Use of a Levy Distribution for Modeling Best Case Execution Time Variation

Jonathan Beard, Roger Chamberlain

SBS
Stream Based Supercomputing Lab
http://sbs.wustl.edu

Washington University in St. Louis

Work also supported by:

exegy

Outline

- **Motivation**
  - **Stream Processing**
  - **Optimization Goals**
- **Methodology**
- **Distributions**
- **Results**

Streaming Computing

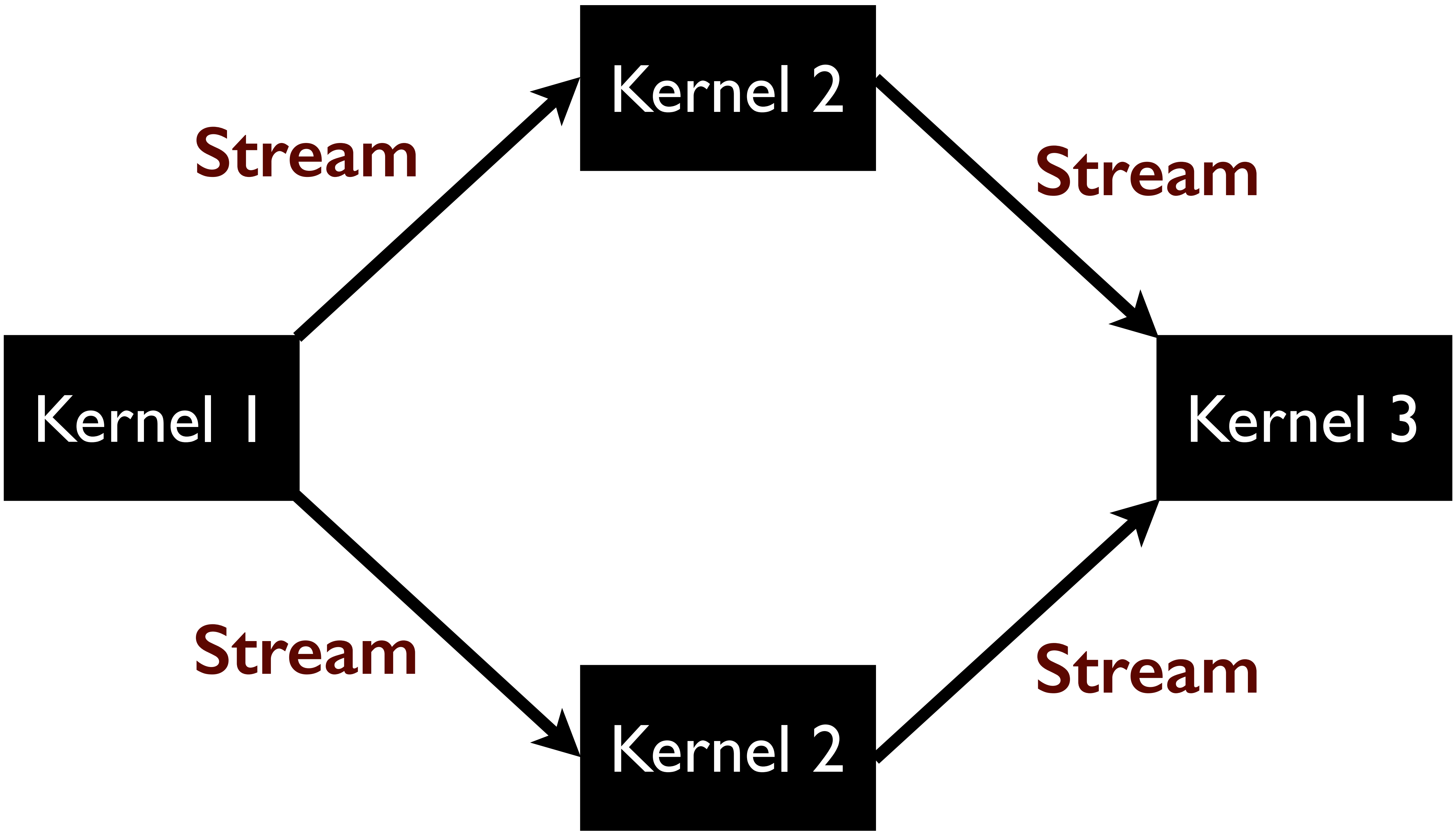
```
streams [[ Output ]] Work( InputOne, InputTwo )
{
      X =  InputOne.get( );
      Y =  InputTwo.get( );
      out = do_something( X, Y );
      Output.push( out );
}
```

Streaming Computing

```
streams [[ Output ]] Work( InputOne, InputTwo )
{
      X =  InputOn
      Y =  InputTw
      out = do_som                 );
      Output.push( out );
}
```

