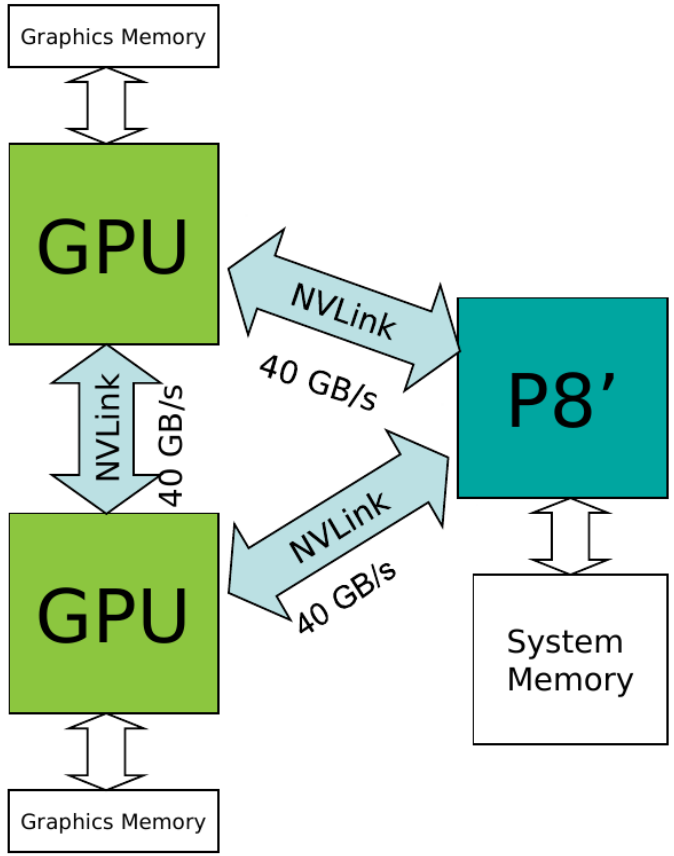
OpenPOWER™

GPU-accelerated POWER for Supercomputing

D. Pleiter (Jülich Supercomputing Centre)

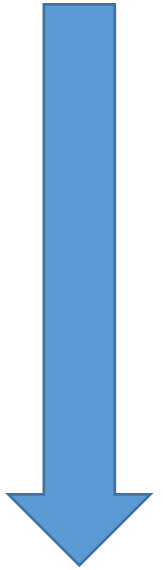
Introduction

- Massively-parallel compute devices are becoming commonly used in supercomputers
  - E.g., GPUs
- Significant changes for OpenPOWER server architectures like Minsky
  - More GPUs per CPU socket
  - High-speed interconnect CPU-GPU and GPU-GPU
  - Better support for data migration between compute devices
- Challenge: Application enablement
  - Efficient exploitation, maintaining scalability

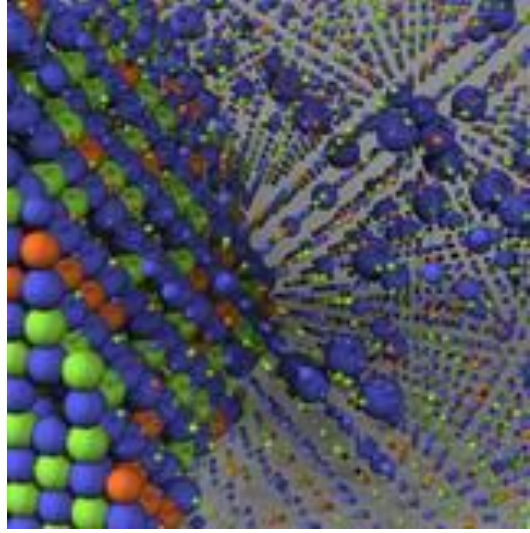


Approach to Application Enablement

- Application vs. HPC expert
  - Create common understanding of the application
- Process
  - Perform anamnesis
    - Define goals and define constraints
  - Create mini-applications
    - Easy to modify, simplified version of the application
  - Implement and evaluate proof-of-concepts
    - Proof benefits of specific porting strategies
  - Performance modelling
    - Create understanding for architectural requirements



