Concepts Over Facts in HPC Education

David Marchant
Niels Bohr Institute, Copenhagen University

Introduction

- This presentation supports the paper ‘Teaching Concurrent and Distributed Programming With Concepts Over Mathematical Proofs’.

- Worked in Conjunction with Brian Vinter, Kenneth Skovhede, and Carl-Johannes Johnsen.

- Focus on how a concept focussed approach was introduced.

Background

- New HPC Course needed from 2018/2019.
- Update previous materials and techniques.
- Determine way of introducing complex topic to non-computer scientists.

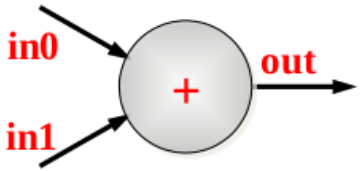
Old Material

- Focussed on the facts of parallel programming.
- Encouraged rote memorization of technical details, inhibited comprehension.
- Looked ugly and inconsistent.

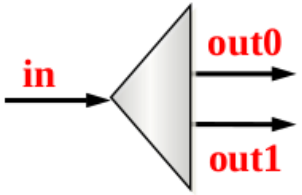
‘Legoland’ Catalog

IdInt (in, out)

SuccInt (in, out)



PlusInt (in0, in1, out)



Delta2Int (in, out0, out1)

PrefixInt (n, in, out)

TailInt (in, out)

Better approach

$$S = m!x \rightarrow m_{\text{ACK}} \rightarrow S'$$

$$B_c = m?x \rightarrow \left(\Vert_{i=0}^{n} c_i!x \rightarrow c_{i,\text{ACK}} \rightarrow \checkmark \right) ; \; m_{\text{ACK}} \rightarrow B_c'$$

$$P_i = c_i?x \rightarrow c_{i,\text{ACK}} \rightarrow P_i'$$

$$S \parallel B_c \parallel \left(\Vert_{i=0}^{n} P_i \right)$$

Facts vs Concepts

- New course should focus on concepts of parallel.
- Foster more conversation and experimentation.
- But we still need facts.

A Curriculum of Concepts

- Start with CSP to introduce concurrent and parallel concepts locally.
- ZeroMQ can introduce distributed systems.
- SME acts as a way into hardware programming.

A Curriculum of Concepts

- CSP – Communicating Sequential Processes
- Introduces non-determinism, race conditions, deadlock and other base concepts.
- PyCSP allows for simple to learn syntax that science students should be familiar with.

A Curriculum of Concepts

- ZeroMQ allows students to explore distributed systems.
- Easy and common protocol to understand.
- PyZMQ is again simple for students.

A Curriculum of Concepts

- SME – Synchronous Message Exchange
- Compiles into VHDL code for FPGAs
- Coding/testing is a lot simpler and quicker than VHDL.

Fostering Concepts

- More conversation.
- Learn by doing.
- Only show necessary facts on slides.

Designing Materials

- Slides should be simple and clean.
- Only a few key supporting facts on each slide.
- Adopt a non-linear structure.

Concepts Over Facts in HPC

David Marchant
Niels Bohr Institute, Copenhagen University

Introduction

- This presentation supports the paper 'Teaching Concurrent and Distributed Programming With Concepts Over Mathematical Proofs'.
- Worked in Conjunction with Carl-Johannes Johnsen, Brian Vinter and Kenneth Skovhede.
- Focus on how a Conceptual approach was introduced.

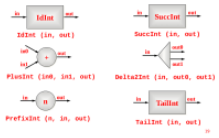
Background

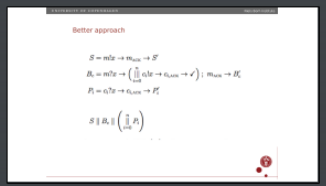
- New HPC Course needed from 2018/2019.
- Update previous materials and techniques.
- Determine way of introducing complex topic to non-computer scientists.

Old Material

- Focussed on the facts of parallel programming.
- Encouraged rote memorization of technical details.
- Looked ugly and inconsistent.

