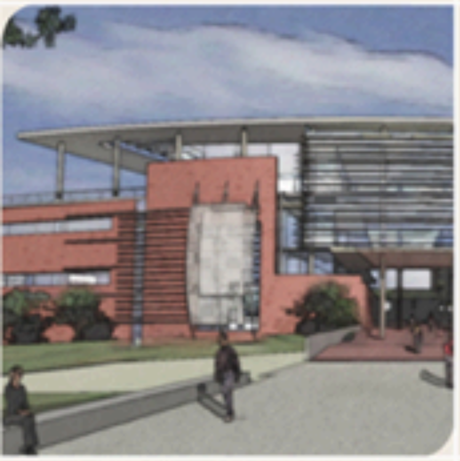




Energy Efficiency via the N-way Model

Romain Cledat and Santosh Pande

Georgia Tech College of Computing

Current trends

Current trends

- Moore's law is still in force:
  - More **cores** instead speed

Current trends

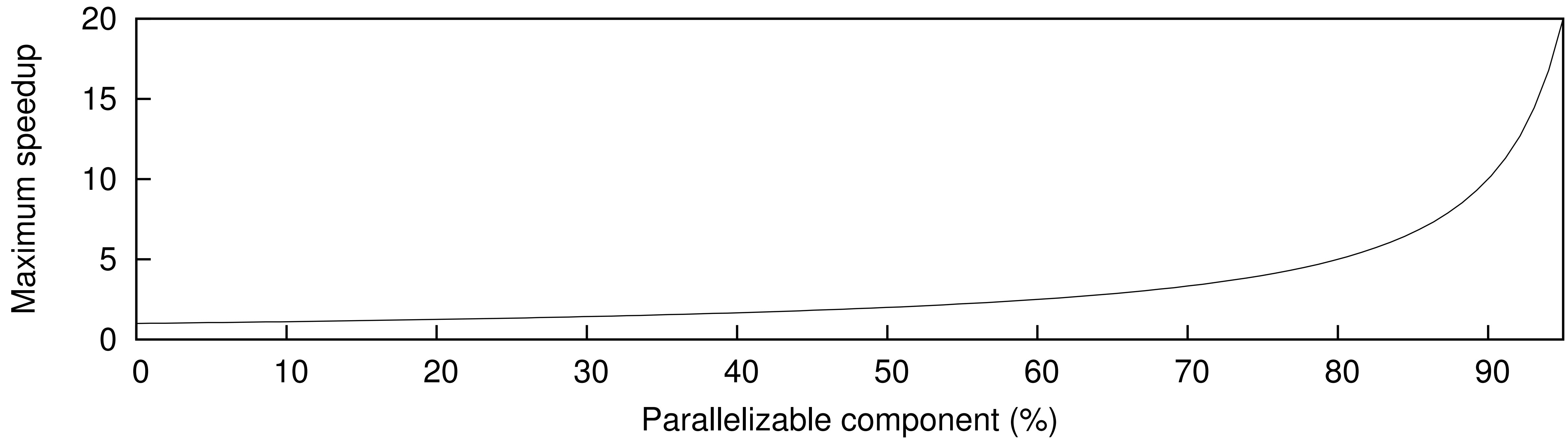
- Moore’s law is still in force:
  - More **cores** instead speed
- Higher power consumption but also finer control

Current trends

- Moore's law is still in force:
  - More **cores** instead speed
- Higher power consumption but also finer control
- Amdahl's law still restricts the performance of programs

Current trends

- Moore's law is still in force:
  - More **cores** instead speed
- Higher power consumption but also finer control
- Amdahl's law still restricts the performance of programs



Research questions

Research questions

- Parallel codes efficiently utilize parallel resources

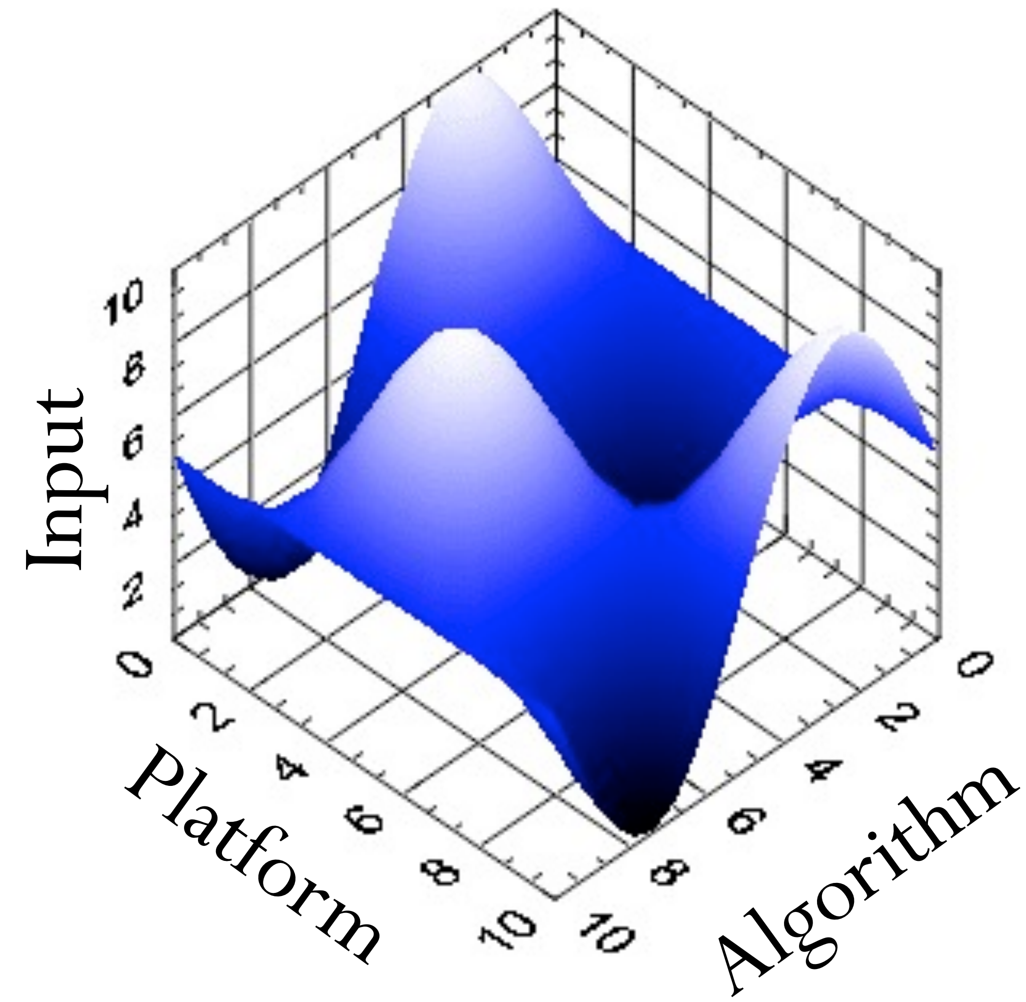
Research questions

- Parallel codes efficiently utilize parallel resources
- For sequential codes:
  - Can parallel resources be utilized?
    - ILP uses in-core parallelism
    - Other cores can be used to speculate, prefetch, etc.
  - Can they be used efficiently in terms of energy?

Research questions

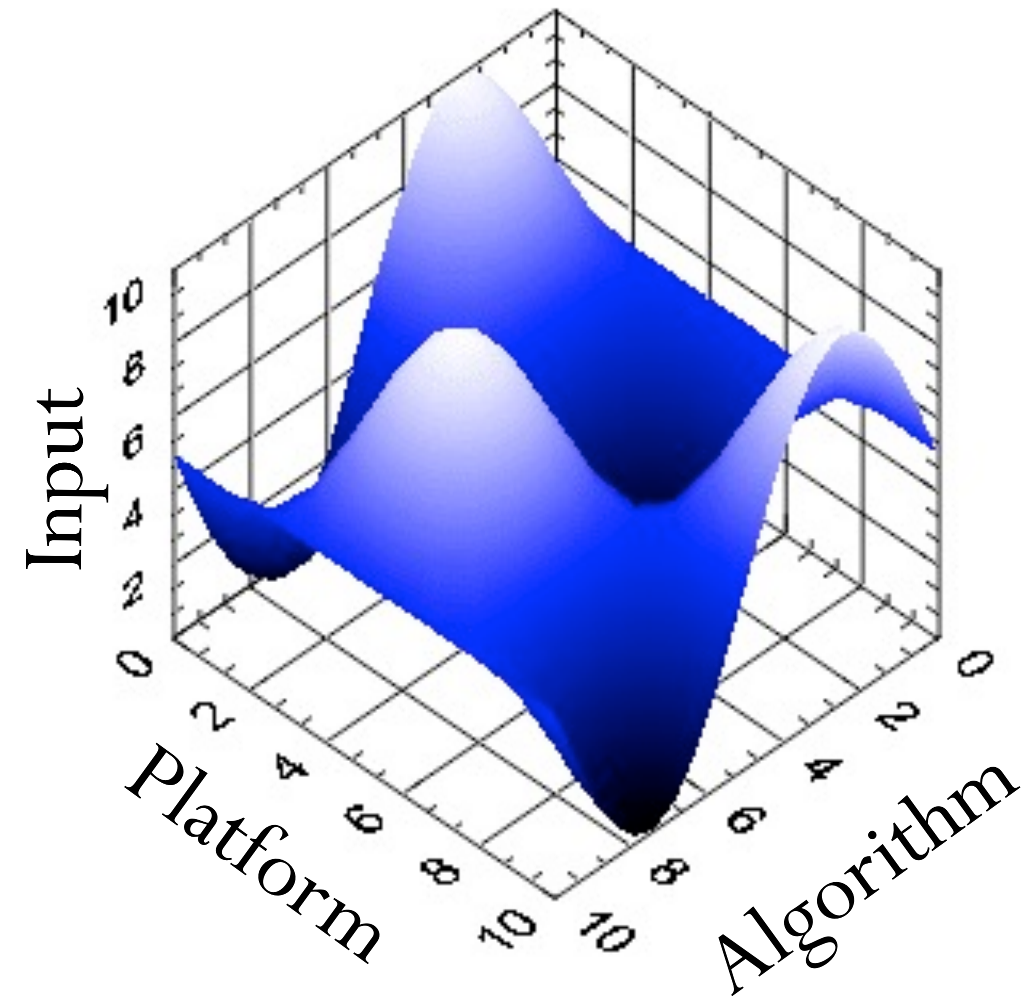
- Parallel codes efficiently utilize parallel resources
- For sequential codes:
  - Can parallel resources be utilized?
    - ILP uses in-core parallelism
    - Other cores can be used to speculate, prefetch, etc.
  - Can they be used efficiently in terms of energy?
- In this work, we present a use of parallel cores to improve the *algorithmic energy efficiency* of sequential algorithms