Kernel

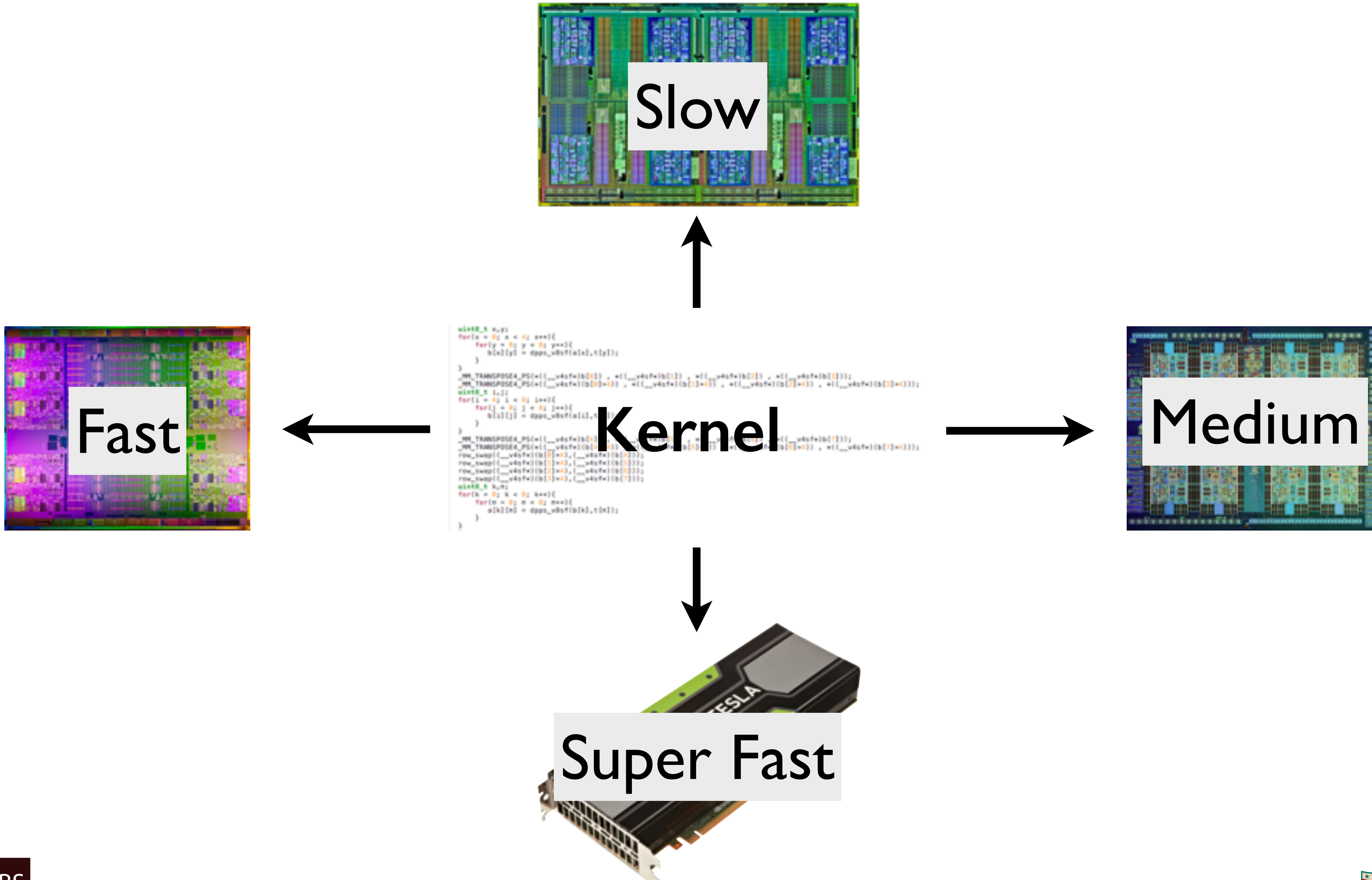
Streaming Computing



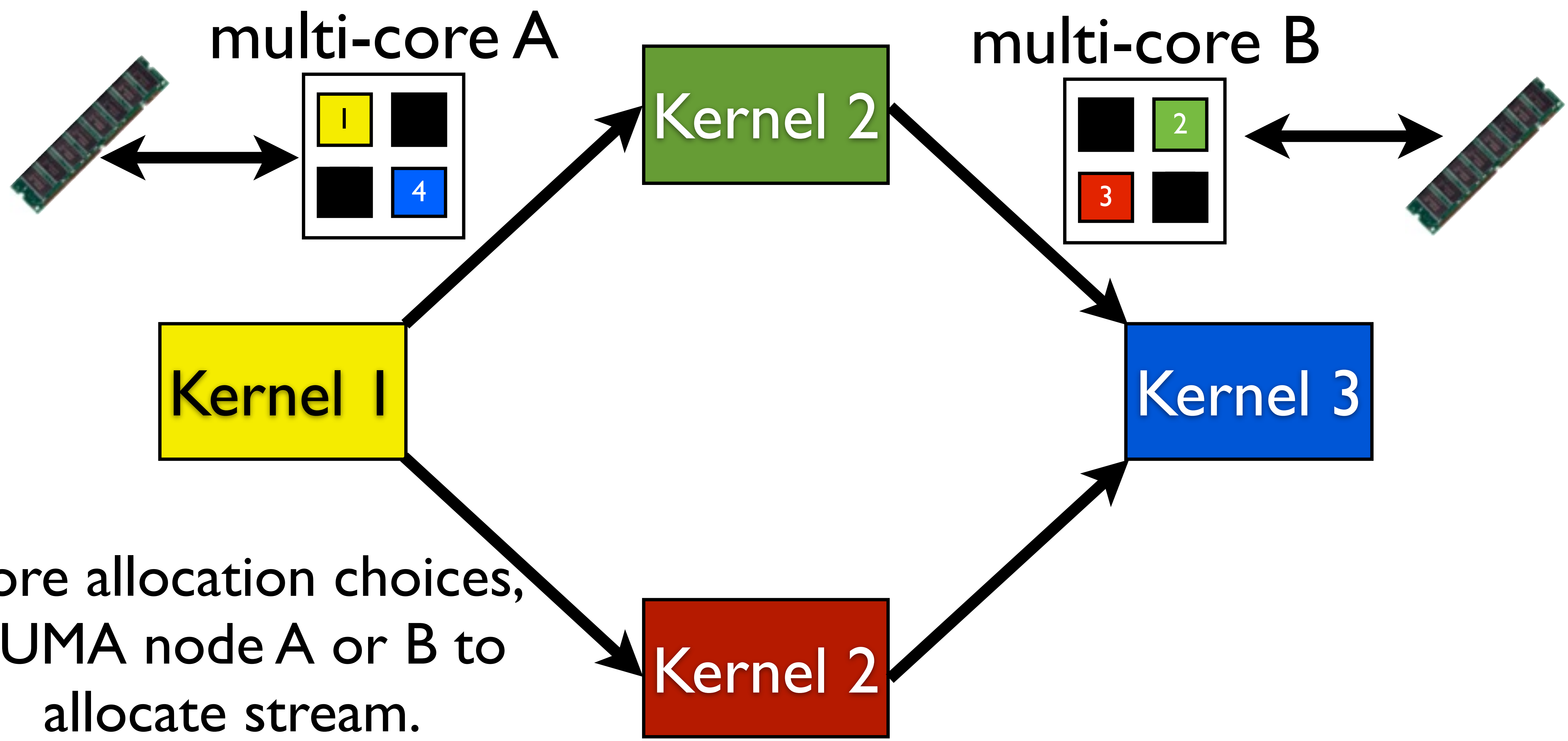
Streaming Languages

StreamIt, Auto-Pipe, Brook, Cg, S-Net, Scala-Pipe, Streams-C