Transactions

Advanced Operating Systems
Tutorial 6
Tutorial Outline

• Review of lectured material
• Key points
• Discussion
  • Transactions for managing concurrency
  • Transactions vs. message passing
Review of Lectured Material

• Concepts of transactions
  • ACID properties
  • Concurrent execution
  • Possible to compose transactions

• Implementation challenges
  • Controlling I/O operations
  • Controlling memory access – rollback and recovery
  • Implementation using monadic concepts

• Integration into Haskell

• Integration challenges for other languages
Key Points

• Understanding of the concepts of transactions
• Understanding of implementation techniques in functional languages
• Awareness of practical challenges
Discussion: Transactions

- T. Harris, S. Marlow, S. Peyton Jones and M. Herlihy, “Composable Memory Transactions”, CACM, 51(8), August 2008. DOI:10.1145/1378704.1378725

- Is transactional memory a realistic technique?
- Do its requirements for a purely functional language, with controlled I/O, restrict it to being a research toy?
- How much benefit can be gained from transactional memory in more traditional languages?
Discussion: Transactions vs. Messages

- Two very different approaches to concurrency offered by transactions and message passing

- Conceptual purity vs. engineering pragmatics?
  - Message passing is intuitive, easy to integrate into existing systems, but doesn’t solve the problem of composition?
  - Transactions are theoretically elegant, but cannot be integrated into real-world systems?

- How should future systems be designed?

- Are we still missing the right programming model for massively concurrent systems?
Next Lecture

• No AOS(M) lectures next week – next lecture is on 16 March 2015