Algorithms, platforms and inputs



Algorithms, platforms and inputs

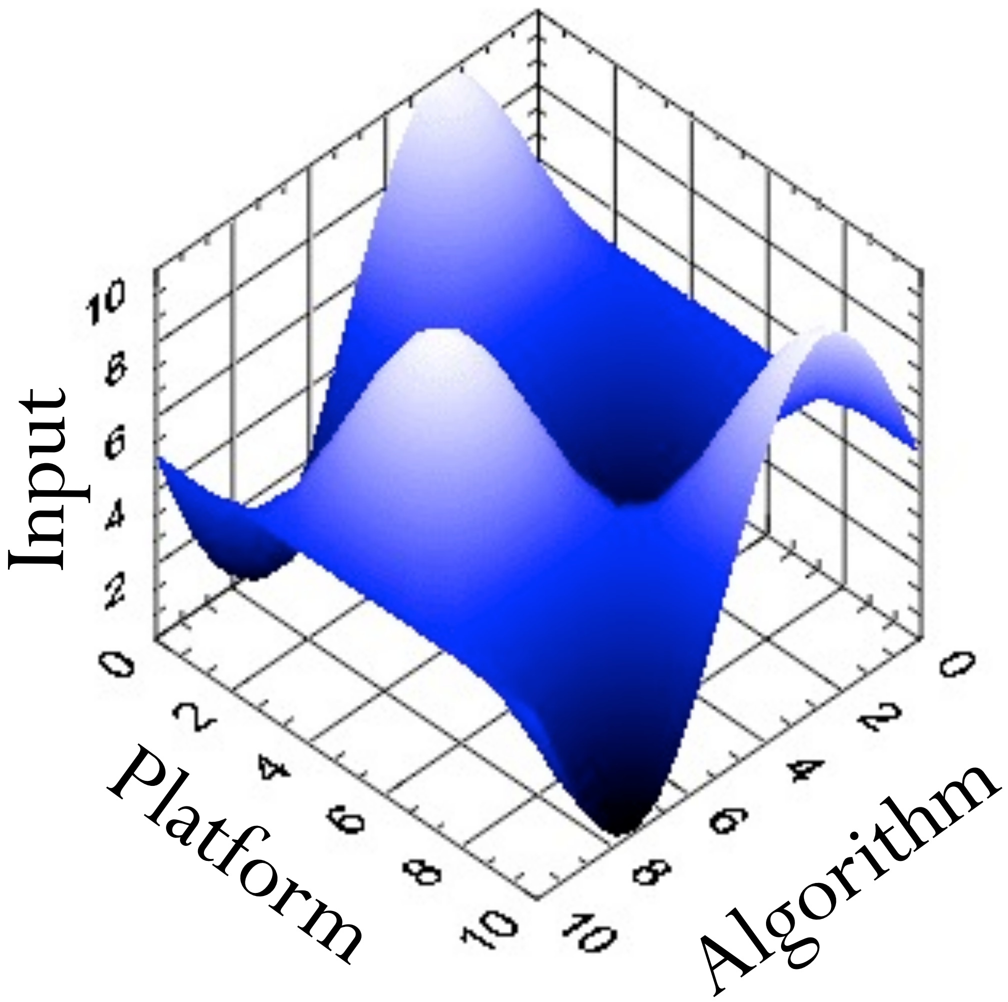
- An algorithm specifies a particular way to solve a problem



Algorithms, platforms and inputs

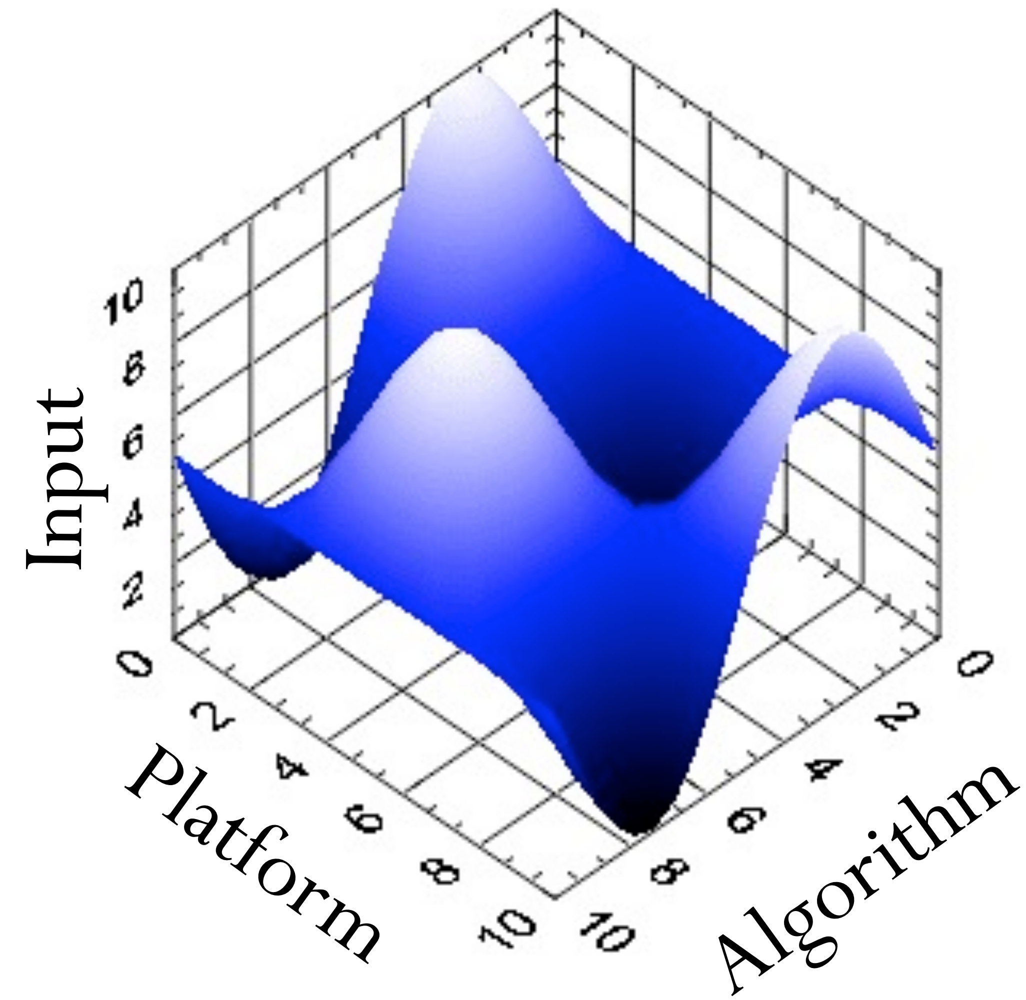
- An algorithm specifies a particular way to solve a problem

- It runs on a particular platform



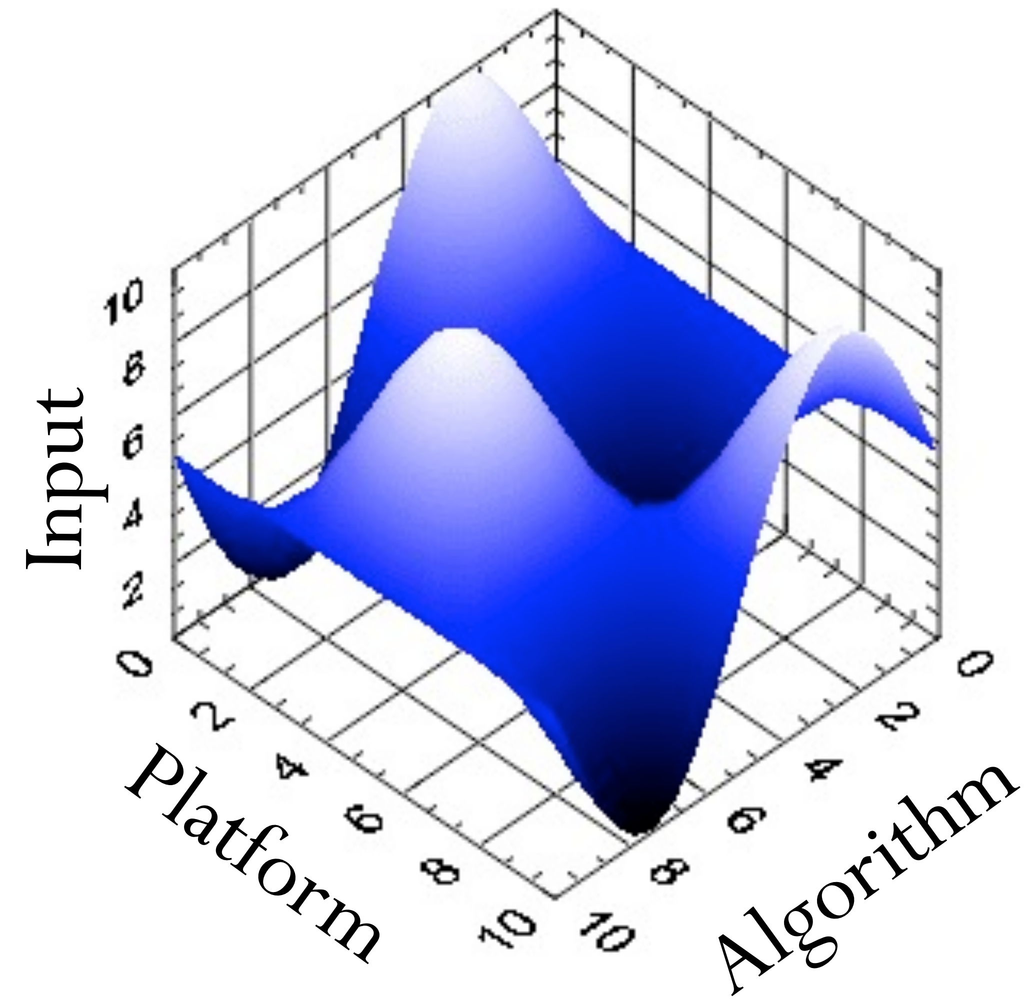
Algorithms, platforms and inputs

- An algorithm specifies a particular way to solve a problem
- It runs on a particular platform
- Its input data is dynamically provided at runtime



Algorithms, platforms and inputs

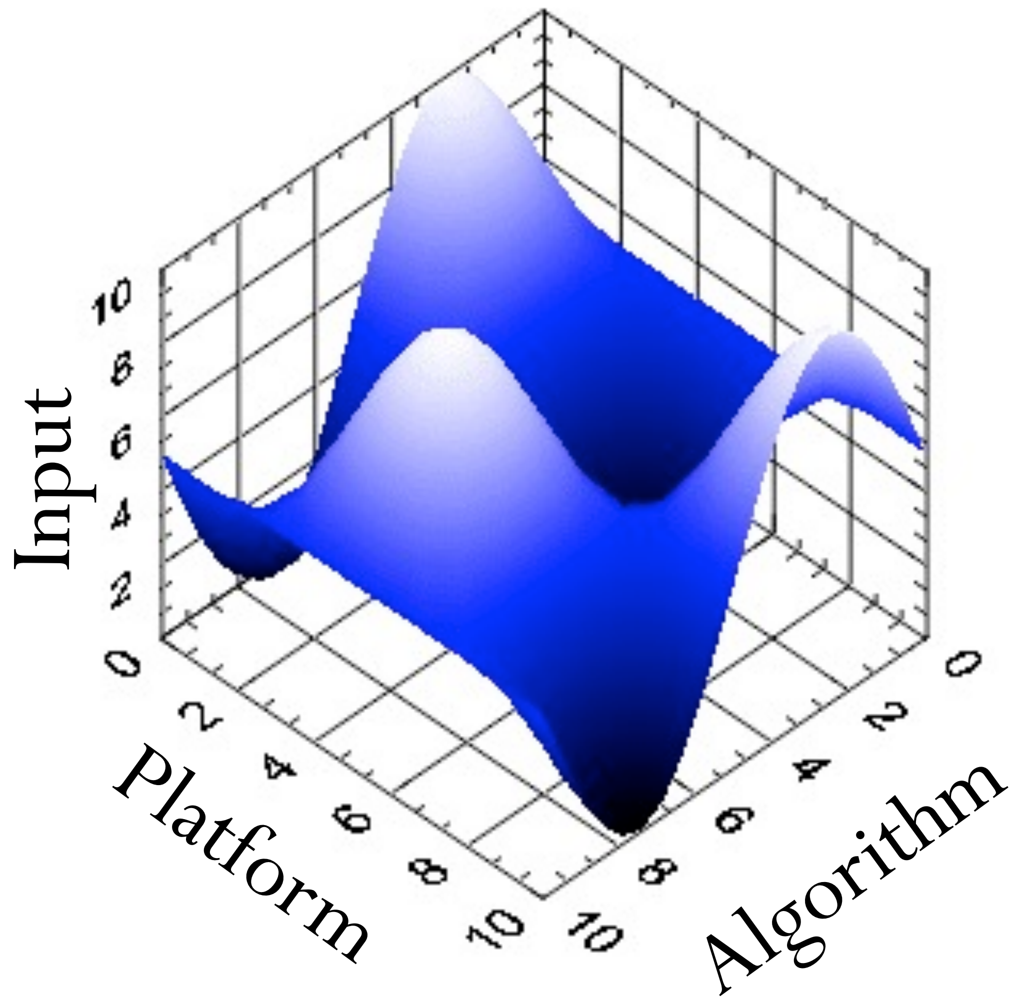
- An algorithm specifies a particular way to solve a problem
- It runs on a particular platform
- Its input data is dynamically provided at runtime



- **For a given problem and input, what is the best match?**

Algorithms, platforms and inputs

- An algorithm specifies a particular way to solve a problem
- It runs on a particular platform
- Its input data is dynamically provided at runtime



- For a given problem and input, what is the best match?
- For a given input, a particular algorithm may be better.