and many others

Optimization

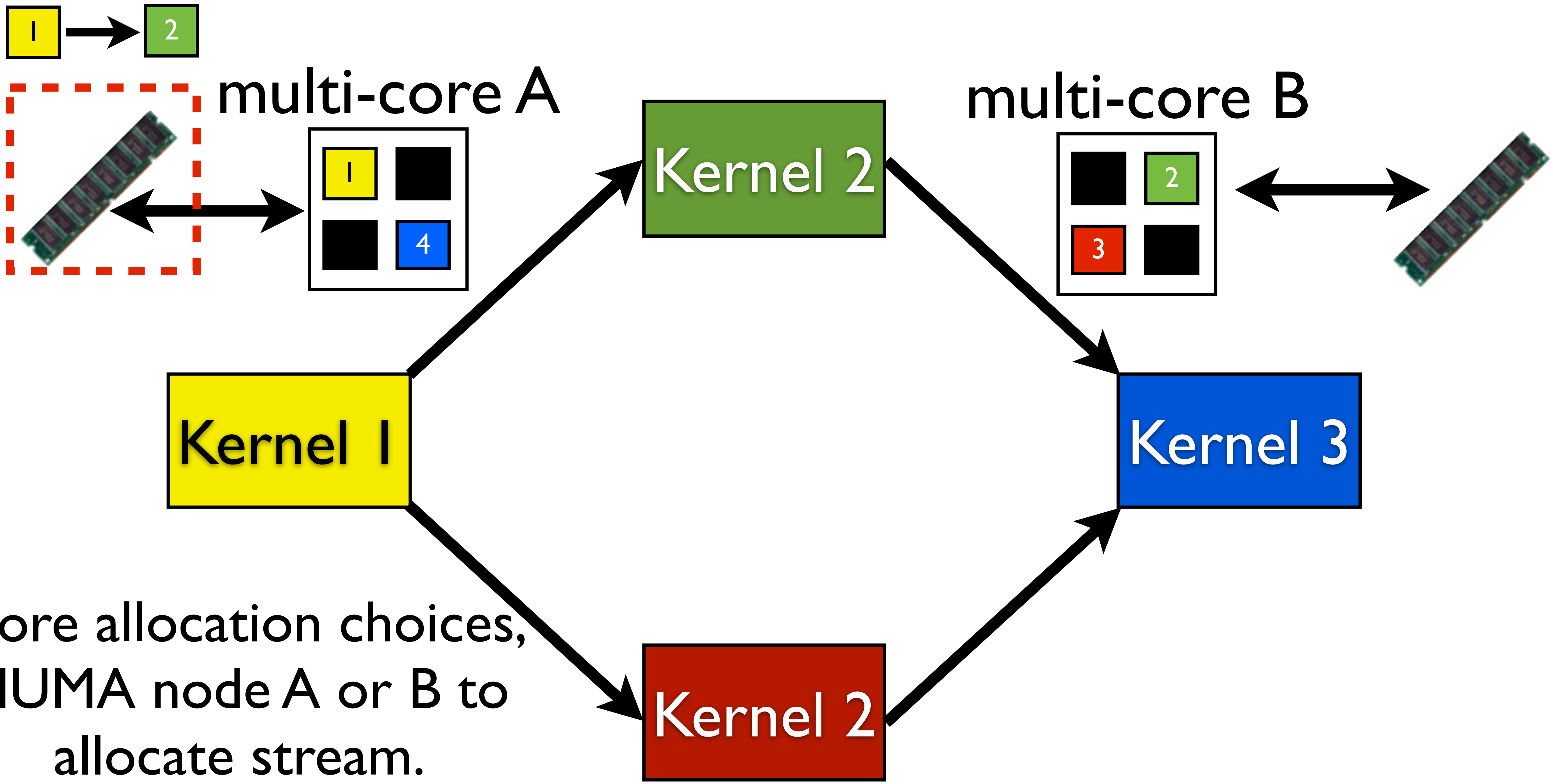


Optimization



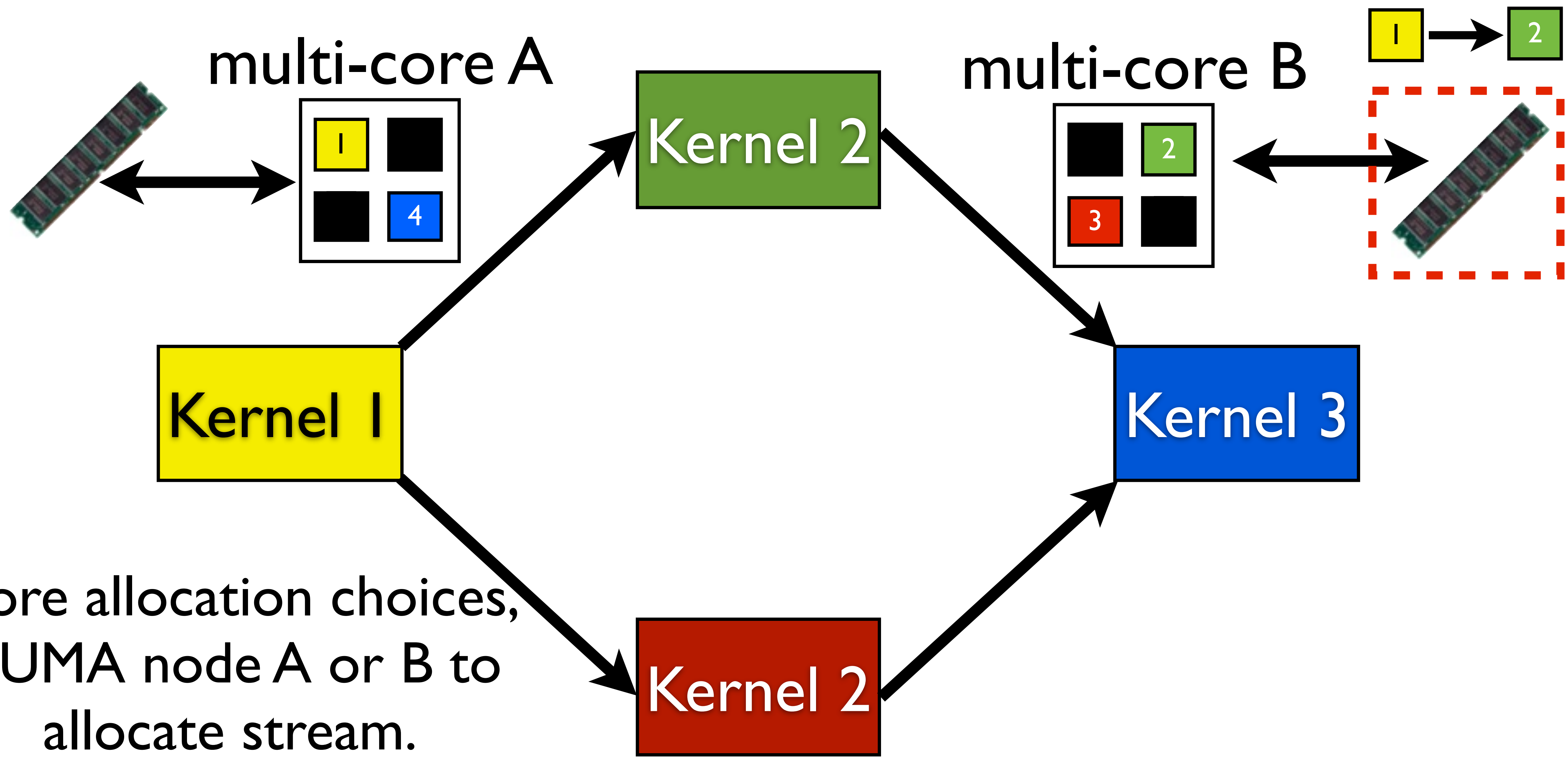