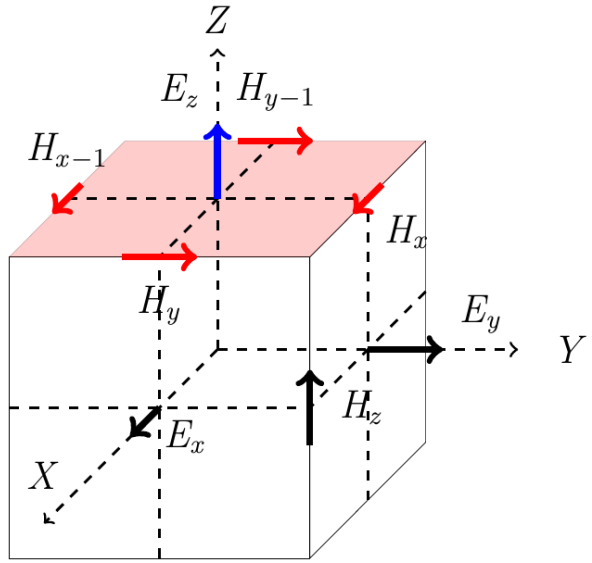
Applications: KKRnano

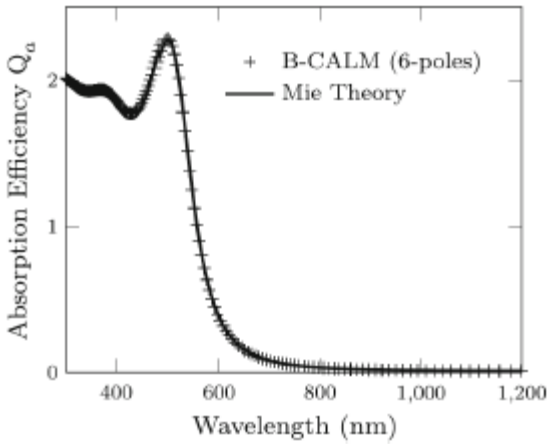


- Materials science application based on Density Functional Theory (DFT) method
  - High scalability due to truncation of long-range interactions → linear scaling in number of atoms
- Performance characteristics
  - Most time spent in iterative solver
  - Dense matrix-matrix multiplications dominate performance (AI ≥ 4)
- Implementation properties: Fortran, MPI, OpenMP
- Exascale needed for simulating systems with $O(10^6)$ atoms

Applications: B-CALM

- Based on FDTD
  = Finite Difference Time Domain
  - Method for electro-magnetic calculations
- Example usages: Analysis of dispersive media
  - Information technology: Development of optical interconnects
  - Energy technology: Research on photo-electric cells

