livelock and Deadlocks
- Parallel Algorithms
- Parallel Programming paradigms
- Parallel Architectures
- OpenMP, OpenCL, Cilk++, CUDA
- Heterogeneous Computing

Concurrent versus Parallel

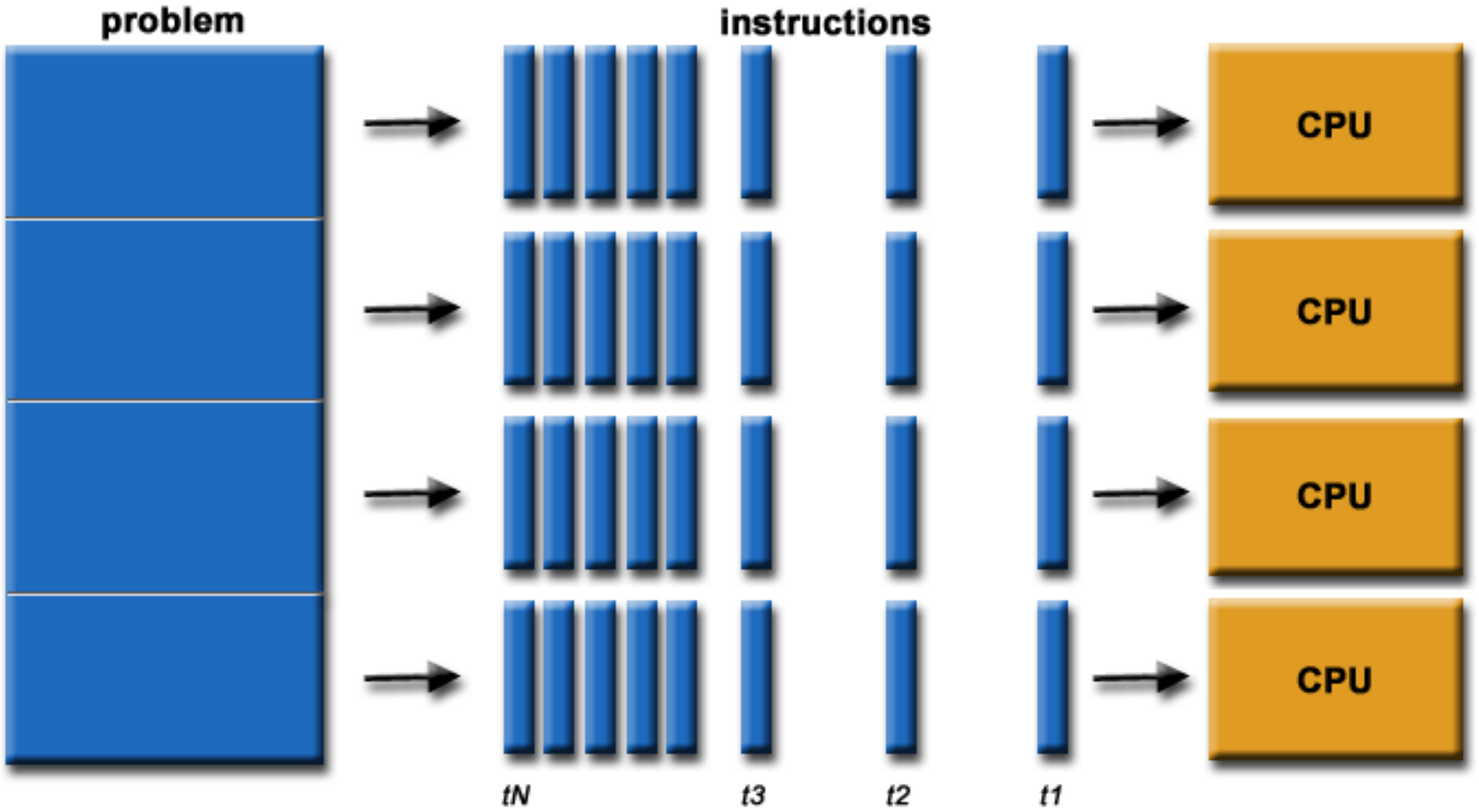
- **Concurrency** is when two tasks can start, run, and complete in overlapping time periods.
- **Parallelism** is when tasks literally run at the same time, eg. on a multicore processor.

- **Concurrency:** A condition that exists when at least two threads are making progress.
- **Parallelism:** A condition that arises when at least two threads are executing simultaneously.