Outline

- The challenges of sequential code on multi-cores
- Using diversity to expose parallelism
- Measures of energy efficiency
  - Progress measure
  - Power measure
- The n-way programming model
- Preliminary results and future work

Diversity in platforms

Diversity in platforms

- Heterogeneity used to be present across machines
  - ASICs and FPGAs were optimized for certain algorithms

Diversity in platforms

- Heterogeneity used to be present across machines
  - ASICs and FPGAs were optimized for certain algorithms
- Today, chips are becoming heterogeneous
  - Asymmetric multi-cores to save space and power
  - Specialized accelerators (GPUs, network processors...)

Diversity in platforms

- Heterogeneity used to be present across machines
  - ASICs and FPGAs were optimized for certain algorithms
- Today, chips are becoming heterogeneous
  - Asymmetric multi-cores to save space and power
  - Specialized accelerators (GPUs, network processors...)
- The efficiency of each chip is different and depends on the algorithm
  - GPUs adapted for massive data parallelism
  - Cell SPUs excel at SIMD code

Diversity in platforms

Diversity in platforms

- Sequential algorithms currently do not exploit it
  - Parallelism limited to ILP
  - Extra cores can help in speculation, prefetching, ...

Diversity in platforms

- Sequential algorithms currently do not exploit it
  - Parallelism limited to ILP
  - Extra cores can help in speculation, prefetching, ...
- Cores can differ:
  - branch logic, prefetching logic, amount of speculation, cache sizes, ...
  - leads to different energy requirements depending on use

Diversity in platforms

- Sequential algorithms currently do not exploit it
  - Parallelism limited to ILP
  - Extra cores can help in speculation, prefetching, ...
- Cores can differ:
  - branch logic, prefetching logic, amount of speculation, cache sizes, ...
  - leads to different energy requirements depending on use
- Tools to match code to cores are rare (GLIMPSES for SPUs: http://sourceforge.net/projects/glimpses/)

Diversity in algorithms

Diversity in algorithms

- For a given problem, many “ways” to solve it

Diversity in algorithms

- For a given problem, many "ways" to solve it
- Diversity present:
  - Across algorithms
  - In tuning parameters for algorithms (CPLEX for example)
  - Within an algorithm (randomization for example)

Diversity in algorithms

- For a given problem, many "ways" to solve it
- Diversity present:
  - Across algorithms
  - In tuning parameters for algorithms (CPLEX for example)
  - Within an algorithm (randomization for example)
- The choice for the "best" algorithm is not always clear statically or even when the input is known

Diversity across algorithms

Diversity across algorithms

- For hard problems (NP-hard or even very large problems), exact solutions are impractical

Diversity across algorithms

- For hard problems (NP-hard or even very large problems), exact solutions are impractical
- *Acceptable* solutions rather than exact ones are sought

Diversity across algorithms

- For hard problems (NP-hard or even very large problems), exact solutions are impractical
- *Acceptable* solutions rather than exact ones are sought
- Variety of techniques used:
  - Approximation algorithms
  - Heuristics

Diversity across algorithms

- For hard problems (NP-hard or even very large problems), exact solutions are impractical
- *Acceptable* solutions rather than exact ones are sought
- Variety of techniques used:
  - Approximation algorithms
  - Heuristics
- Solutions across algorithms may be different but all are correct

Diversity across algorithms

- For hard problems (NP-hard or even very large problems), exact solutions are impractical
- *Acceptable* solutions rather than exact ones are sought
- Variety of techniques used:
  - Approximation algorithms
  - Heuristics
- Solutions across algorithms may be different but all are correct
- Examples: SAT solvers, path finding algorithms, ...

Diversity within an algorithm

Diversity within an algorithm

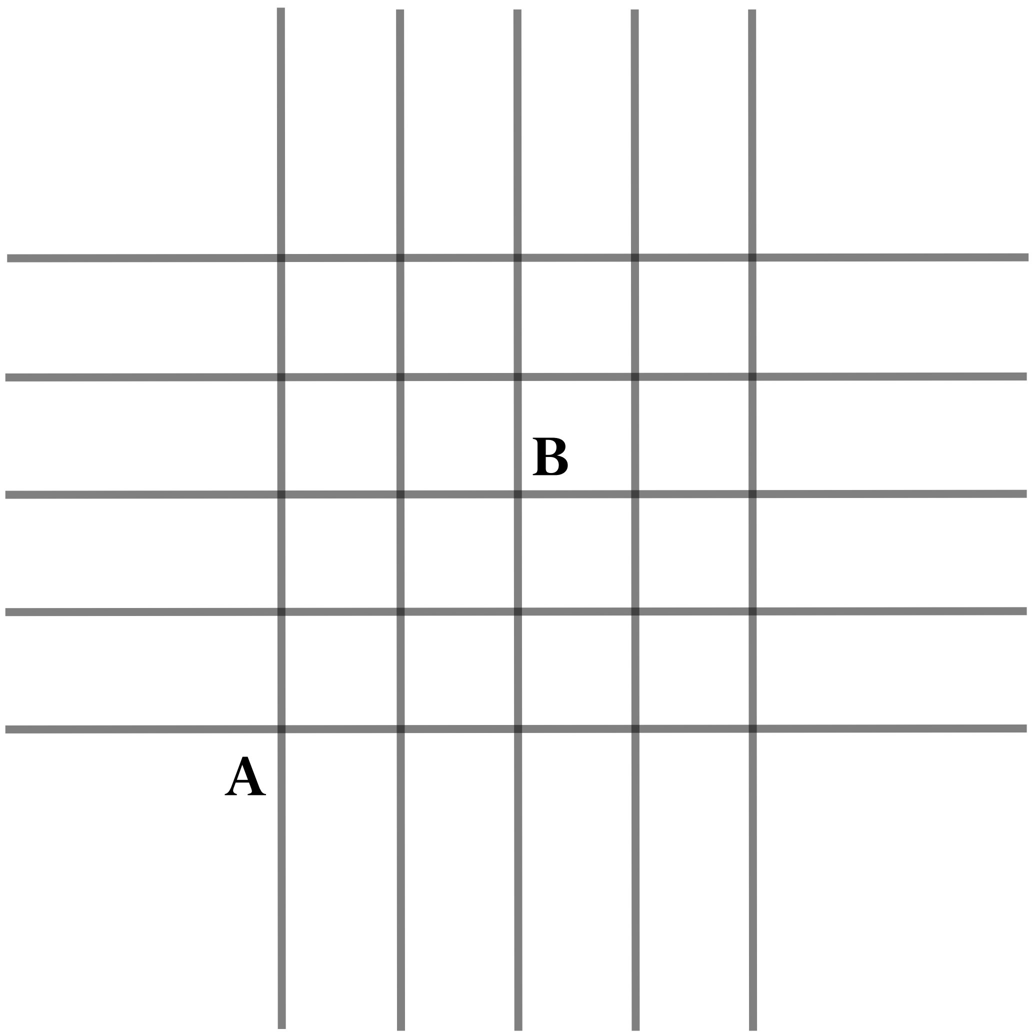
- Certain algorithms are also intrinsically diverse:
  - Distinct executions will lead to distinct execution paths and possibly distinct solutions

Diversity within an algorithm

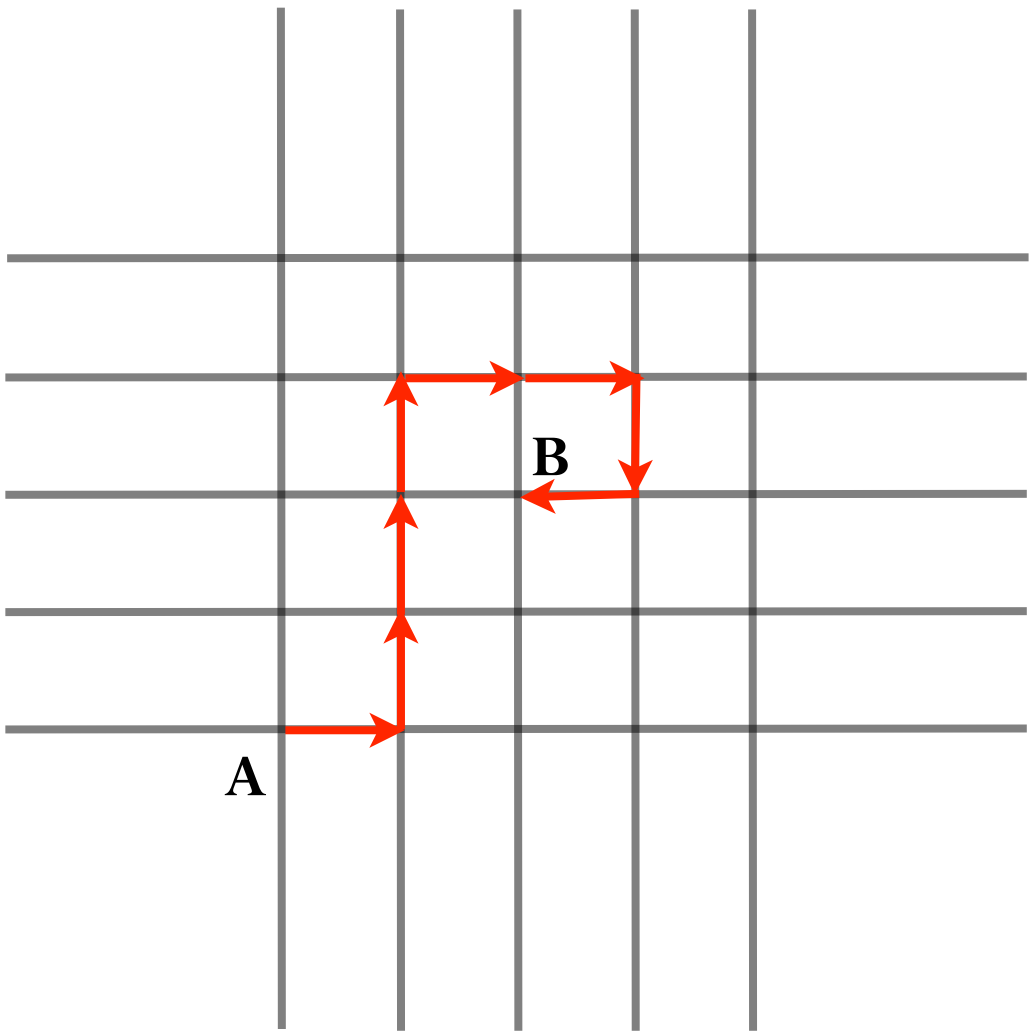
- Certain algorithms are also intrinsically diverse:
  - Distinct executions will lead to distinct execution paths and possibly distinct solutions
- Intuition: randomized algorithms
  - Utilized for their simplicity and quicker execution
  - Random choices lead to different executions each time