More allocation choices, NUMA node A or B to allocate stream.

Optimization



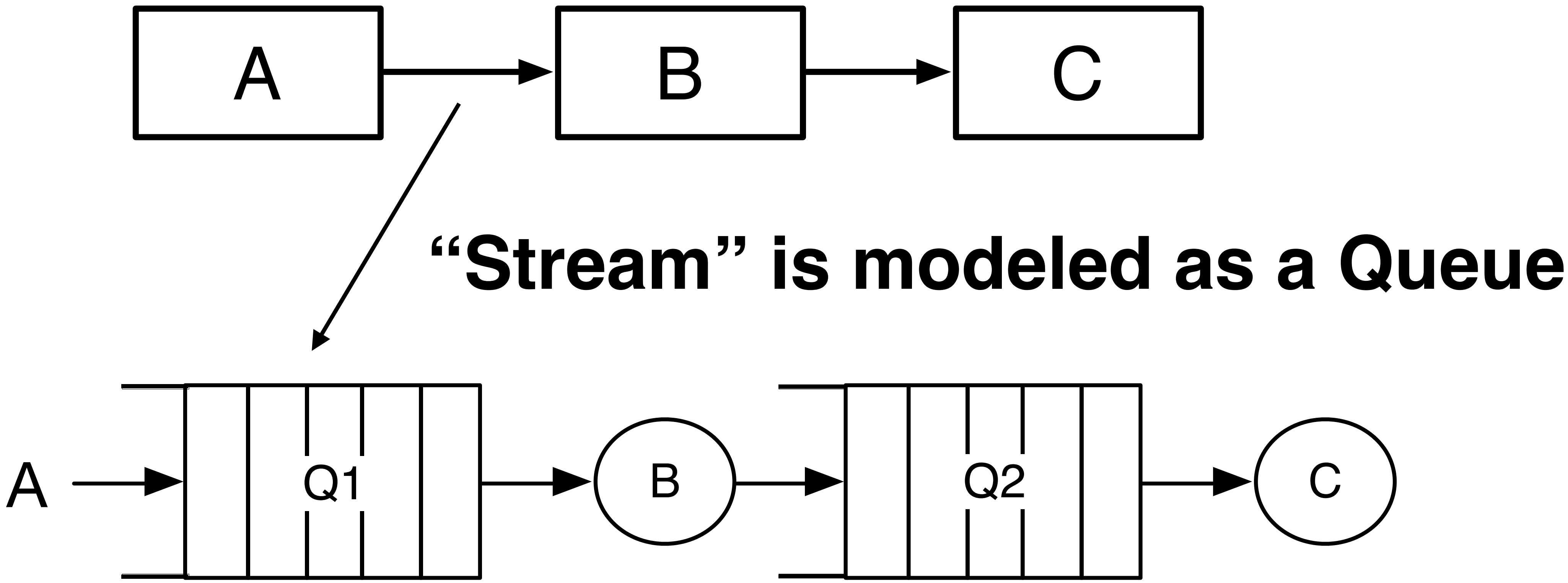
More allocation choices, NUMA node A or B to allocate stream.