Concurrent versus Parallel



Concurrent versus Parallel



Concurrent versus Parallel

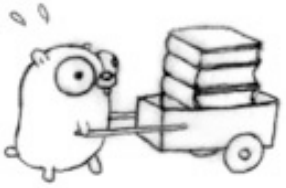
- **Concurrency:** When multiple tasks performed simultaneously with shared resources
- **Parallelism:** when single task divided into multiple simple independent tasks which can be performed simultaneously

- **Concurrency:** Programming as the composition of independently executing processes.
- **Parallelism:** Programming as the simultaneous execution of (possibly related) computations.

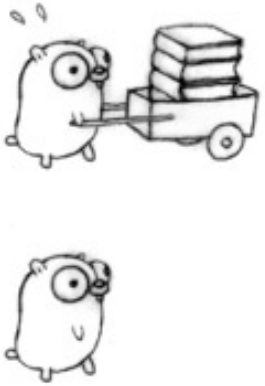
Concurrent versus Parallel

- Sequential
- Concurrent
- Parallel
- Concurrent But Not Parallel
- Parallel But Not Concurrent
- Concurrent and Parallel

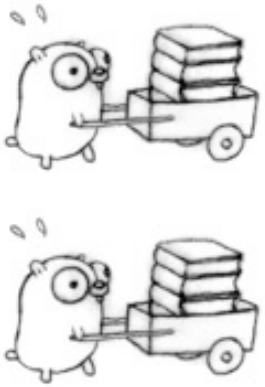
Problem !



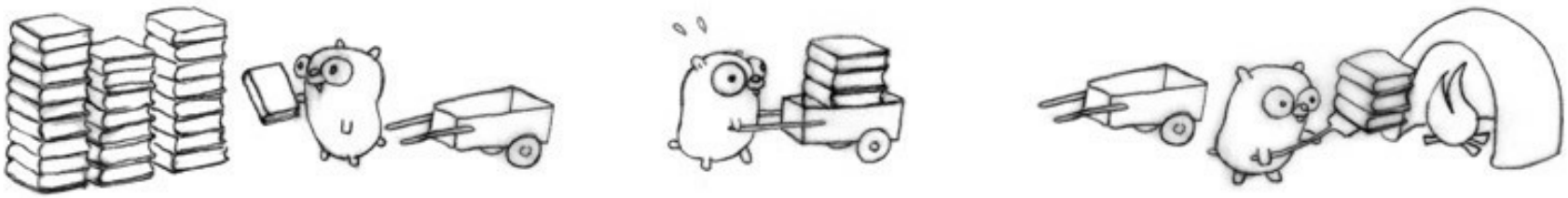
Solution ?



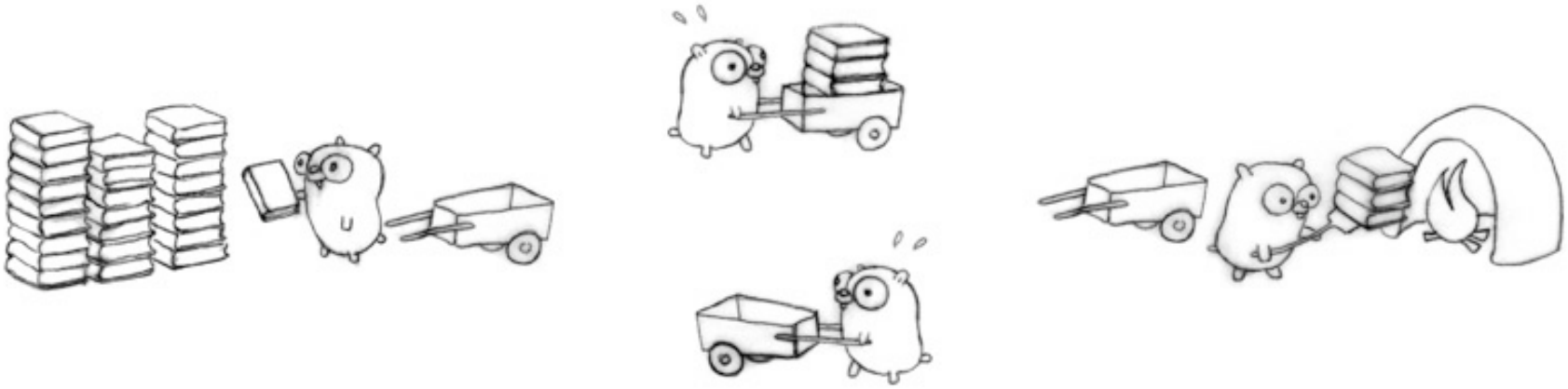
Parallel ?



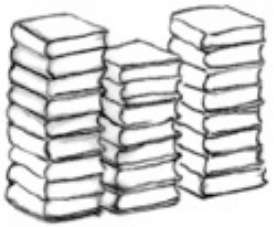
Concurrent ?