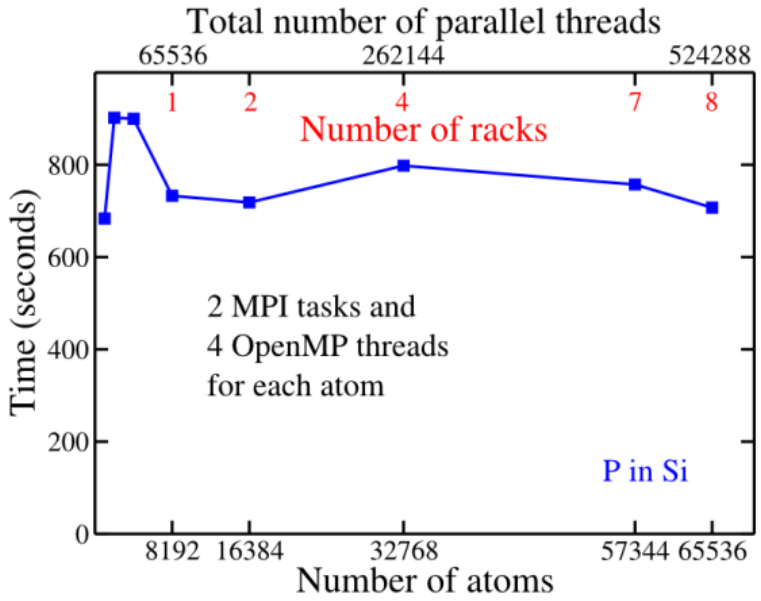


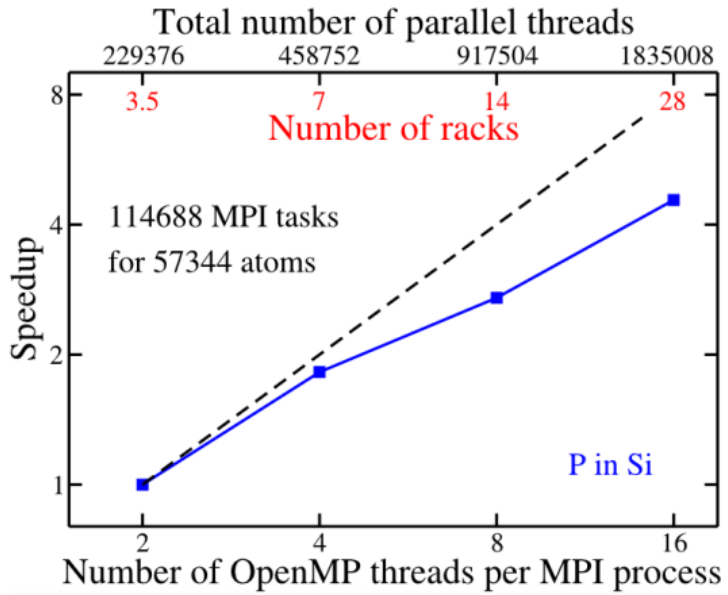


[P. Wahl et al., 2012]

Focus on Scalability: High-Q Club

- Eligible members: Applications that demonstrated scalability up to 28 Blue Gene/Q racks, i.e. 458,752 cores
- Example: KKRnano



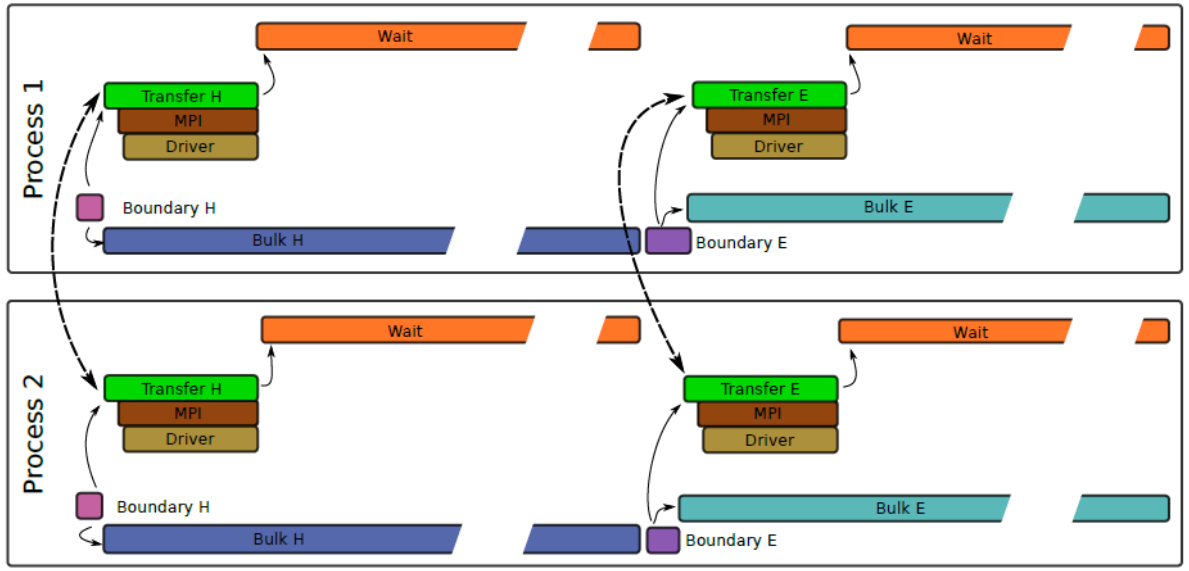


Research Questions

- Research questions
  - How well can application exploit architecture?
  - How could architecture optimized for application?
- Methodology based on performance models to enable comparison with architectural parameters
  - Support implementation decisions
  - Enable understanding of optimal performance
  - Allow for hypothetical tuning of hardware parameters