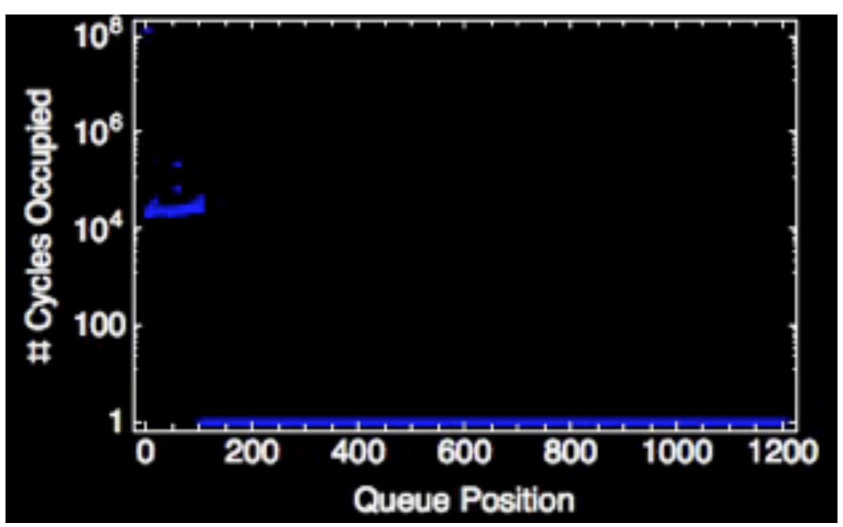
Optimization

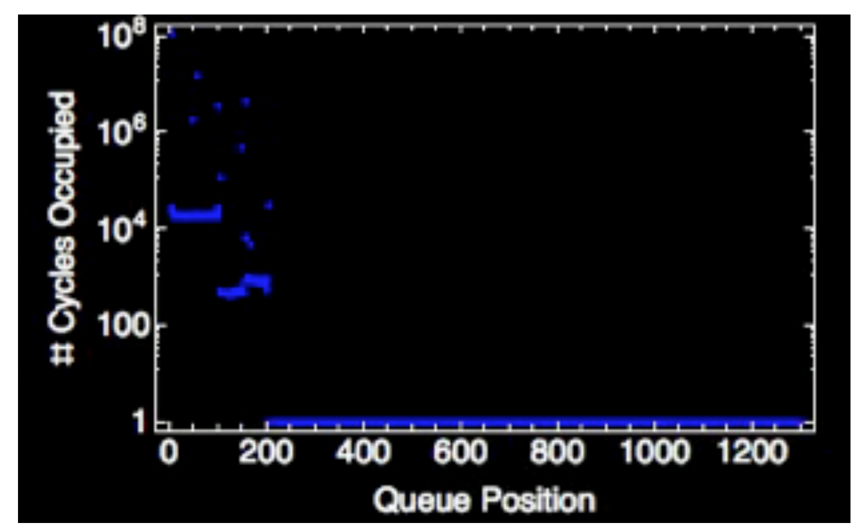


More allocation choices, NUMA node A or B to allocate stream.

