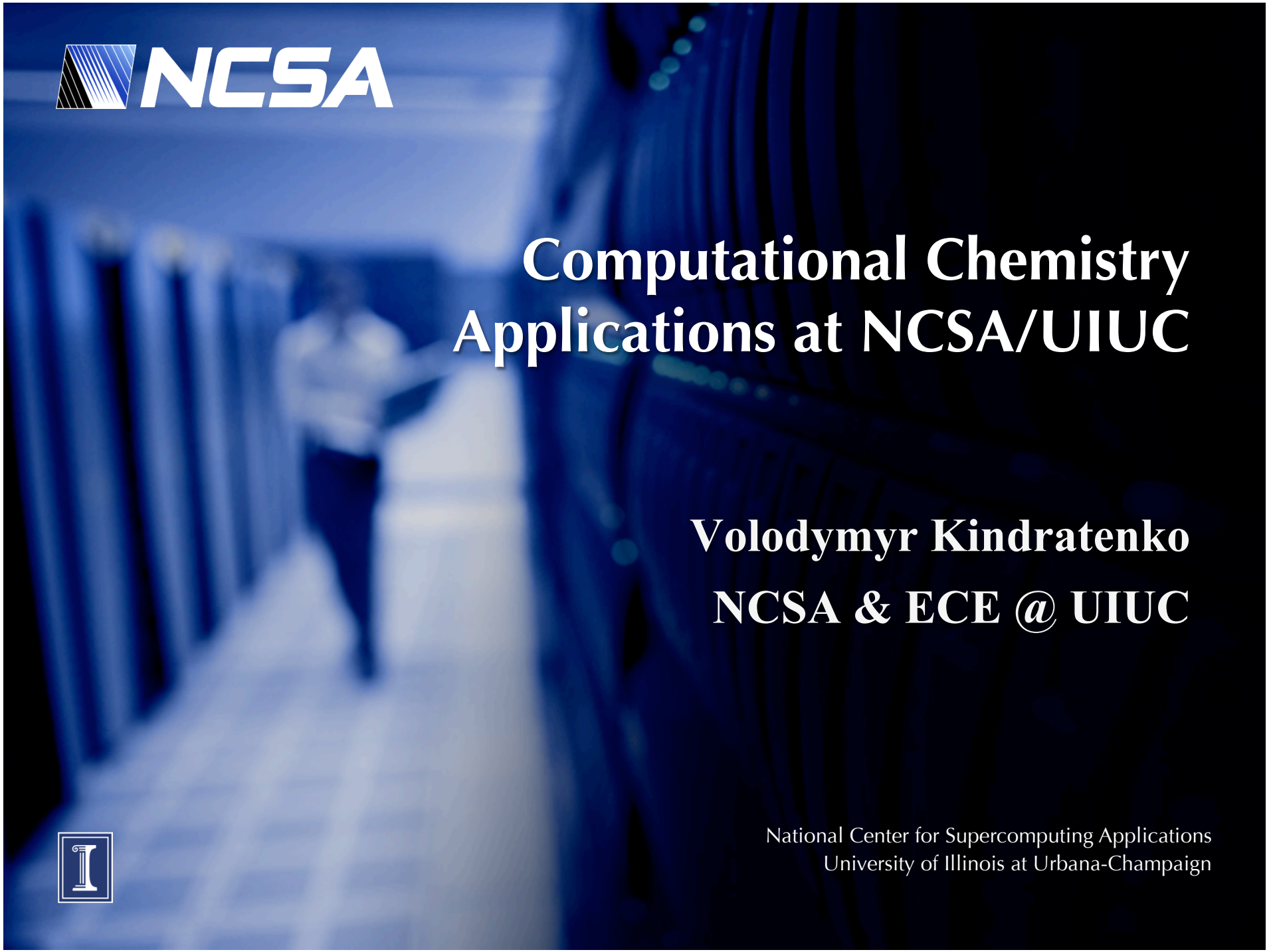


Computational Chemistry Applications at NCSA/UIUC

Volodymyr Kindratenko

NCSA & ECE @ UIUC

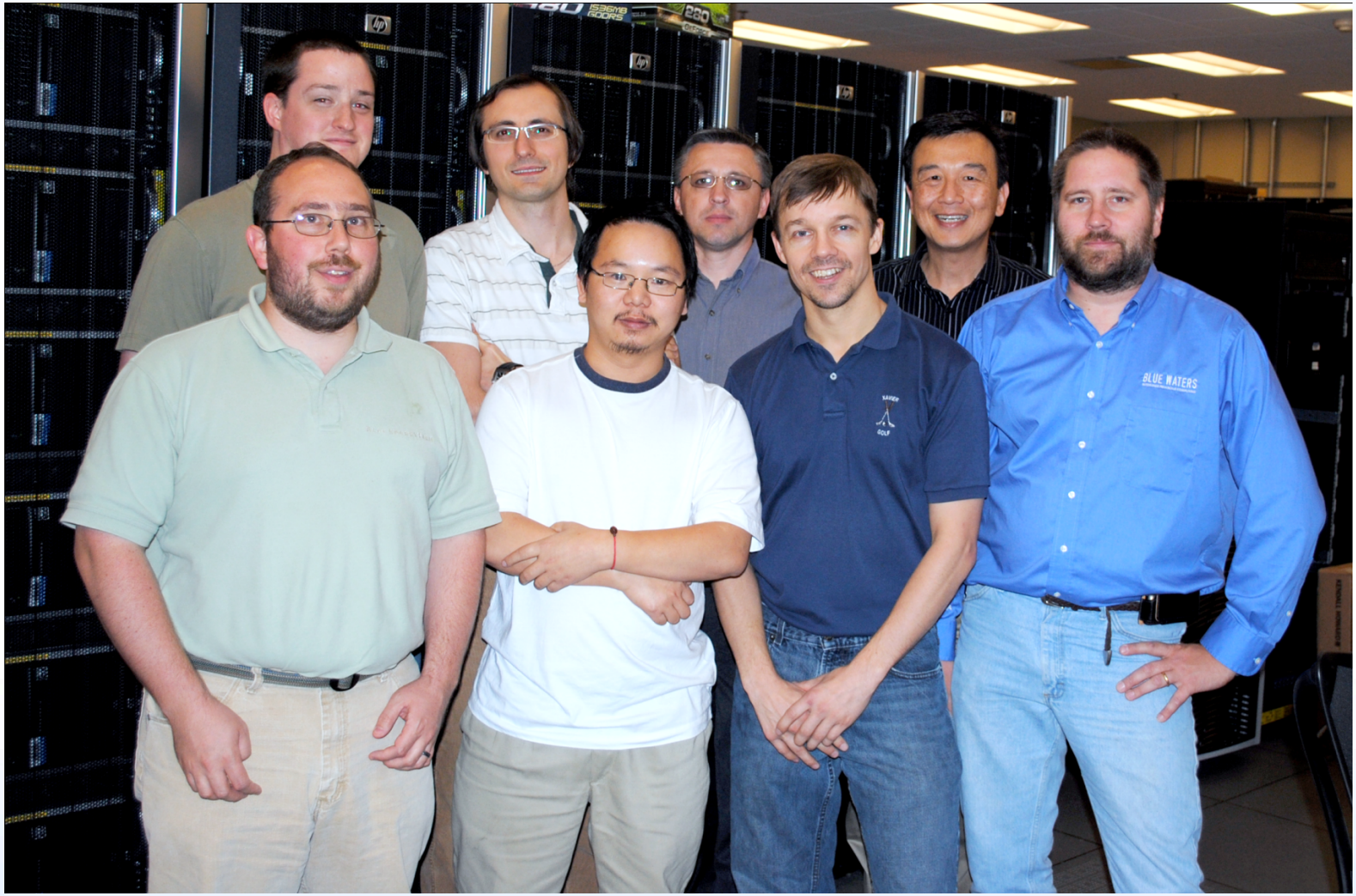
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

National Center for Supercomputing Applications

Blue Waters


NCSA
IBM
BLUE WATERS
SUSTAINED PETASCALE COMPUTING
ILLINOIS

Innovative Systems Laboratory at NCSA



ISL Research

- **Evaluation of emerging computing architectures**
  - Reconfigurable computing
  - Many-core (GPU) architecture
  - Heterogeneous clusters
- **Systems software research and development**
  - Run-time systems
  - GPU accelerator cluster management
    - Tools and utilities: GPU memory test, power profiling, etc.
- **Application development on emerging computing applications**
  - Computational chemistry (electronic structure, MD)
  - Computational physics (QCD)
  - Cosmology
  - Data mining

Computational Chemistry Applications at NCSA/UIUC

- **NAMD**
  - Theoretical and Computational Biophysics Group at Beckman Institute
  - Key researchers
    - Prof. Klaus Schulten, Director
    - Dr. James Phillips, NAMD developer
    - John Stone, VMD developer
  - http://www.ks.uiuc.edu/Research/namd/
- **Electronic structure**
  - Innovative Systems Laboratory at NCSA
  - Key researchers
    - Prof. Thom Dunning, project PI
    - Prof. Todd Martinez, project co-PI, now at Stanford
    - Dr. Volodymyr Kindratenko, project co-PI
    - Dr. Alexey Titov, quantum chemistry on GPU research/development
    - Guochun Shi, GPU/Cell development/implementation
  - http://cyberchem.ncsa.illinois.edu/