'Legoland' Catalog





Facts vs Concepts

- Facts are base pieces of knowledge.
- Easy to be rote memorized through one-way learning.
- Are foundation of, but not the same as a deeper understanding of core issues.

Facts vs Concepts

- Concepts are core to understanding.
- Can be extrapolated out to new knowledge.
- Requires facts, time, and experience.

Facts vs Concepts

- New course should focus on concepts of parallel.
- Foster more conversation and experimentation.
- But we still need facts.

Fostering Concepts

- More conversation.
- Learn by doing.
- Only show necessary facts on slides.

Designing Materials

- Slides should be simple and clean.
- Only a few key supporting facts on each slide.
- Adopt a non-linear structure.

A Curriculum of Concepts

- Start with CSP to introduce concurrent and parallel concepts locally.
- ZeroMQ can introduce distributed systems.
- SME acts as a way into hardware programming.

A Curriculum of Concepts

- CSP – Communicating Sequential Processes
- Introduces non-determinism, race conditions, deadlock and other base concepts.
- PyCSP allows for simple to learn syntax that science students should be familiar with.

A Curriculum of Concepts

- ZeroMQ allows students to explore distributed systems.
- Easy and common protocol to understand.
- PyZMQ is again simple for students.

A Curriculum of Concepts

- SME – Synchronous Message Exchange
- Compiles into VHDL code for FPGAs
- Coding/testing is a lot simpler and quicker than VHDL.

Designing a System

- We need to design an IoT system for a building.
- There will be a central control device that can update or shutdown all other devices.
- In each room there is an infrared sensor to detect when the room is in use. If it is, then it should turn on the light for that room. If no activity is detected in a certain amount of time then the light should be turned off.
- The central control device should be able to display status information about each device.

Designing a System

- To solve this problem you will need to define many things.
- When does the status update?
- Is the timer part of the light, the sensor, or the controller?
- Does the control device need a connection to all process, or can it propagate over a network?

Designing a System

- Our system is so good that everyone wants in our building, so we're going to have to add a security system.
- Each door has an Card Scanner. When a card is scanned it will consult a database of acceptable cards. If a card is accepted, the scanner will unblock the Lock for the corresponding door only.
- As an additional security measure, only one Lock can be unlocked at a time.
- The central controller still needs its status updates on the Locks, Scanners, and Database.
- The central controller must be able to unlock all doors in an emergency

Results

- Curriculum worked well.
- Conceptual understanding was demonstrated by students.
- Takes a lot of time to implement.
- Small sample size limits conclusions.

Acknowledgments

- Thank you to my supervisors:
  - Brian Vinter, NBI
  - Hans Fangohr, EuXFEL
- The European Training Network MUMMERING
- The EU for financial support as part of grant no. 765604.

Close

Designing a System

- We need to design an IoT system for a building.
- There will be a central control device that can update or shutdown all other devices.
- In each room there is an infrared sensor to detect when the room is in use. If it is, then it should turn on the light for that room. If no activity is detected in a certain amount of time then the light should be turned off.
- The central control device should be able to display status information about each device.

Designing a System

- To solve this problem you will need to define many things.
- When does the status update?
- Is the timer part of the light, the sensor, or the controller?
- Does the control device need a connection to all process, or can it propagate over a network?

Designing a System

- Our system is so good that everyone wants in our building, so we’re going to have to add a security system.
- Each door has an Card Scanner. When a card is scanned it will consult a database of acceptable cards. If a card is accepted, the scanner will unlock the Lock for the corresponding door only.
- As an additional security measure, only one Lock can be unlocked at a time.
- The central controller still needs its status updates on the Locks, Scanners, and Database.
- The central controller must be able to unlock all doors in an emergency.

Results

- Curriculum worked well.
- Conceptual understanding was demonstrated by students.
- Takes a lot of time to implement.
- Small sample size limits conclusions.

Acknowledgments

- Thank you to my supervisors:
  - Brian Vinter, NBI
  - Hans Fangohr, EuXFEL

- The European Training Network MUMMERING

- The EU for financial support as part of grant no. 765604.