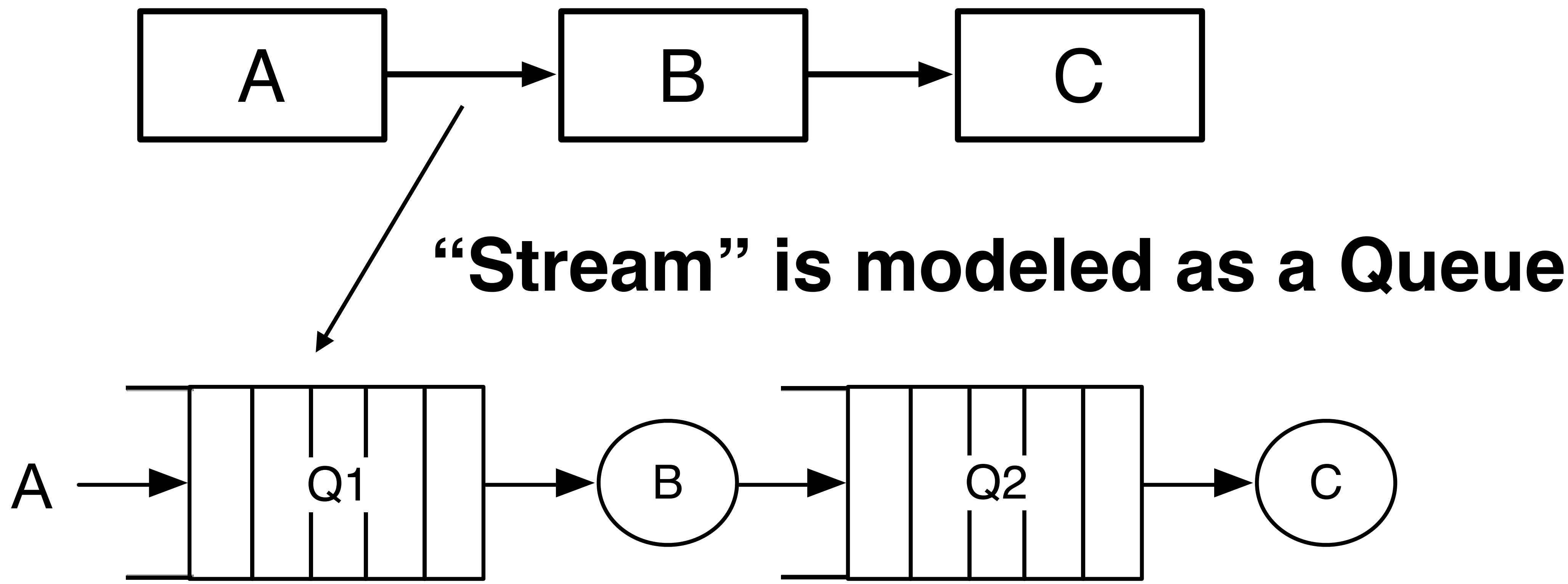
Optimization

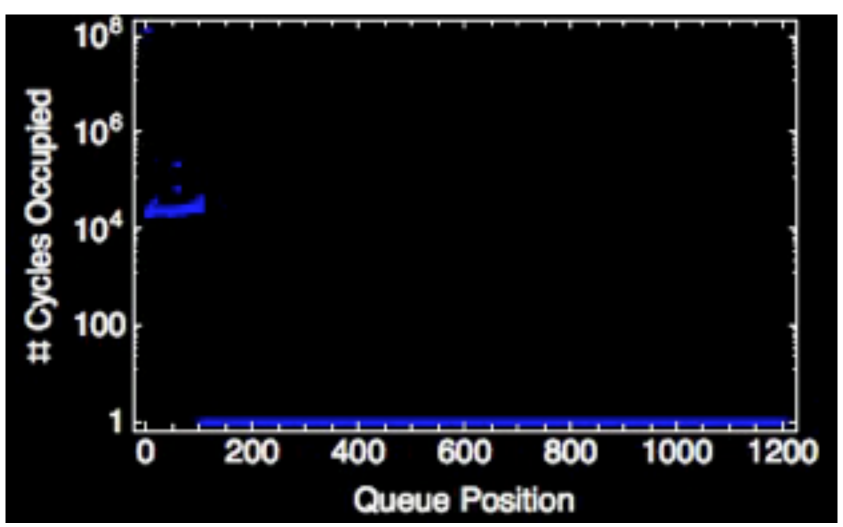


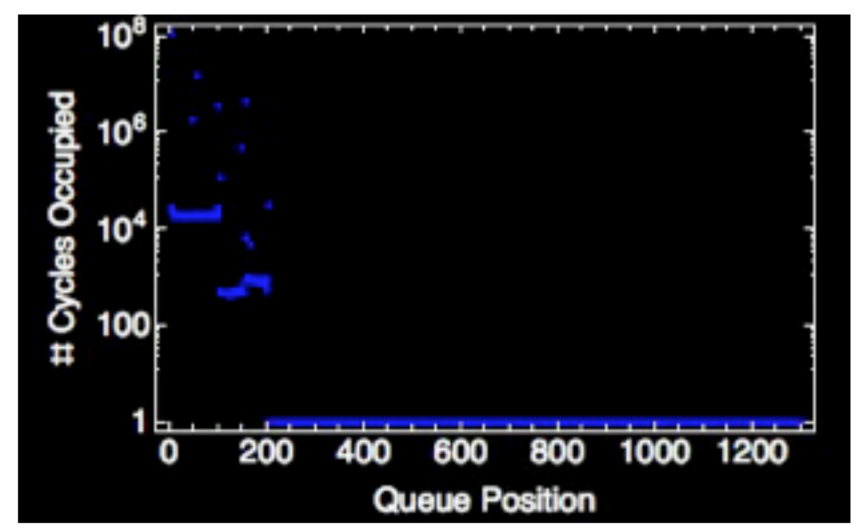




Optimization







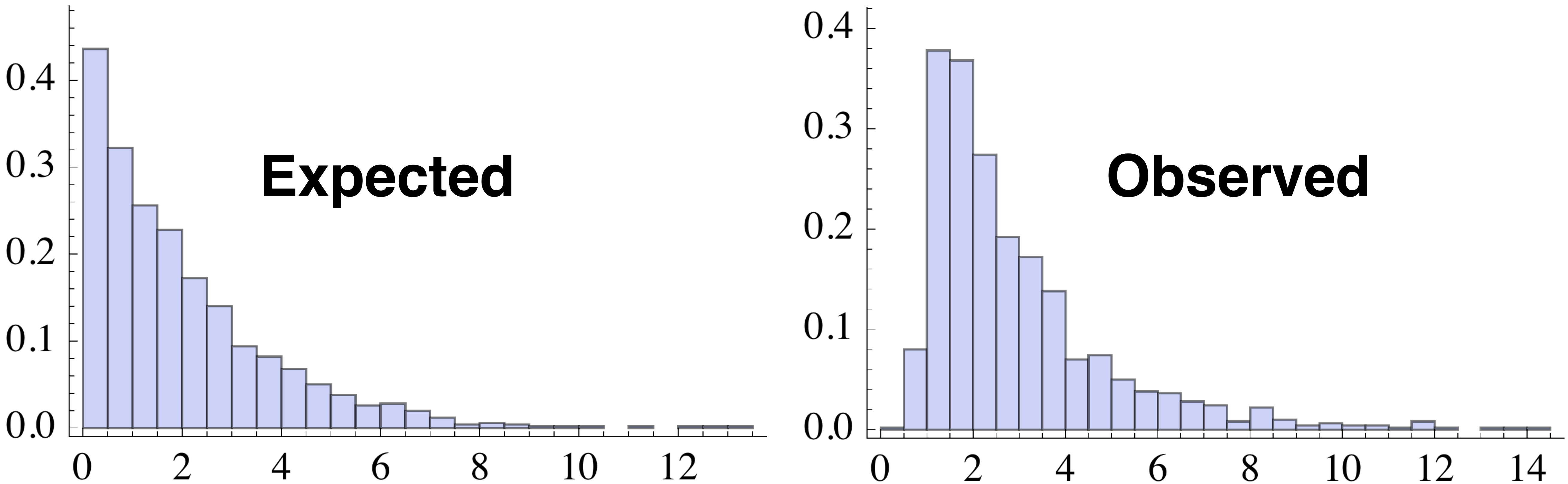
Streaming on Multi-core Systems

We want good models for streaming systems on shared multi-core systems (i.e., a cluster)

Problem: Accurate measurement is very difficult. Is there a way to decide on a model without it.

