Pitfalls

Concurrent and Distributed Devices
(CDD101)

Race Conditions

- Two or more threads perform operations on the same location at the same time
- Sequential Consistency may not be guaranteed!
  - System may reorder operations!
- Task A
  - X=1
  - a=Y
- Task B
  - Y=1
  - b=X

Mutual Exclusions and Locks

- Locks are a last resort
- Use to protect logical invariants not memory locations

Deadlock

- Avoid Mutexes where possible
- Hold at most one lock at a time
  - Never call someone elses code while holding a lock unless you are sure they never acquire a lock
- Always acquire multiple locks in the same order
  - Stratify the mutexes
  - Sort mutexes
  - Backoff

Strangled Scaling

- Each Mutex is a potential bottleneck
  - More threads means more contention
- Fine grained locking helps
- Atomic Operations can also be helpful

Lack of Locality

- Temporal Locality
  - The core is likely to access the same location again in the near future
- Spatial Locality
  - The core is likely to access nearby locations in the near future
- One the cache is full use everything in it before using anything else!
- Try use Cache Oblivious algorithms

Load Imbalance

- Uneven distribution of work accross workers
- Over-decomposition
  - Divide the work into more tasks than there are workers

Overhead

- If tasks are too small then overhead per task is too large
- Watch arithmetic intensity!
  - Ratio of arithmetic opertations per memory access