B-CALM on OpenPOWER

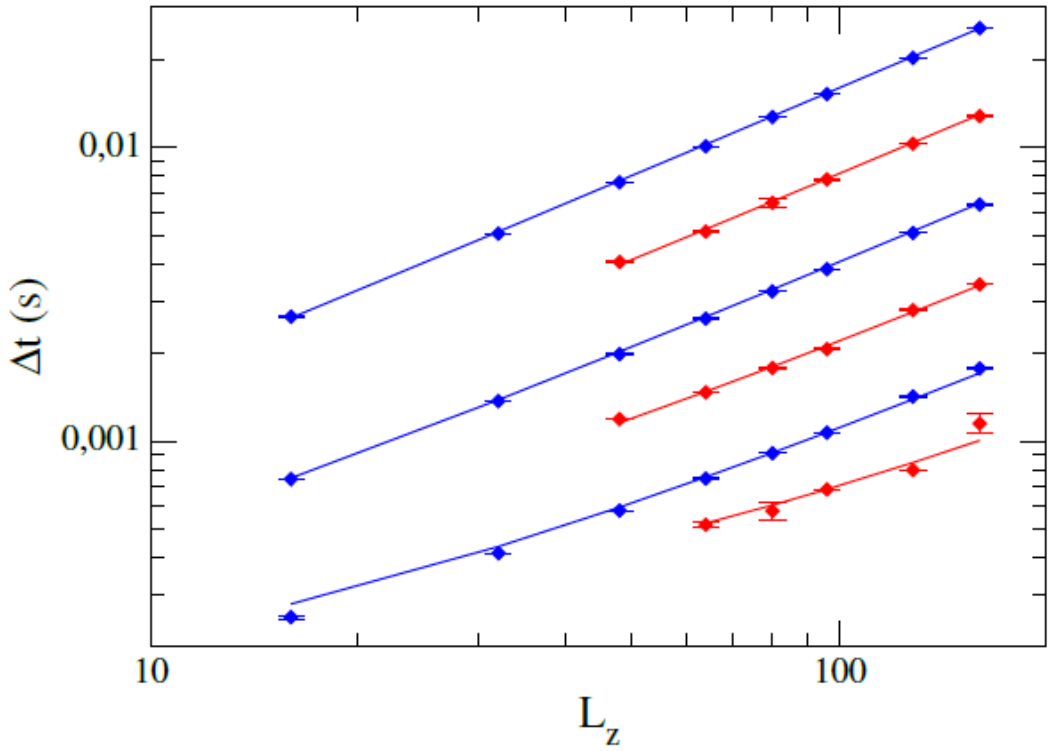
- No specific porting efforts towards POWER + GPU required
  - Main kernels had been ported to GPU already
- Scalability challenge: efficient overlap of communication and computation



B-CALM Results

[P. Baumeister et al., 2015]

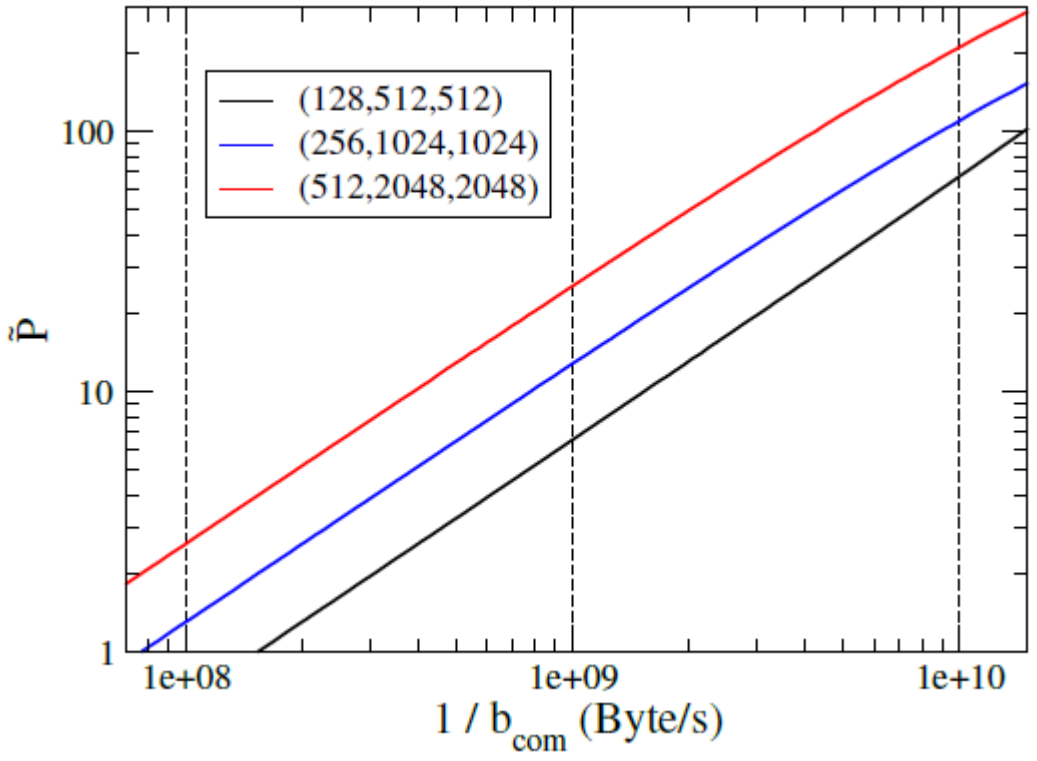
- Ansatz: Model kernel execution time as function of information exchanged between GPU and its memory
- Measurements performed on POWER8 server with K40 GPUs
  - Results for different choices of $L_x = L_y$
  - Different number of MPI ranks



