Teaching with Chapel

Kyle Burke
Wittenberg University

Supercomputing 2011

Teaching with Chapel

Two Classes

Programming Languages

Spring 2010, 2011

Teaching with Chapel

Two Classes

Programming Languages

Spring 2010, 2011

Analysis of Algorithms

Fall 2010, 2011

Programming Languages

- For students with data-structures class

Programming Languages

- For students with data-structures class
- Paradigms:
  - Functional: Scheme
  - Logical: Prolog
  - Event-Driven: Java
  - Object-Oriented: Java

Programming Languages

- For students with data-structures class
- Paradigms:
  - Functional: Scheme
  - Logical: Prolog
  - Event-Driven: Java
  - Object-Oriented: Java
  - High-Performance: Chapel

Programming Languages

- Chapel Topics:
  - task generation (begin, cobegin)

Programming Languages

- Chapel Topics:
  - task generation (begin, cobegin)
  - parallel iteration (forall, coforall)

Programming Languages

- Chapel Topics:
  - task generation (`begin`, `cobegin`)
  - parallel iteration (`forall`, `coforall`)
  - race conditions (`sync`)

Programming Languages

- Chapel Topics:
  - task generation (begin, cobegin)
  - parallel iteration (forall, coforall)
  - race conditions (sync)
  - language additions (reduce)

Programming Languages

- Chapel Topics:
  - task generation (`begin`, `cobegin`)
  - parallel iteration (`forall`, `coforall`)
  - race conditions (`sync`)
  - language additions (`reduce`)
- Cover lots of HPC material

Programming Languages

- Projects:
  - binary xor
  - matrix multiplication
  - collatz conjecture testing

Programming Languages

- Projects:
  - binary xor
  - matrix multiplication
  - collatz conjecture testing

Programming Languages

Matrix Multiplication

Serial Time: $\Theta\left(n^3\right)$

Programming Languages

Matrix Multiplication

Serial Time: $\Theta(n^3)$

Parallel Time: $\Theta(n^2)$ (n processors)

Programming Languages

Conclusions

- Lots of material
- Usually favorite language in class

Analysis of Algorithms

- For students with data-structures and discrete math

Analysis of Algorithms

- For students with data-structures and discrete math
- Already Sequential and Parallel

Analysis of Algorithms

- For students with data-structures and discrete math
- Already Sequential and Parallel
- Replaced C with Chapel
  - only teach `cobegin` and `forall`
  - ~ 1 day of class time (use tutorial)

Analysis of Algorithms

- Projects
  - set partition
  - sorting (mergeSort, bubbleSort)
  - nearest neighbors

Analysis of Algorithms

- Projects
  - set partition
  - sorting (mergeSort, bubbleSort)
  - nearest neighbors

Project: Nearest Neighbors

Project: Nearest Neighbors

Project: Nearest Neighbors

- Two Algorithms:
  - Divide-and-Conquer:

$$\Theta(n\log(n)) \rightarrow \Theta(n)$$

Project: Nearest Neighbors

- Two Algorithms:
  - Divide-and-Conquer:

$$\Theta(n\log(n)) \rightarrow \Theta(n)$$

  - Brute-Force:

$$\Theta(n^2) \rightarrow \Theta(n)$$

Project: Nearest Neighbors

- Two Algorithms:
  - Divide-and-Conquer:

$$\Theta(n\log(n)) \rightarrow \Theta(n)$$

  - Brute-Force:

$$\Theta(n^2) \rightarrow \Theta(n)$$

Divide-and-Conquer is more difficult to program...

Algorithms

Conclusion

Little Class Time to Teach,

Students learn Parallel Theory

Conclusions

- Chapel has easy-to-learn parallel constructs
- Less time lecturing, more time using
- Useful in different contexts

Conclusions

- Wittenberg: modest linux cluster
  - speedup noticeable
  - bigger cluster would be better
  - biggest problem: cluster issues!

Conclusions

- More info:
  - me: kburke@wittenberg.edu
  - Chapel Education: http://chapel.cray.com/education.html
  - Chapel Education sourceforge mailing list

Conclusions

- More info:
  - me: kburke@wittenberg.edu
  - Chapel Education: http://chapel.cray.com/education.html
  - Chapel Education sourceforge mailing list

Thank You!

Extra Thanks: Ernie Heyder