Example: random walk

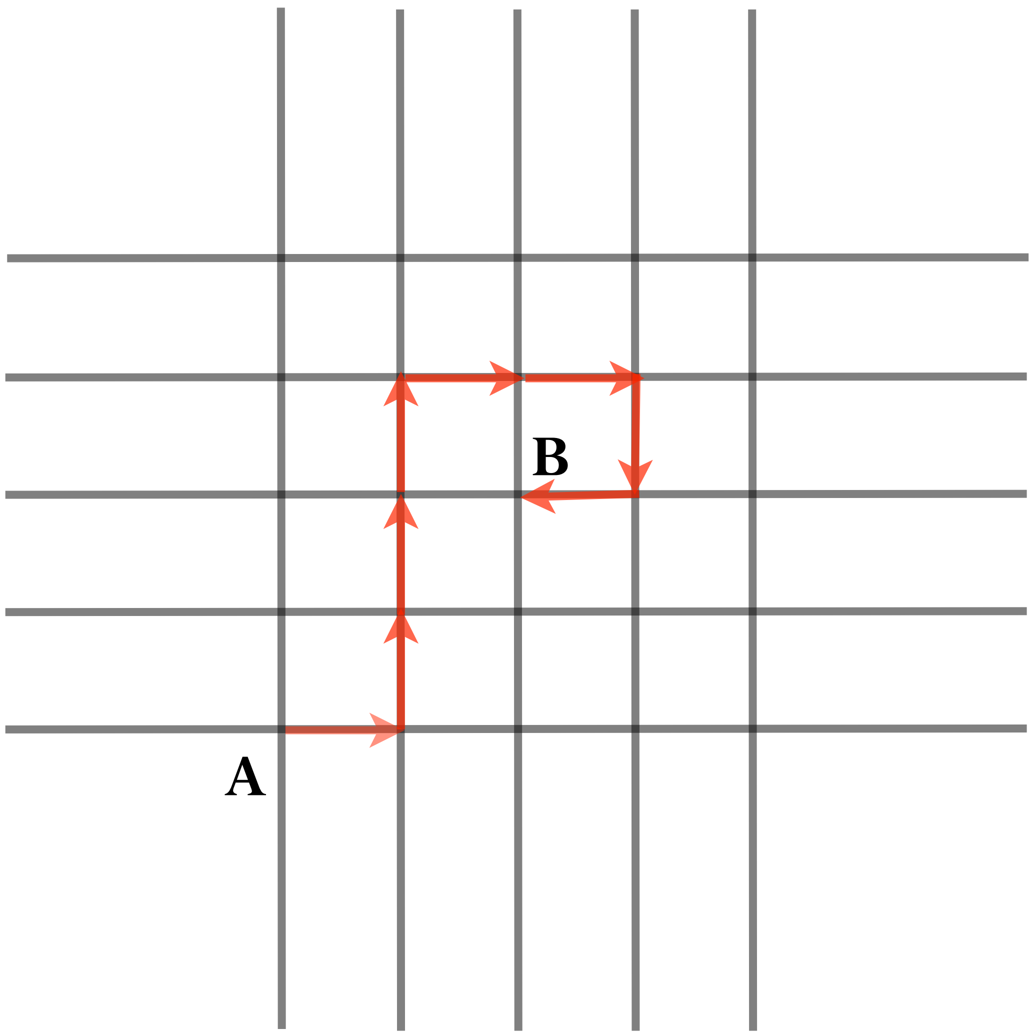
Example: random walk



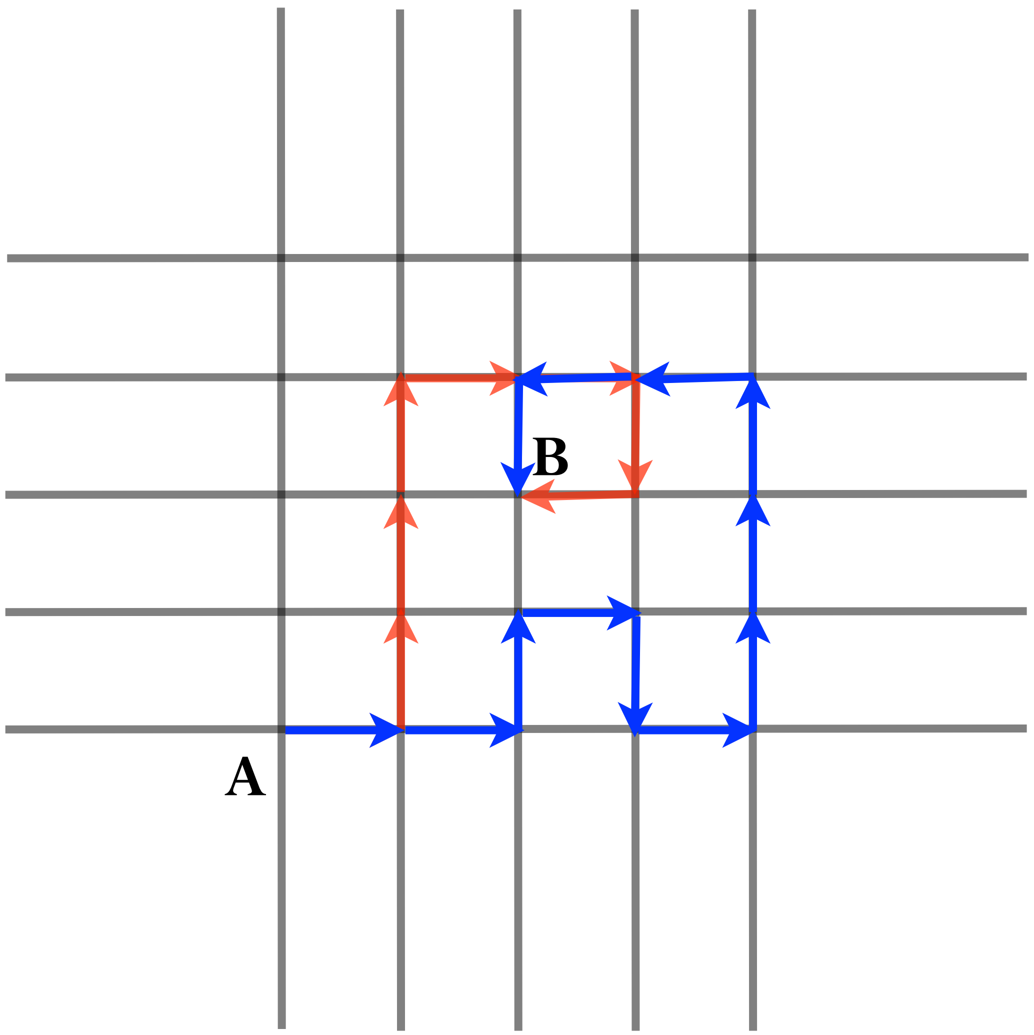
Example: random walk



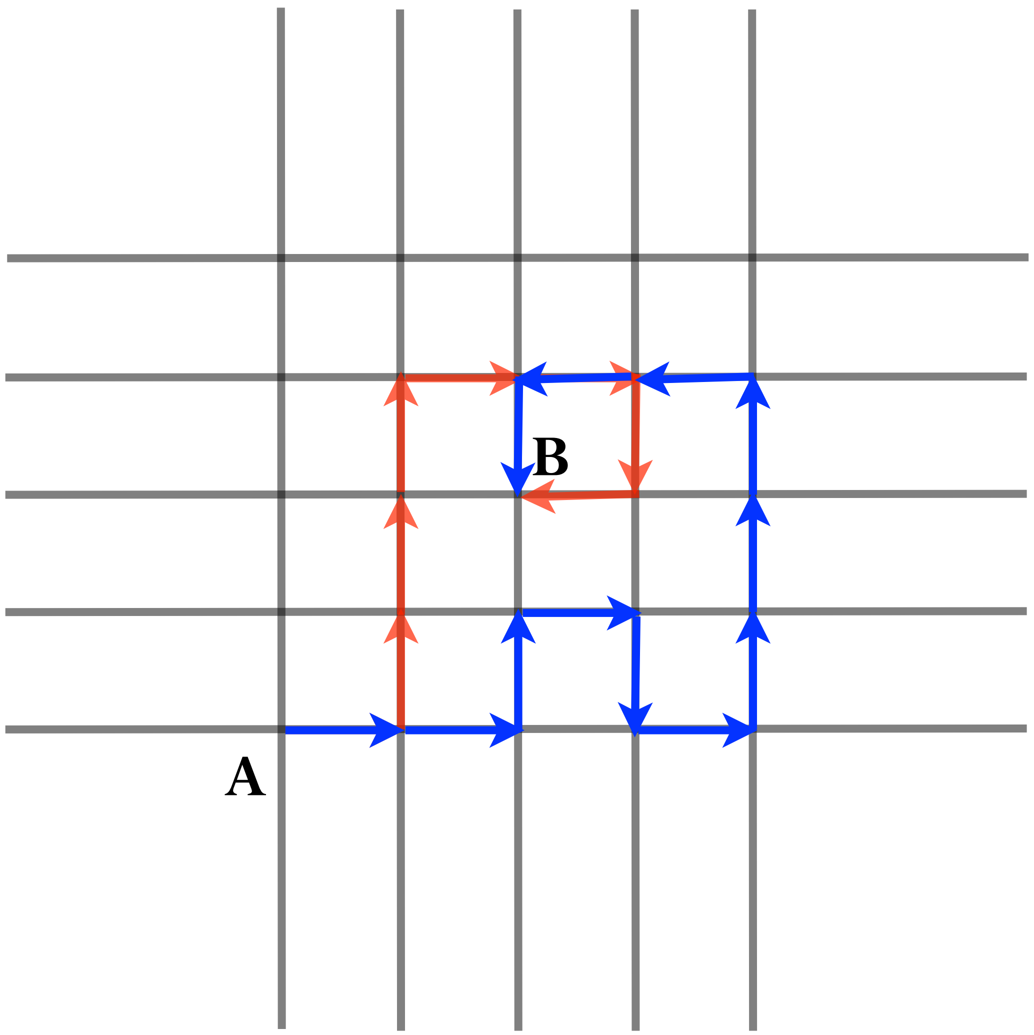
Example: random walk



Example: random walk

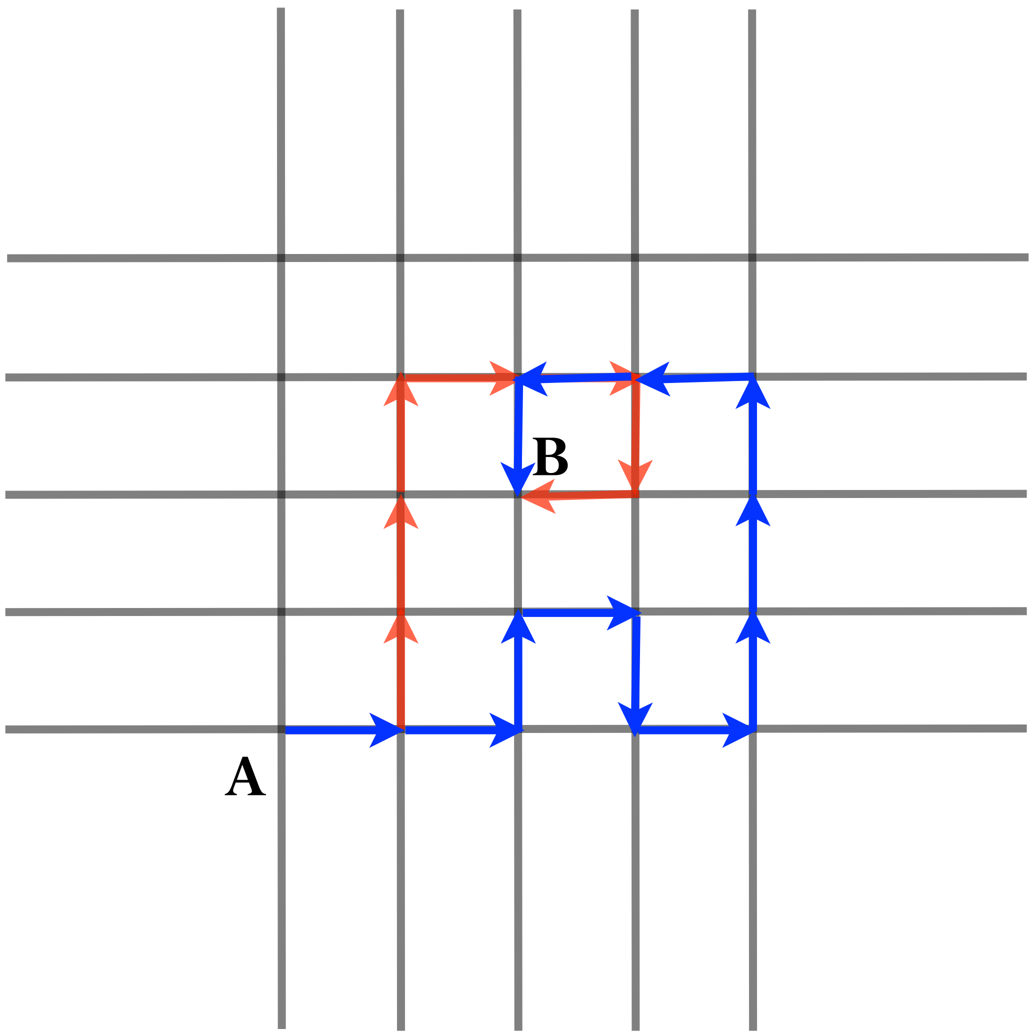


Example: random walk



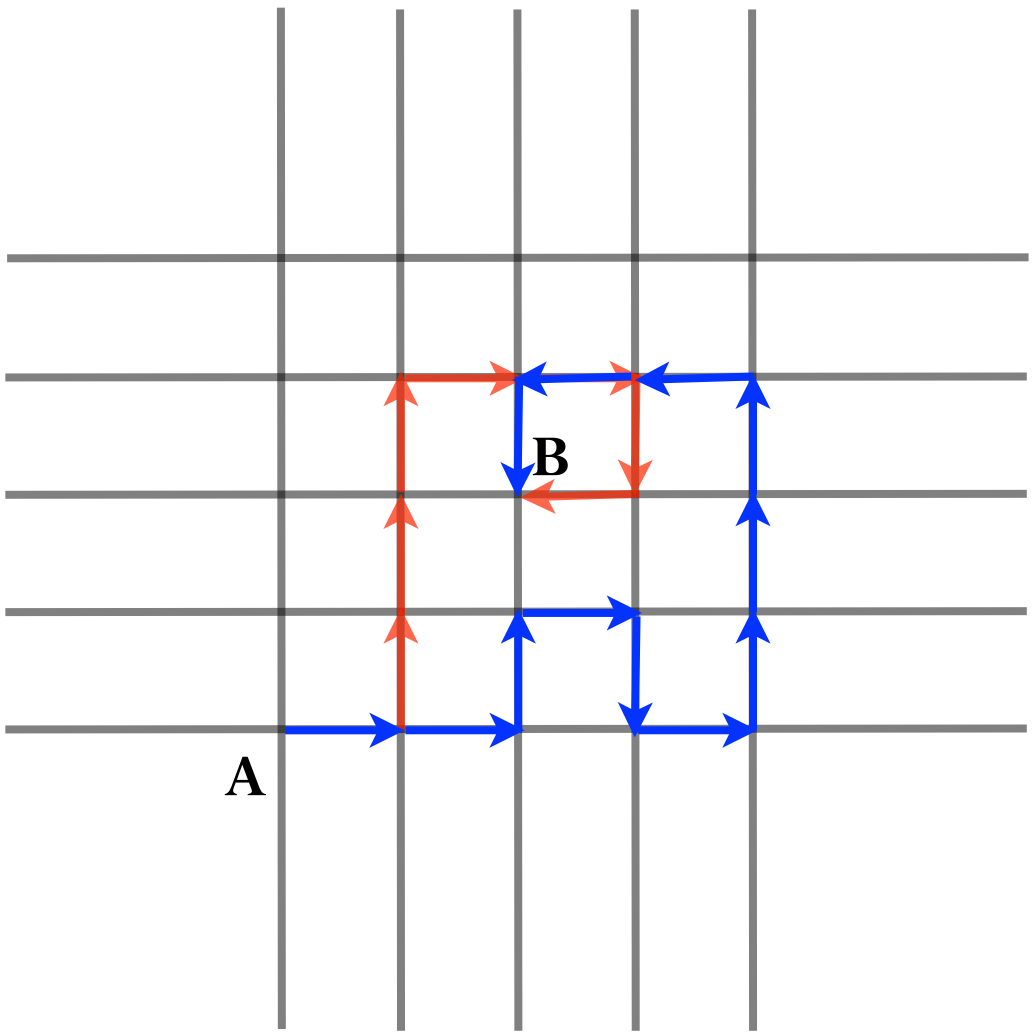
- Differences in:
  - Memory accesses
  - Total compute steps

Example: random walk



- Differences in:
  - Memory accesses
  - Total compute steps
- Energy needs will be different

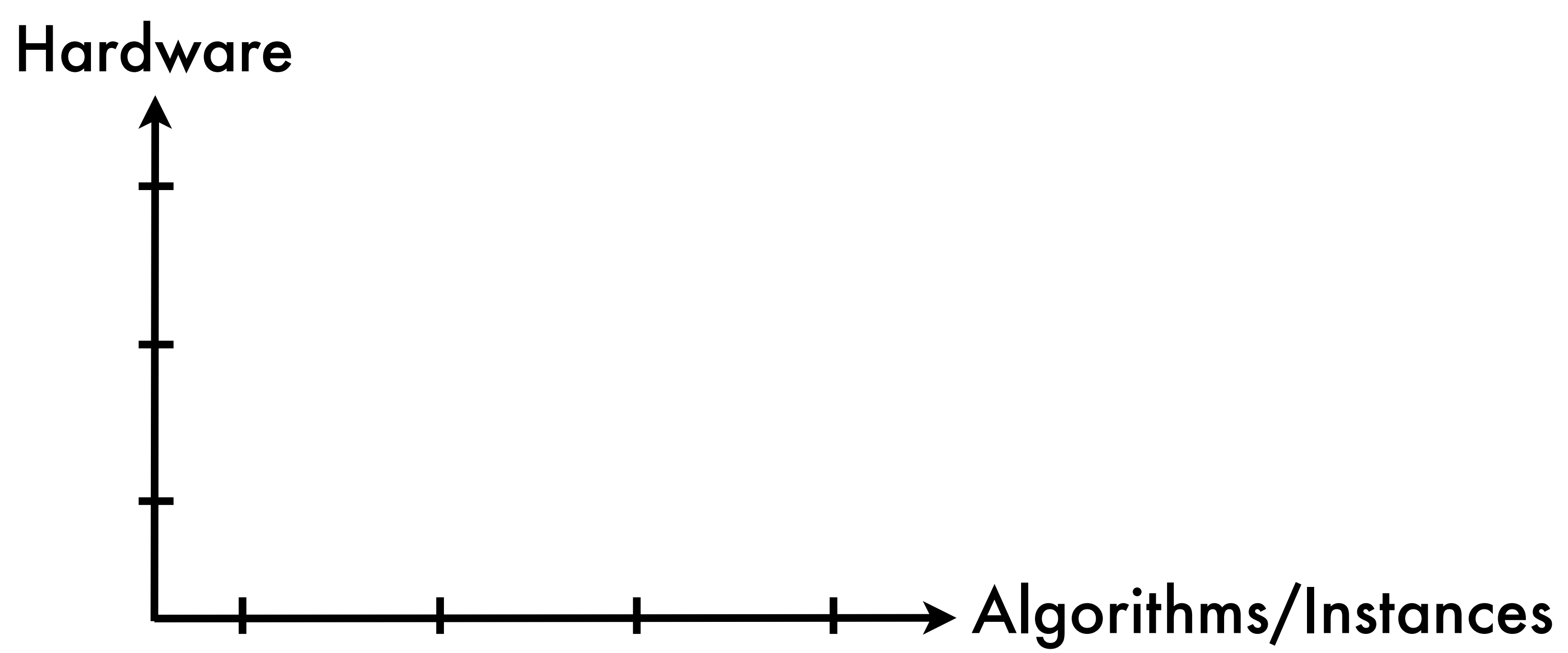
Example: random walk



- Differences in:
  - Memory accesses
  - Total compute steps
- Energy needs will be different
- Highly unpredictable