Exploring B-CALM Scaling Limits

[P. Baumeister et al., 2015]

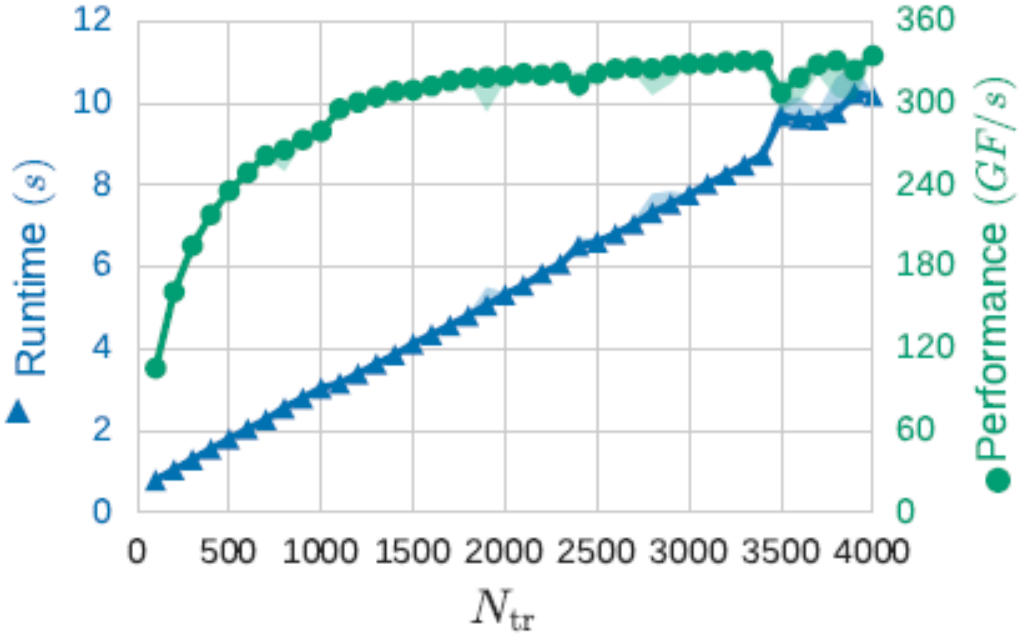
- Ansatz: Model time needed for communication as function of exchanged information
- Balance condition: Perfect overlap of computation and communication
  → Relation between number of MPI ranks and network bandwidth



Porting KKRnano to OpenPOWER

[P. Baumeister et al., 2016]