- Commodity multi-core timer availability and latency
- Frequency scaling and core migration
- Measuring modifies the application behavior

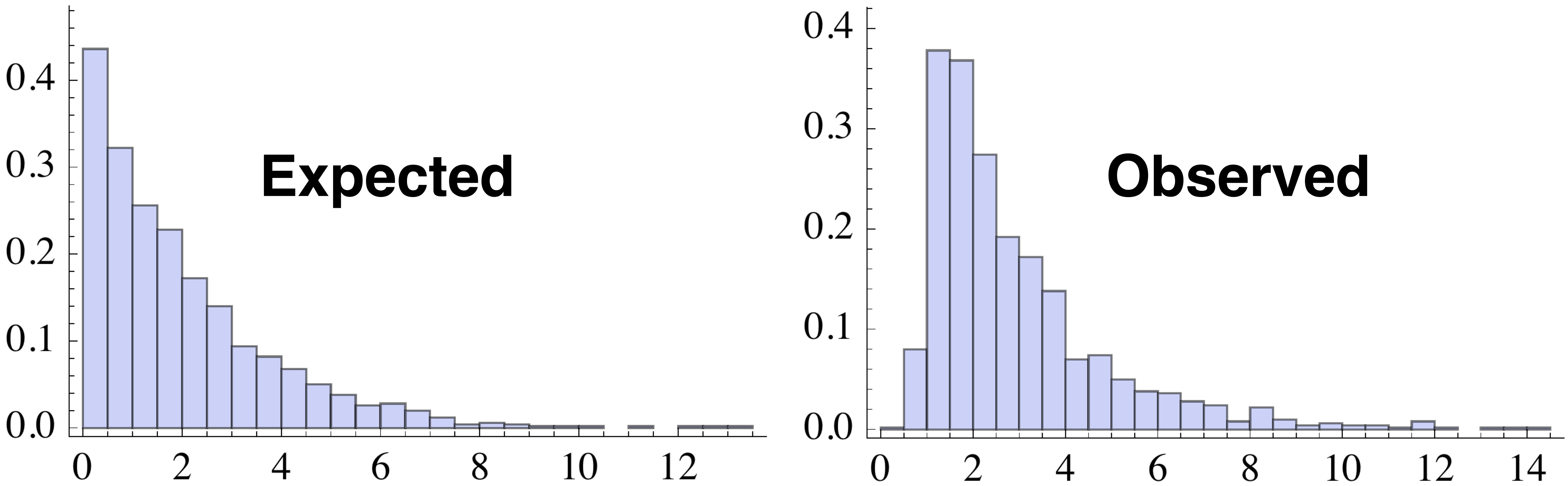
Derived Information



SBS
Stream Based
Supercomputing Lab
http://sbs.wustl.edu

Washington
University in St. Louis

Derived Information



Is there a pattern of minimal variation within the systems we're running on?