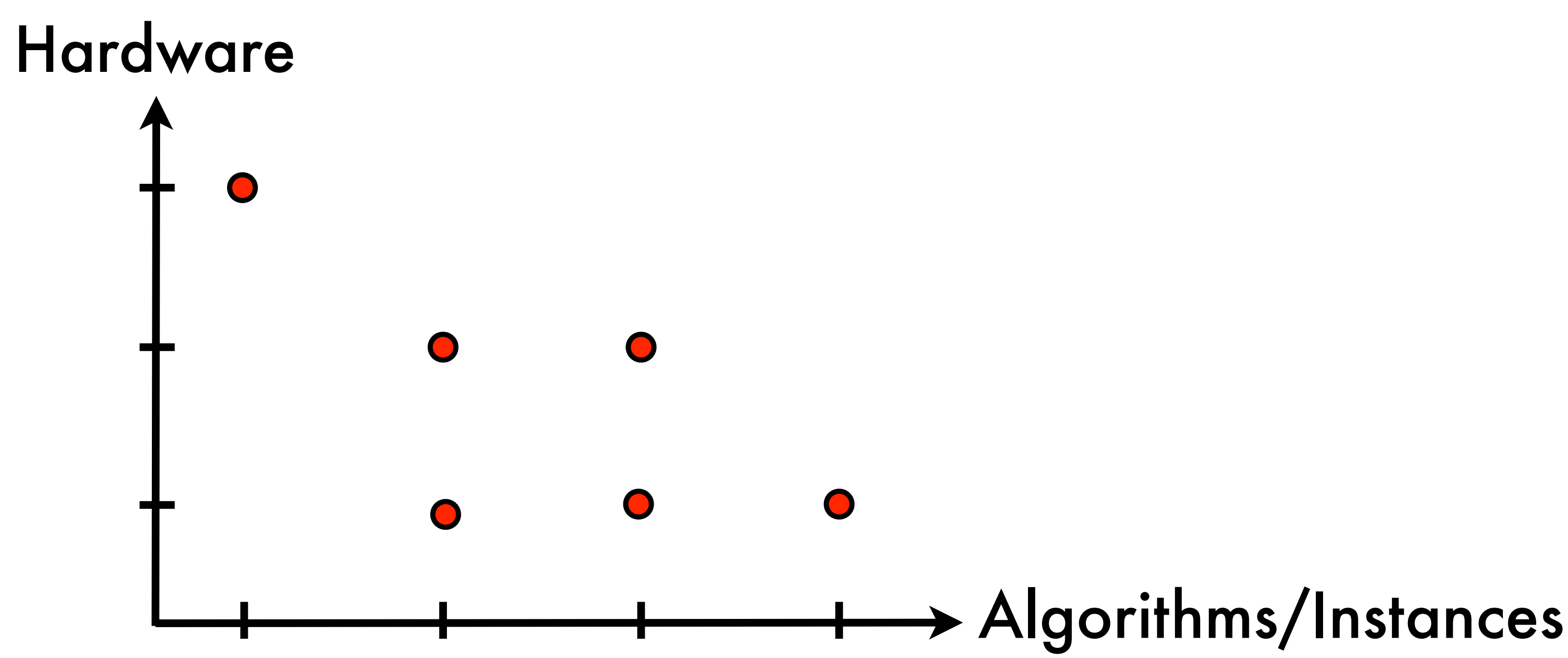
Problem statement

Problem statement



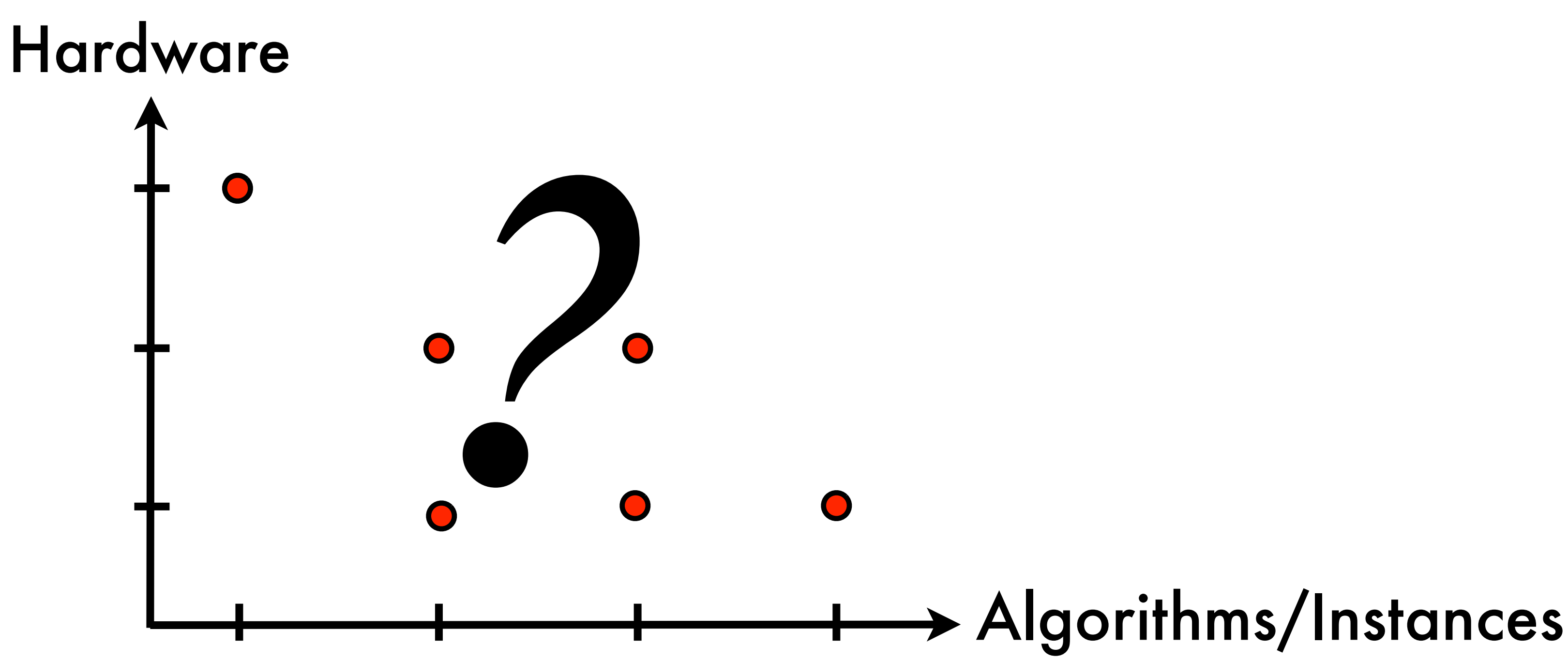
Problem statement

- Diversity creates the exploration space



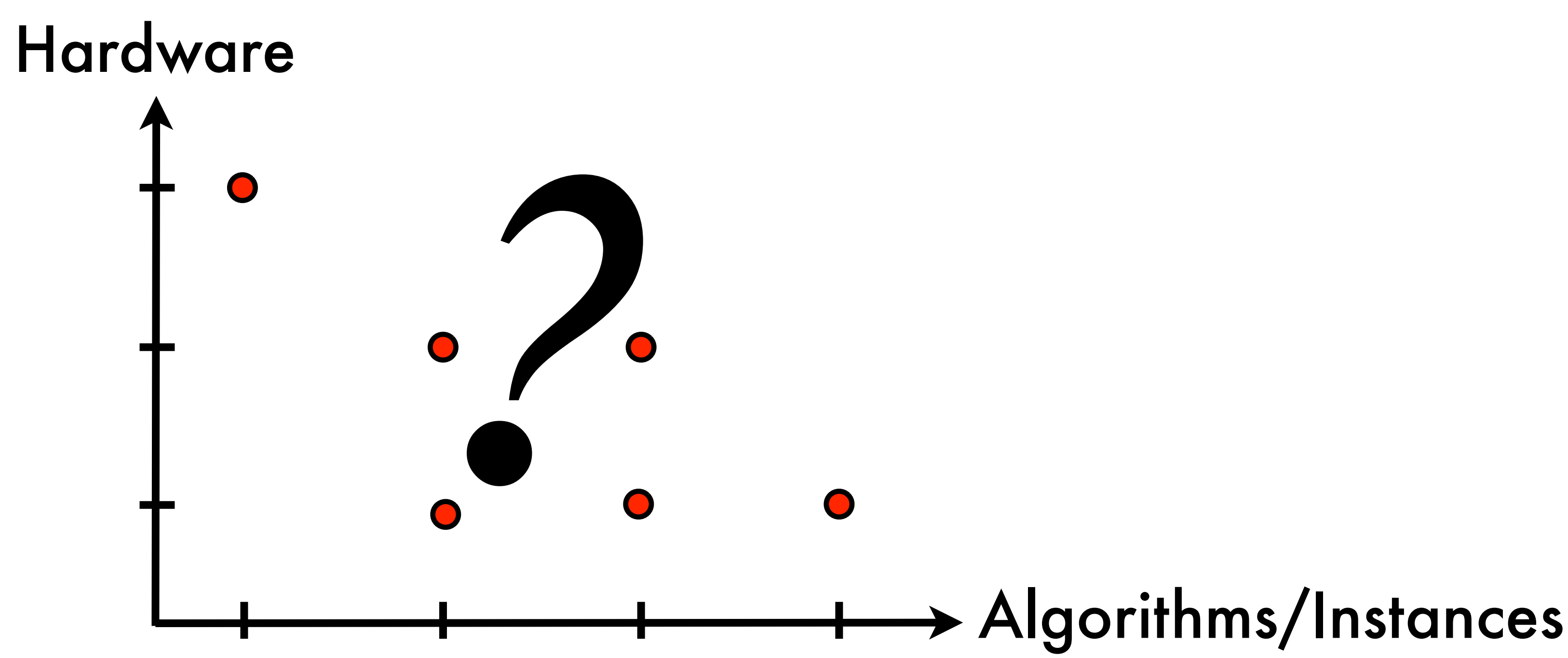
Problem statement

- Diversity creates the exploration space
- We want to utilize multi-cores to dynamically determine the best platform/algorithm match for a given input



Problem statement

- Diversity creates the exploration space
- We want to utilize multi-cores to dynamically determine the best platform/algorithm match for a given input
- Multi-cores allow this exploration to be done without any assumptions
  - Just-in-time decision



Outline

- The challenges of sequential code on multi-cores
- Using diversity to expose parallelism
- Measures of energy efficiency
  - Progress measure
  - Power measure
- The n-way programming model
- Preliminary results and future work

Energy efficiency

Energy efficiency

- *Efficient*: achieving maximum productivity with minimum wasted effort

Energy efficiency

- *Efficient*: achieving maximum productivity with minimum wasted effort
- For an algorithm:
  - The “product” is the solution to the problem
  - The “effort” is the energy/resources used

Energy efficiency

- *Efficient*: achieving maximum productivity with minimum wasted effort
- For an algorithm:
  - The "product" is the solution to the problem
  - The "effort" is the energy/resources used
- An algorithm is therefore energy efficient if it makes the most progress per unit of energy used

Progress measure

Progress measure

- *Progress* is the amount of work done towards finding the solution to the problem:
    - Defined in terms of the problem, not the algorithm
    - Very problem dependent

Progress measure

- *Progress* is the amount of work done towards finding the solution to the problem:
  - Defined in terms of the problem, not the algorithm
  - Very problem dependent
- Progress needs to be:
  - Monotonic: cannot undo progress
  - Comparable

Progress examples

Progress examples

- In a “sort” problem
  - Number of elements correctly placed

Progress examples

- In a "sort" problem
  - Number of elements correctly placed
- Greedy constructive algorithms:
  - Natural notion of progress

Progress examples

- In a "sort" problem
  - Number of elements correctly placed
- Greedy constructive algorithms:
  - Natural notion of progress
- Search-based problems (SAT problems):
  - The amount of space explored

Progress examples

- In a "sort" problem
  - Number of elements correctly placed
- Greedy constructive algorithms:
  - Natural notion of progress
- Search-based problems (SAT problems):
  - The amount of space explored
- Not always evident but widely applicable

Energy measure

Energy measure

- Energy is hard to measure in real-time

Energy measure

- Energy is hard to measure in real-time
- Approximate models are required

Energy measure

- Energy is hard to measure in real-time
- Approximate models are required
- Performance monitoring counters (PMCs) can be used:
  - Run micro-benchmarks exercising PMCs
  - Measure power consumption with a Watt meter
  - Correlate PMCs and consumption and build a model

Energy measure

- Energy is hard to measure in real-time
- Approximate models are required
- Performance monitoring counters (PMCs) can be used:
  - Run micro-benchmarks exercising PMCs
  - Measure power consumption with a Watt meter
  - Correlate PMCs and consumption and build a model
- Estimating power in real-time is thus possible

Outline

- The challenges of sequential code on multi-cores
- Using diversity to expose parallelism
- Measures of energy efficiency
  - Progress measure
  - Power measure
- The n-way programming model
- Preliminary results and future work

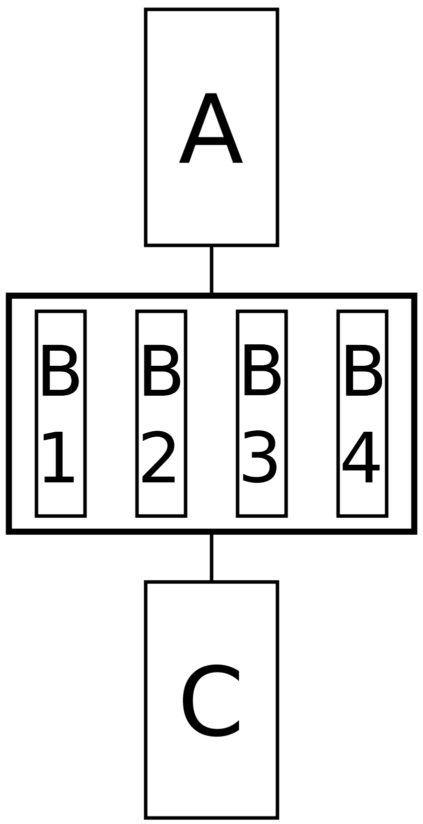
Diversity and the n-way model

Diversity and the n-way model

- The n-way model exploits diversity:
  - To speedup or improve the QoR [HotPar 2009]
  - To find the best match just-in-time [PESPMA 2010]