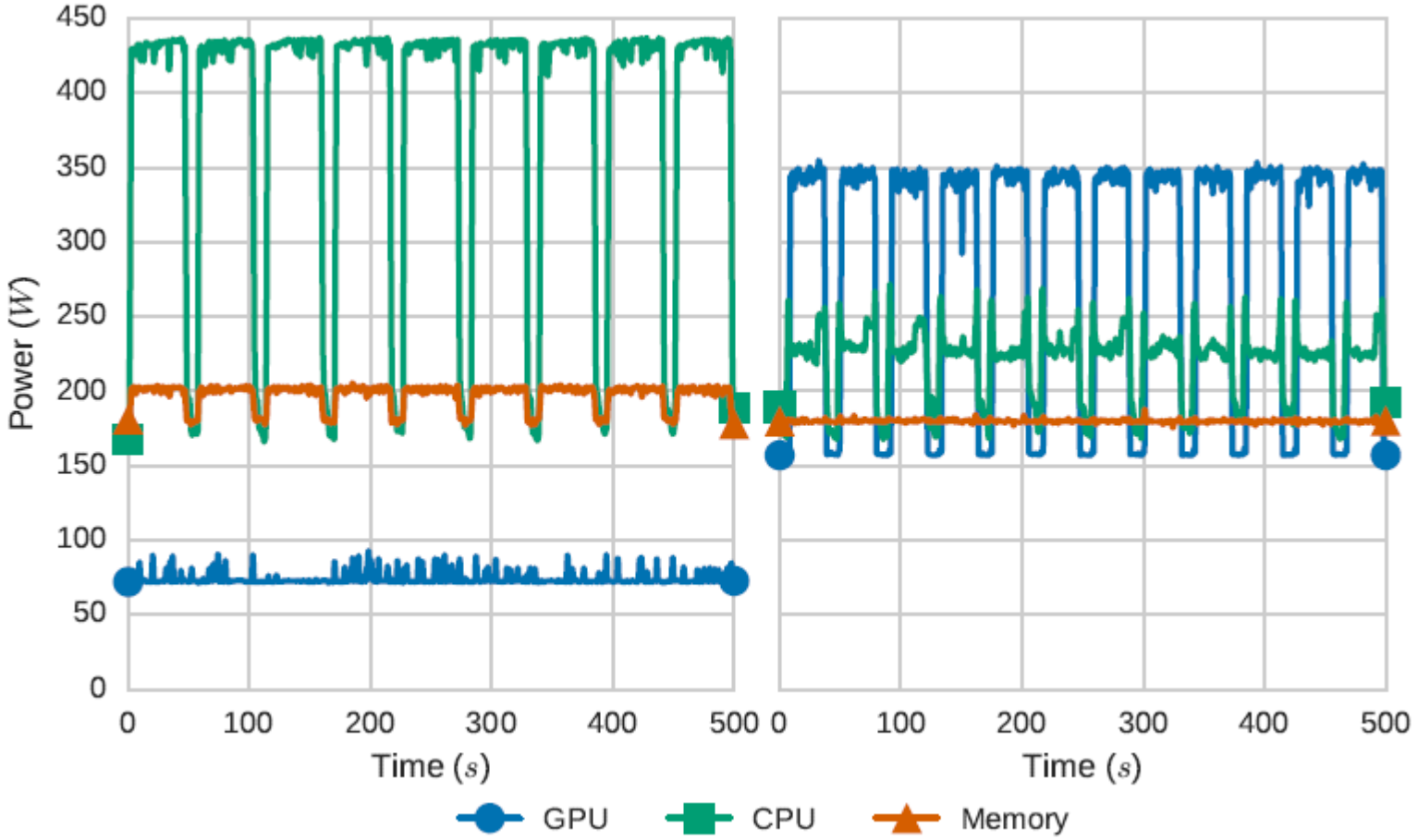
- Porting strategy
  - Main application executed on POWER processor
  - Dedicated implementation of solver for POWER8 or GPU
- Performance limits
  - POWER8: compute-limited
    - Reach 365 GFlop/s
  - K40: memory-bandwidth limited
    - Reach 152 GByte/s



KKRnano Scaling Exploration

- Target system with performance ~6 PFlop/s
  - Requires ~2100 nodes with 2x POWER8 and 2x K40 GPUs
- Minimal problem size determination
  - Need >20 atoms to saturate single node resources
  - Need >42,000 atoms on target system
- Communication time determined from performance modelling approach
  - Find time for communication << time for computation
  - No communication during solver execution

KKRnano Power Efficiency Analysis



POWER8: 2.5 nJ/Flop
K40: 1.95 nJ/Flop
→ 22% gain

Summary and Conclusions

- Opportunity of new OpenPOWER architectures: broaden number of applications that can exploit GPU
  - High-bandwidth interconnect
  - System support for data transport
- Current systems suitable for “classical” compute-intensive applications
  - DFT-based application KKRnano
  - FDTD-based application B-CALM
- High scalability is achievable
  - Small number of fat nodes reduces node-scaling challenge