NAMD Acceleration Efforts: FPGAs

- **Work done at ISL (V. Kindratenko)**
- **SRC-6 Reconfigurable Computer platform**
- **Observations/Conclusions**
  - FPGA technology (Xilinx Virtex II) enabled some acceleration
    - ~3x speedup for the non-bonded force-field kernel
  - But it quickly started to loose its ground to the emerging multi-core architectures
    - We have not revisited this with the latest (e.g., Virtex 6) FPGAs
  - Programming efforts are substantial
  - Hardware is expensive and is not widely used

Ref: V. Kindratenko, and D. Pointer, A case study in porting a production scientific supercomputing application to a reconfigurable computer, in Proc. IEEE Symposium on Field-Programmable Custom Computing Machines - FCCM'06, 2006, pp. 13-22.

NAMD Acceleration Efforts: Cell

- **Work done at ISL (G. Shi)**
- **IBM Cell blade platform**
- **Observations/Conclusions**
  - Cell enabled a very substantial acceleration
    - ~45x speedup for the non-bonded force-field kernel
    - A very high percentage of device utilization is achievable
  - Programming efforts are substantial
    - Explicit code vectorization, explicit memory management, …
  - Hardware is expensive and is not widely used

Ref: G. Shi, V. Kindratenko, Implementation of NAMD molecular dynamics non-bonded force-field on the Cell Broadband Engine processor, In Proc. 9th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing - PDSEC'08, 2008.

NAMD Acceleration Efforts: GPU

- **Work done at TCBG (J. Phillips)**
- **NVIDIA CUDA GPU platform**
- **Observations/Conclusions**
  - Moderate acceleration is achievable
    - ~5-7x speedup for the non-bonded force-field kernel on GT100 architecture
    - Note that the original CPU code is already highly optimized
  - The GPU version of NAMD is now available for download from the developers' website
  - Programming efforts are substantial
    - Memory management, overlapping data transfers with computations, …
  - Commodity nature of GPUs makes it practical for NAMD

Electronic structure calculations: FPGAs

- **Work done at ISL (V. Kindratenko)**
- **SRC-6/7 Reconfigurable Computer platform**
- **Observations/Conclusions**
  - Available hardware resources were only sufficient to implement a highly restructured and simplified kernel for computing (ss|ss) integrals over contracted *s*-orbitals, with marginal speedup
    - We have not revisited this with the latest (e.g., Virtex 6) FPGAs
  - Programming efforts are substantial

Ref: V. Kindratenko, I. Ufimtsev, T. Martínez, Evaluation of two-electron repulsion integrals over Gaussian basis functions on SRC-6 reconfigurable computer, In Proc. 4th Annual Reconfigurable Systems Summer Institute - RSSI'08, 2008.

Electronic structure calculations: Cell

- **Work done at ISL (G. Shi)**
- **IBM Cell blade platform**
- **Observations/Conclusions**
  - Cell enabled a very substantial acceleration
    - ~7x speedup on relatively uncontracted basis set
    - ~65x speedup on a highly contracted basis set
    - (kernels for computing (ss|ss) integrals over contracted *s*-orbitals)
  - Programming efforts are substantial
    - Explicit code vectorization, explicit memory management, …
  - Hardware is expensive and is not widely used

Ref: G. Shi, V. Kindratenko, I. Ufimtsev, T. Martinez, J. Phillips, S. Gottlieb, Implementation of scientific computing applications on the Cell Broadband Engine, Scientific Programming, vol. 17, no. 1-2, pp. 135-152, 2009.

Electronic structure calculations: GPU

- **Work done at ISL (A. Titov)**
- **NVIDIA CUDA GPU platform**
- **Observations/Conclusions**
  - Substantial all-application speedup (SCF including *s*-, *p*-, and *d*-orbitals)
    - Up to 50x compared to GAMESS performing the same calculations
  - Programming efforts are huge
    - ~300,000 lines of autogenerated GPU code
    - Symbolic computations are used to manipulate the contraction equations needed for GPU implementation
  - This work is now integrated into a commercial code TeraChem from Stanford

Ref: A. Titov, V. Kindratenko, I. Ufimtsev, T. Martinez, Generation of Kernels to Calculate Electron Repulsion Integrals of High Angular Momentum Functions on GPUs – Preliminary Results, In Proc. Symposium on Application Accelerators in High-Performance Computing - SAAHPC'10, 2010.

Summary

- **NVIDIA CUDA GPU efforts have been most fruitful so far**
  - Commodity nature of the hardware
    - Widely available, inexpensive
  - Relative maturity of the software development tools
- **But application acceleration continues to be very labor-intensive**
  - New/different algorithms
  - Extensive code transformations
- **When there are challenges, there are opportunities to do some interesting work and to collaborate**