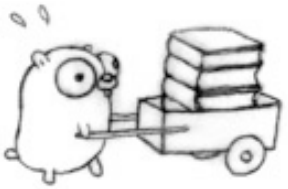


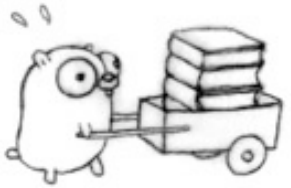
More Concurrent ?



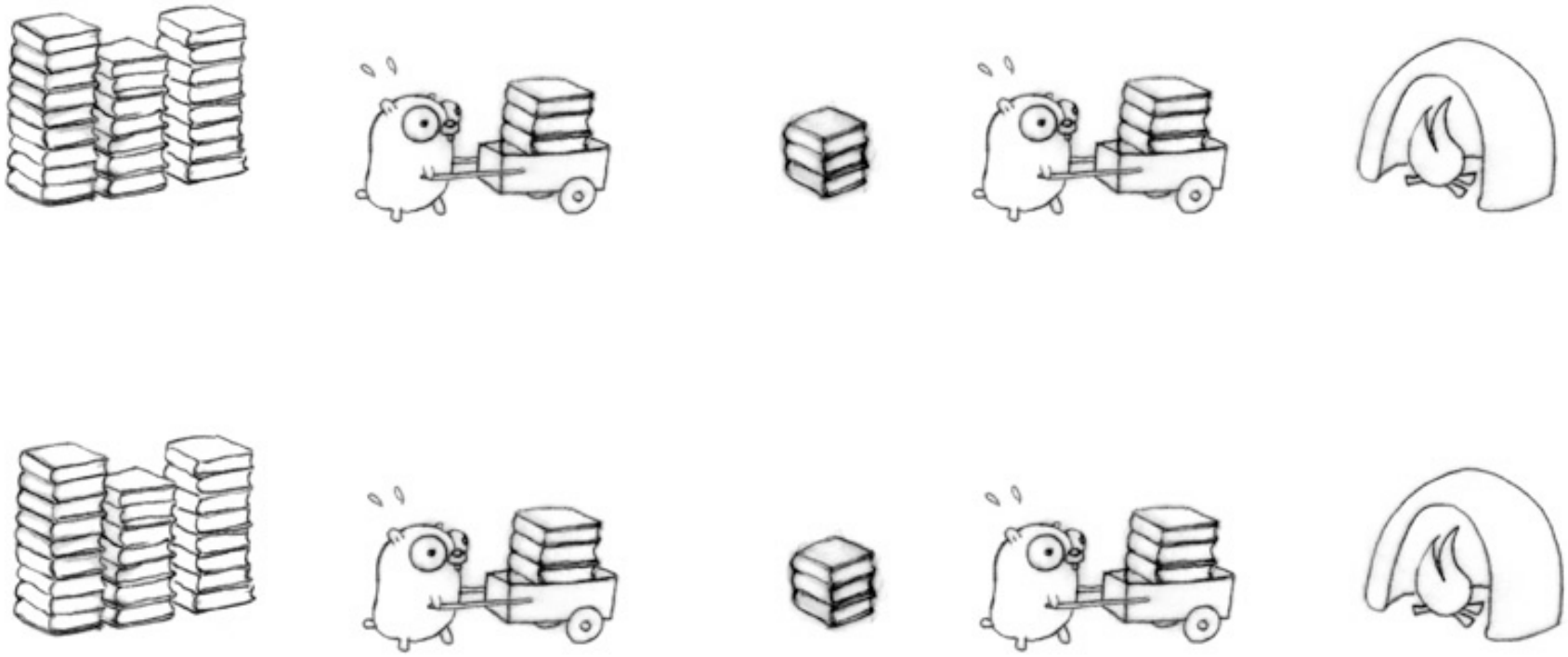
Parallel ?