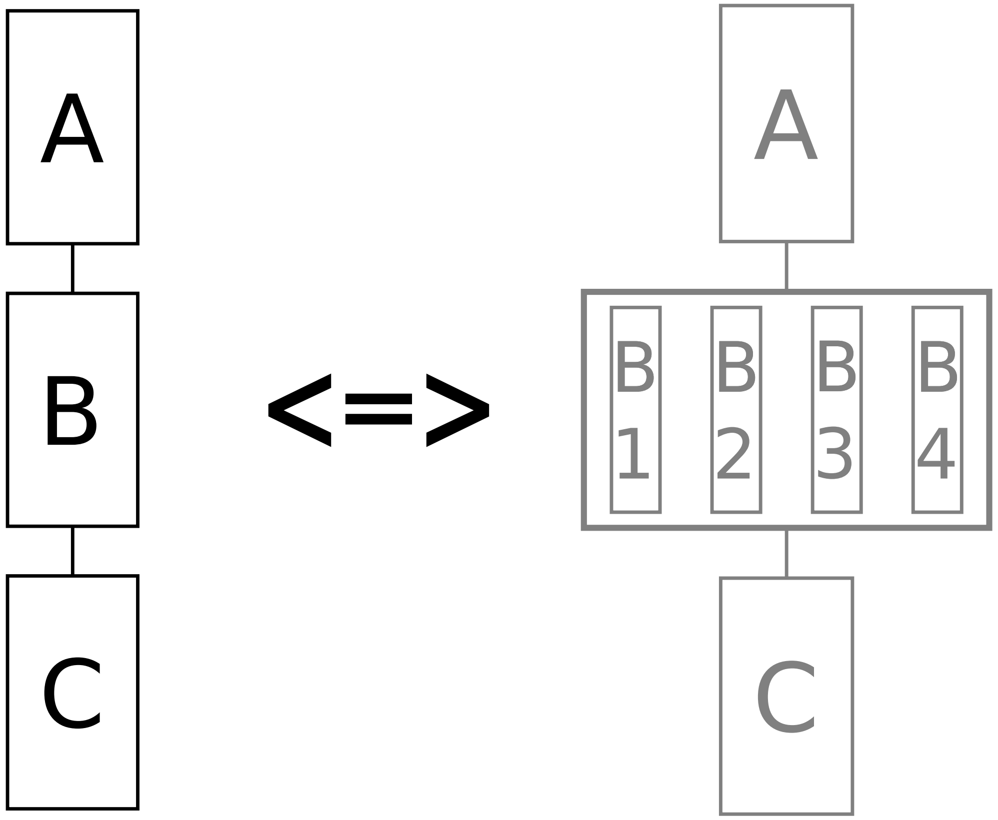
Diversity and the n-way model

- The n-way model exploits diversity:
  - To speedup or improve the QoR [HotPar 2009]
  - To find the best match just-in-time [PESPMA 2010]



Diversity and the n-way model

- The n-way model exploits diversity:
  - To speedup or improve the QoR [HotPar 2009]
  - To find the best match just-in-time [PESPMA 2010]



The n-way model

The n-way model

- **Isolation**
  - Provides semantic equivalence with the sequential execution

The n-way model

- **Isolation**
  - Provides semantic equivalence with the sequential execution
- **Progress monitor**
  - Each way reports back its progress

The n-way model

- **Isolation**
  - Provides semantic equivalence with the sequential execution
- **Progress monitor**
  - Each way reports back its progress
- **Dynamic choice and culling**
  - The most energy efficient way is automatically selected
  - It must be late enough to have gathered enough information
  - But also early enough to save energy

Available API

Available API

- Identify a "problem" and attach "ways"
  - All ways solve the same problem

Available API

- Identify a "problem" and attach "ways"
  - All ways solve the same problem
- Identify a "progress" structure to report back progess

Available API

- Identify a "problem" and attach "ways"
  - All ways solve the same problem
- Identify a "progress" structure to report back progess

```
/* Each way has its own "runWay" function */
void runWay(NVWayMetrics& progressMetric, …) {
    <code for the way>
    <updates progressMetric along the way>
}

/* The main code */
NVGoal myGoal;
myGoal.attach(NVWay(runWay));
<attach more ways>

myGoal.run();
```

Outline

- The challenges of sequential code on multi-cores
- Using diversity to expose parallelism
- Measures of energy efficiency
  - Progress measure
  - Power measure
- The n-way programming model
- Preliminary results and future work

Experimental setup

Experimental setup

- Evaluated idea with different implementations of the "sort" benchmark
  - Chosen for illustration purposes
  - Progress is the "number of moves"

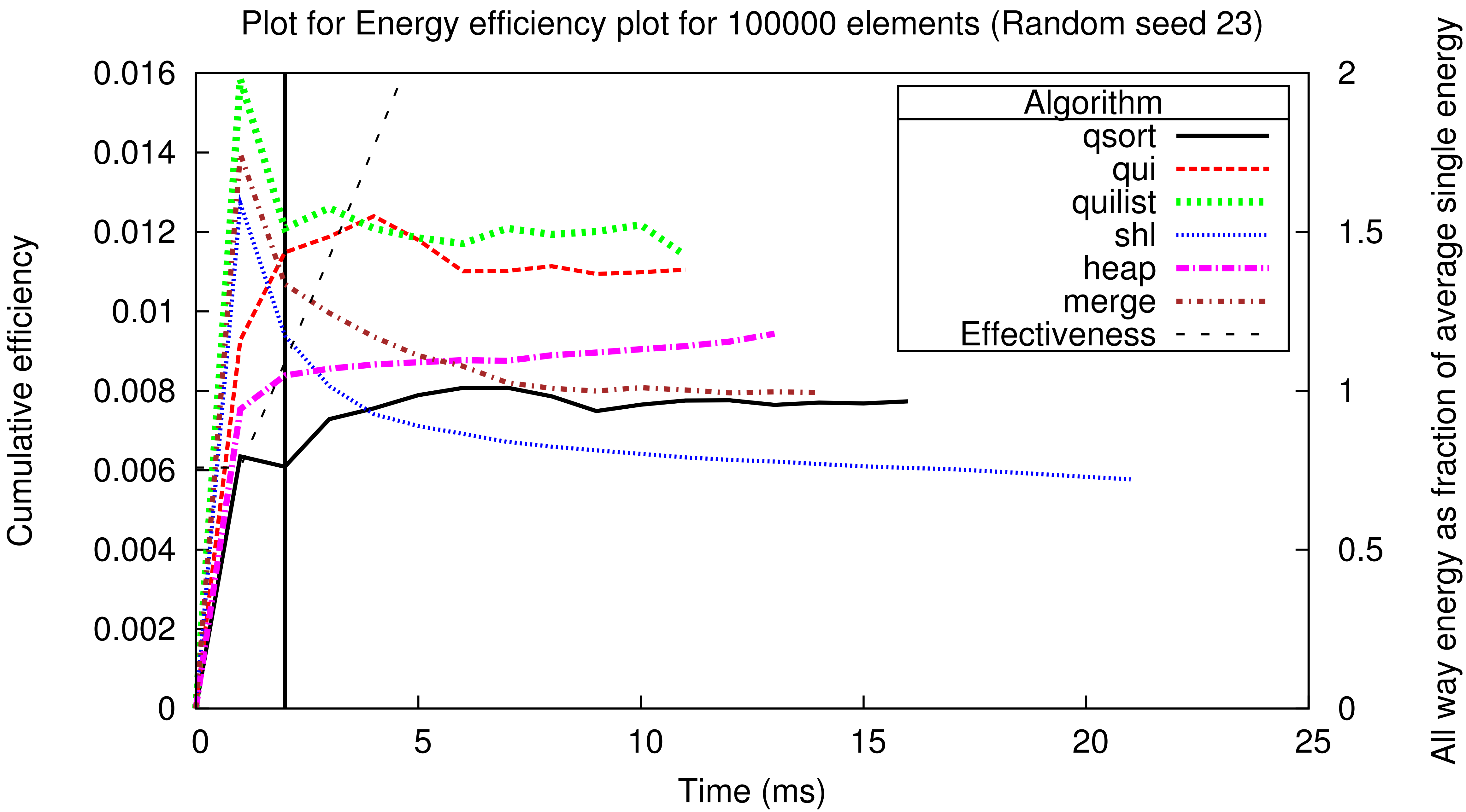
Experimental setup

- Evaluated idea with different implementations of the "sort" benchmark
  - Chosen for illustration purposes
  - Progress is the "number of moves"
- Energy model based on K. Singh's model [ICS 2009]

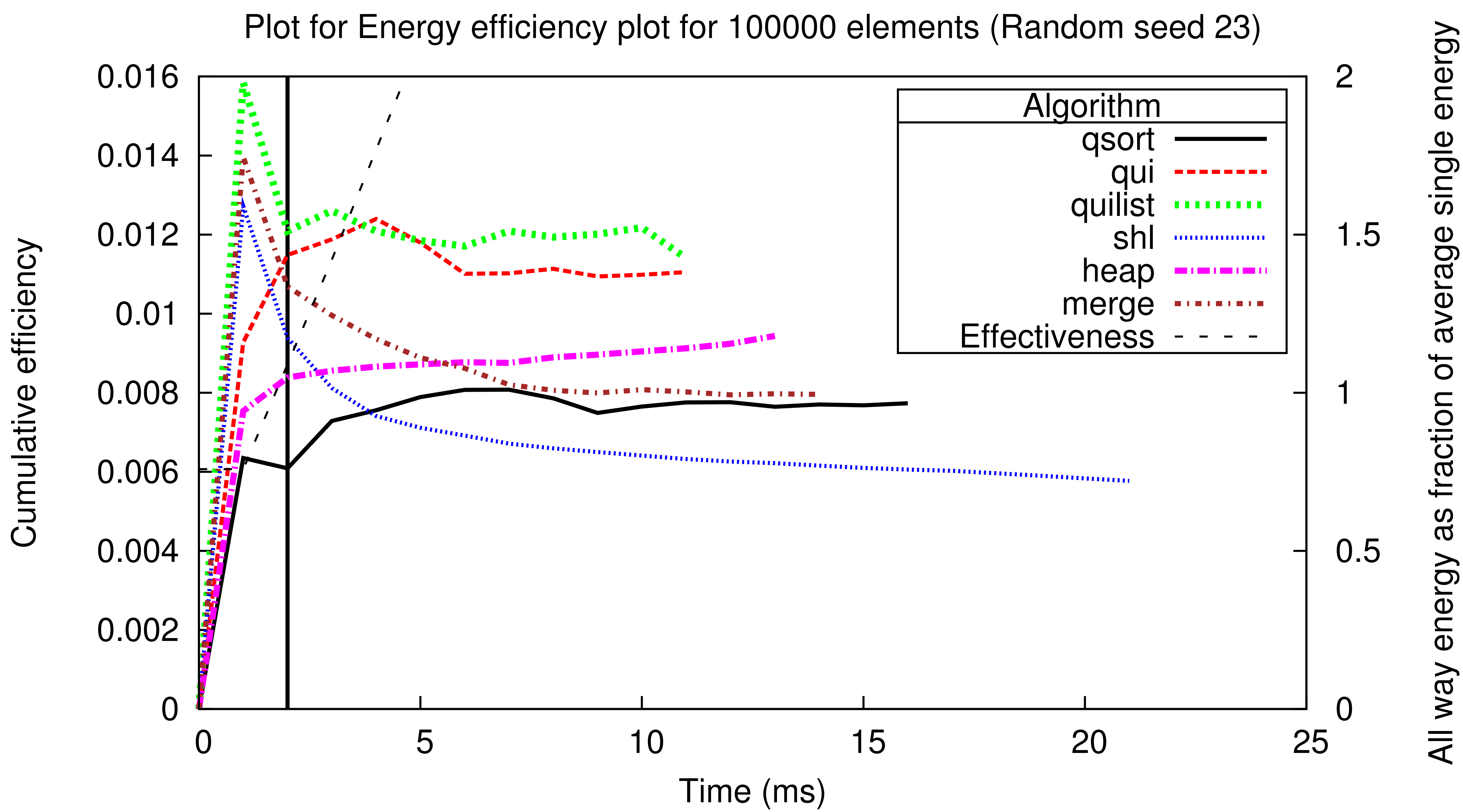
Experimental setup

- Evaluated idea with different implementations of the "sort" benchmark
  - Chosen for illustration purposes
  - Progress is the "number of moves"
- Energy model based on K. Singh's model [ICS 2009]
- Used an Intel Core 2 Duo Q6700 with 2 GB of RAM
  - Patched 2.6.26 Linux kernel for PMC monitoring with PAPI

