Summary

• Background

• Introduction in algorithms and applications
  – Methodology to develop efficient parallel (distributed-memory) algorithms
  – Understand various forms of overhead (communication, load imbalance, search overhead, synchronization)
  – Understand various distributions (blockwise, cyclic)
  – Understand various load balancing strategies (static, dynamic master/worker model)
  – Understand correctness problems (e.g. message ordering)
Summary

- Parallel machines and architectures
  - Processor organizations, topologies (mesh, tree, hypercube), criteria
  - Types of parallel machines
    - arrays/vectors, shared-memory (bus-based, switch-based, NUMA), distributed memory
  - Routing
  - Flynn’s taxonomy

- What are cluster computers?
- What networks do real machines (like the Blue Gene) use?
- Which trends does the Top 500 show us?
- What is unique about DAS? How is it used? How are wide-area algorithms optimized?

- Speedup, efficiency (+ their implications), Amdahl’s law, weak and strong scaling
Summary

• Programming methods, languages, and environments
  – Different forms of message passing
    • naming, explicit/implicit receive, synchronous/asynchronous sending
  – Select statement
  – SR primitives (not syntax)

  – MPI: various message passing primitives, collective communication
    • Various communication modes; buffering

  – HPF: problems with automatic parallelization; division of work between programmer and HPF compiler; alignment/distribution primitives; performance implications
Summary

• Applications
  – Climate modelling
    • Community Earth System Model (CESM)
    • Different execution scenarios for CESM
  – N-body problems:
    • load balancing and communication (locality) optimizations, costzones, performance comparison of 4 different parallel algorithms
  – Search algorithm (TDS):
    • use asynchronous communication + clever (transposition-driven) scheduling
Summary

- Chapel:
  - Motivation and design principles
  - Partitioned Global Address Space (PGAS) model
  - Language core
    - Statements, expressions, procedures
    - Domains, tuples, records, classes
  - Data parallel features
    - Arrays, forall, basic operations
  - Task parallel features
    - Begin, cobegin, coforall, sync, etc
  - Distributed computing features
    - Locales, on-statement, domain maps
  - For all the above:
    - Concepts matter!
    - Precise syntax less so!
Parallel Programming Practical

• Kickoff meeting on 31 October:
  – 13:30 – 15:15, HG-12A00
  – Students who want to do PPP in Period 3 may also attend this meeting

• Practical is ``on your own’’
  – Using material on Canvas
  – Guidance from Rutger Hofman(MPI, Java) and Pieter Hijma (Chapel)