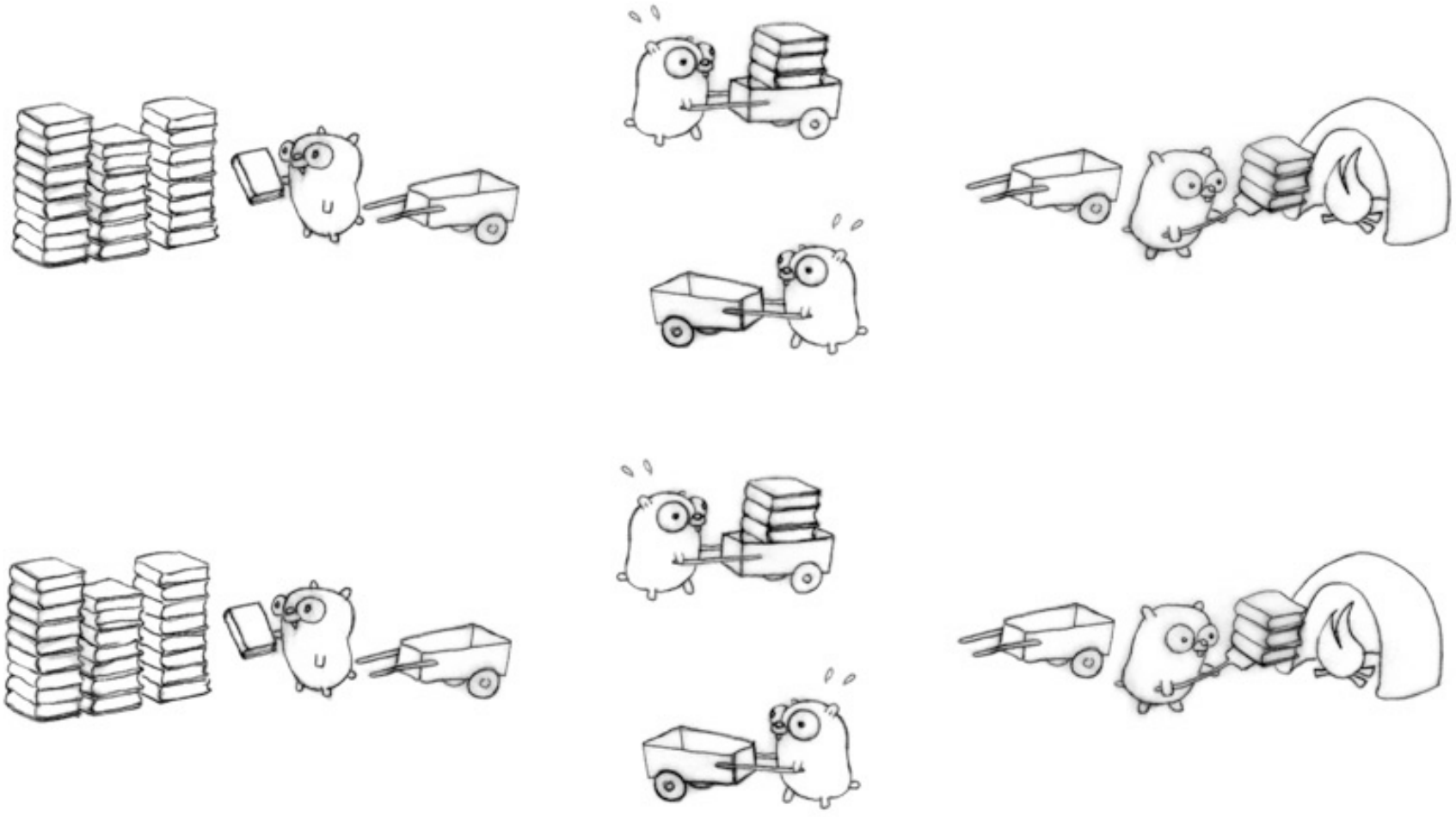




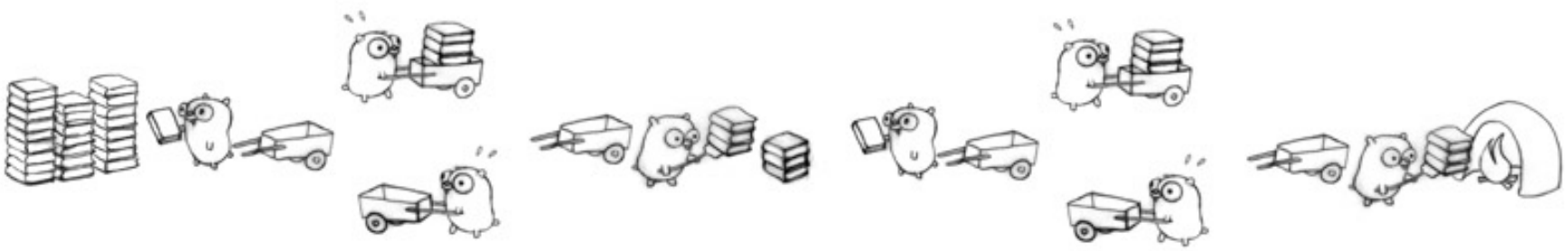
Concurrent or Parallel ?



Concurrent or Parallel ?



Multi Gopher + Staging Pile



Full Optimization