Message Passing

Advanced Operating Systems
Tutorial 5
Tutorial Outline

• Review of Lectured Material
• Discussion: Barrelfish and multi-kernel systems
• Programming exercise
Review of Lectured Material

- **Implications of multicore systems**
  - Hardware trends; NUMA and heterogeneity in multicore systems
  - Challenges of NUMA systems – is a shared memory model appropriate?
  - Multi-kernel systems – distributed operating systems for multicore

- **Message passing systems**
  - Limitations of threads and lock-based concurrency
  - Multicore memory models; composition of lock-based code
  - Concepts of message passing systems
    - Interaction models; communication and the type system; naming communications
    - Message handling; immutability; linear types; use of an exchange heap
    - Pattern matching and state machines
    - Error handling; let-it-crash philosophy; supervision hierarchies; case study
  - Erlang and Scala+Akka as examples
Key Points

• Understand problems of scaling multicore systems while maintaining a shared memory programming model
  • The multi-kernel operating system model
  • The message passing programming model

• Reflect on the suitability of message passing as a concurrency primitive for future systems
  • Advantages and disadvantages compared to lock-based concurrency with shared mutable state
Discussion: Barrelfish


- Is the premise that messages are more suitable than shared memory for future systems reasonable?

- Does it make sense to run a distributed operating system on the cores of a single hardware device?

- Where is the boundary for a Barrelfish-like system?
  - Distinction between a distributed multi-kernel and a distributed system of networked computers?

- Barrelfish is clearly an extreme: a shared-nothing system implemented on a hardware platform that permits some efficient sharing
  - Is it a desirable extreme?
  - Current systems sit at the opposite extreme – shared everything, despite increasingly separate hardware resources
Programming Exercise

- Exercise 3 now available
  - Aim – to explore the ease of use of message passing programming for non-expert programmers