Preliminary results

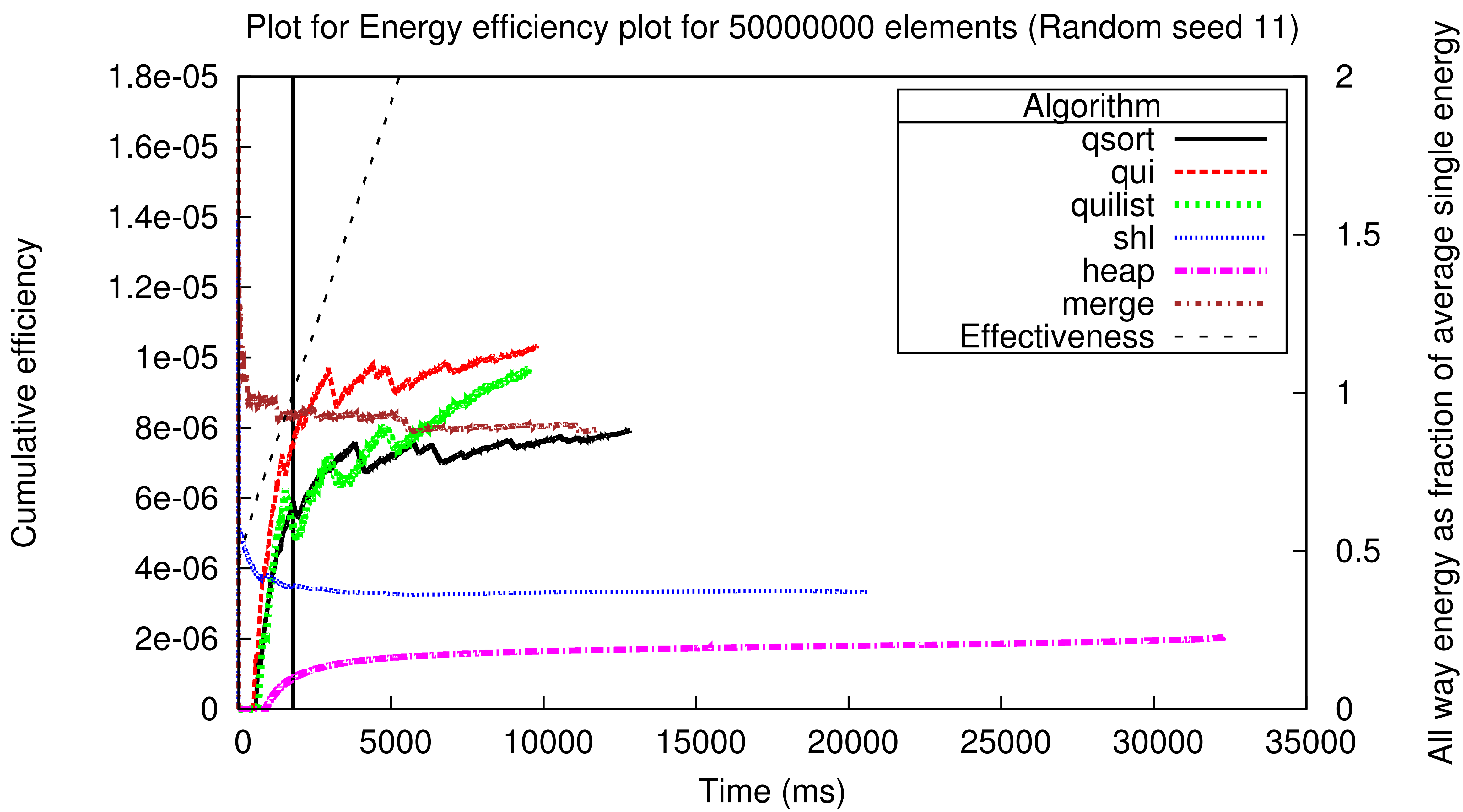


Preliminary results

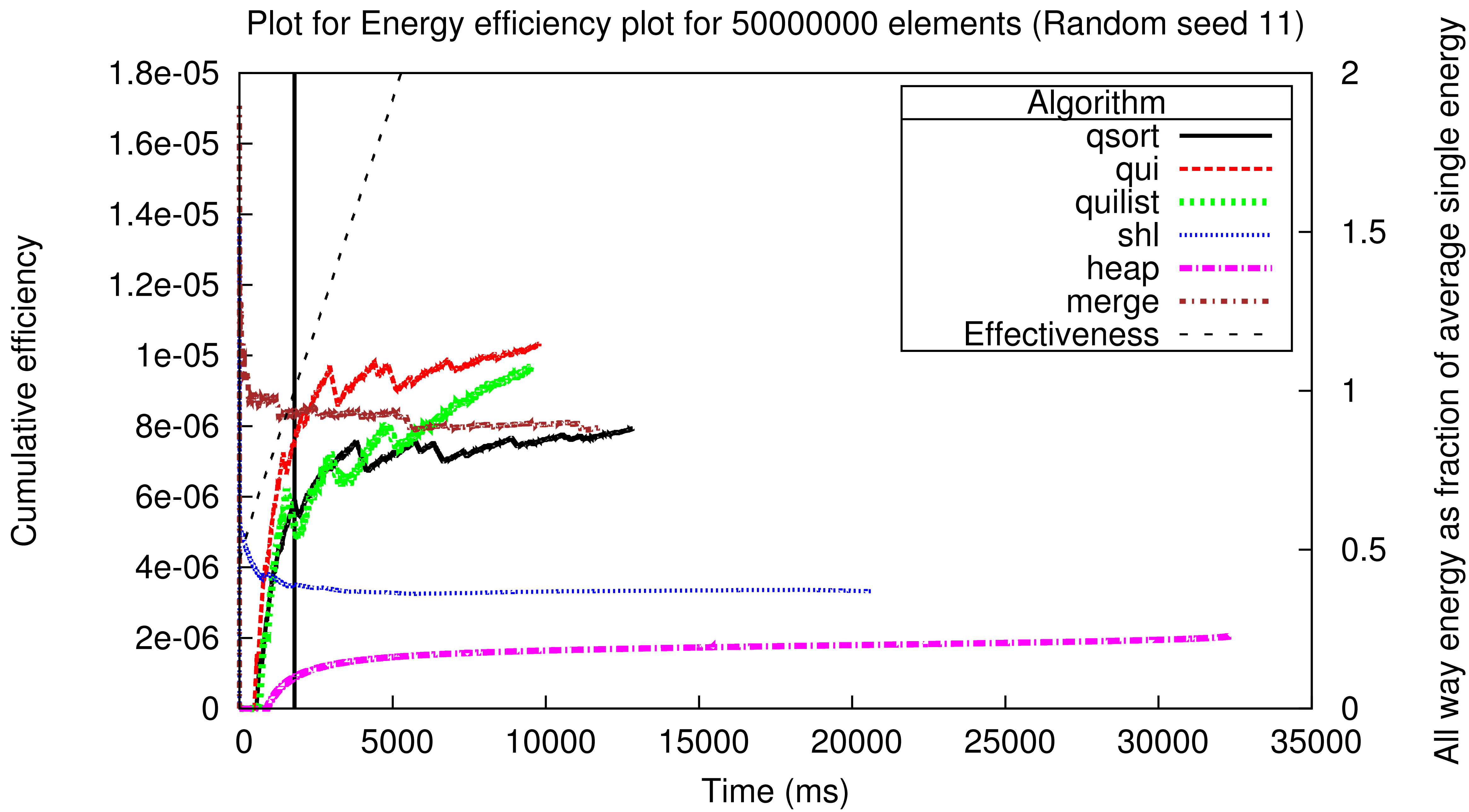


- Pick with good effectiveness possible

Preliminary results



Preliminary results



- "Profiling" becomes required because useful work is not equally distributed

Challenges

Challenges

- Non-uniform energy efficiency
  - Different algorithms have different *usage profile*
  - Possible to pre-profile applications and apply corrective factor

Challenges

- Non-uniform energy efficiency
  - Different algorithms have different *usage profile*
  - Possible to pre-profile applications and apply corrective factor
- Accurate monitoring
  - Energy needs to be accurately modeled
  - Challenges with shared resources

Conclusion

Conclusion

- Introduced the notion of *algorithmic energy efficiency*
  - Should be managed vertically (problem, algorithm, implementation, architecture) for each input
  - Combines algorithmic progress and energy

Conclusion

- Introduced the notion of *algorithmic energy efficiency*
  - Should be managed vertically (problem, algorithm, implementation, architecture) for each input
  - Combines algorithmic progress and energy
- N-way provides a model to utilize multi-cores and exploit diversity to pick the most energy efficient match

Conclusion

- Introduced the notion of *algorithmic energy efficiency*
  - Should be managed vertically (problem, algorithm, implementation, architecture) for each input
  - Combines algorithmic progress and energy
- N-way provides a model to utilize multi-cores and exploit diversity to pick the most energy efficient match
- Progress metric is important to define (monotonic, etc)

Conclusion

- Introduced the notion of *algorithmic energy efficiency*
  - Should be managed vertically (problem, algorithm, implementation, architecture) for each input
  - Combines algorithmic progress and energy
- N-way provides a model to utilize multi-cores and exploit diversity to pick the most energy efficient match
- Progress metric is important to define (monotonic, etc)
- Burst of energy use allows longer period when unused resources can be turned off

Conclusion

- Introduced the notion of *algorithmic energy efficiency*
  - Should be managed vertically (problem, algorithm, implementation, architecture) for each input
  - Combines algorithmic progress and energy
- N-way provides a model to utilize multi-cores and exploit diversity to pick the most energy efficient match
- Progress metric is important to define (monotonic, etc)
- Burst of energy use allows longer period when unused resources can be turned off
- Currently works with algorithms with similar "phases"
  - Looking for input on this and "power-models" for machine and algorithms

Thank you!

Reference: *Opportunistic Computing: a new paradigm for scalable realism on many cores*
R. Cledat, T. Kumar, J. Sreeram, S. Pande [HotPar 2009]