Avg. Service Time = E[X] + Error

SBS
Stream Based Supercomputing Lab
http://sbs.wustl.edu

Washington University in St.Louis

Goal

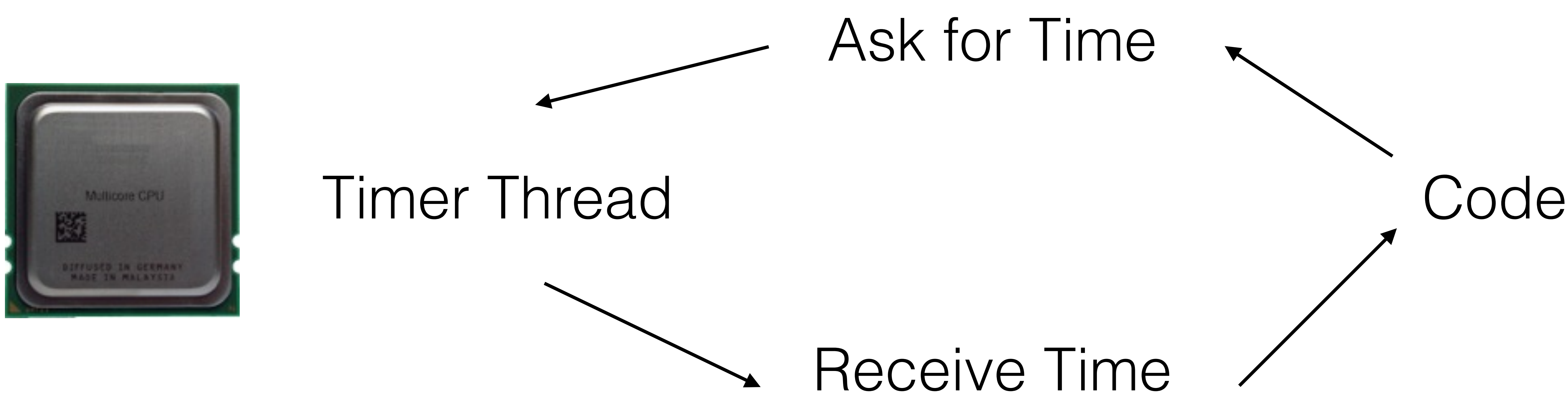
Find a distribution that characterizes the minimum expected variation of a hardware and software system

Use this characterization to accept or reject models

Process

- Measurement
- Workload definition
- Find a distribution
- Utilize the distribution to aid model selection

Timer Mechanism



Timer Mechanism

Timer Thread

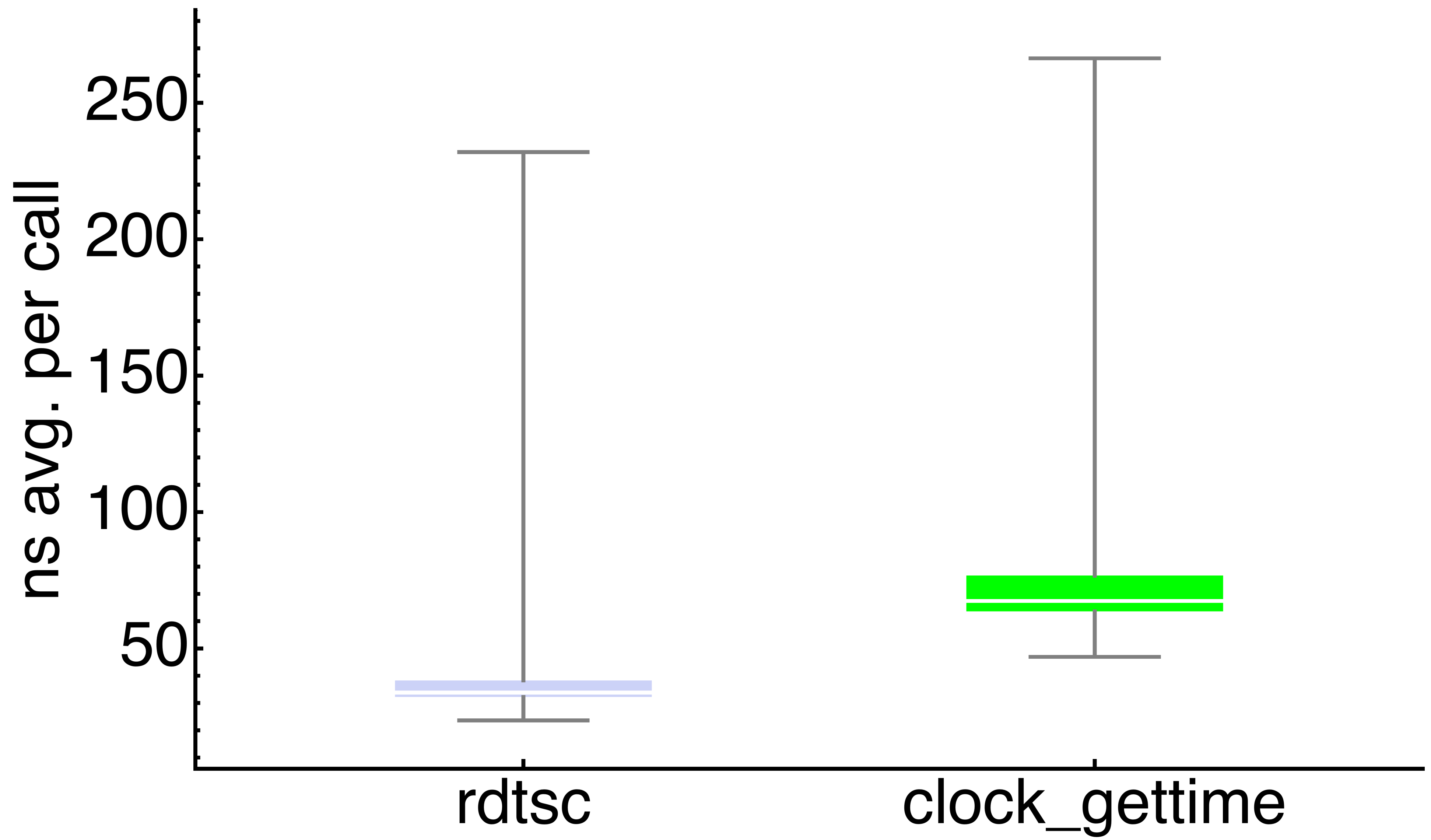
`rdtsc`

- x86 assembly
- varying methods to serialize
- relatively fast
- multiple drift issues

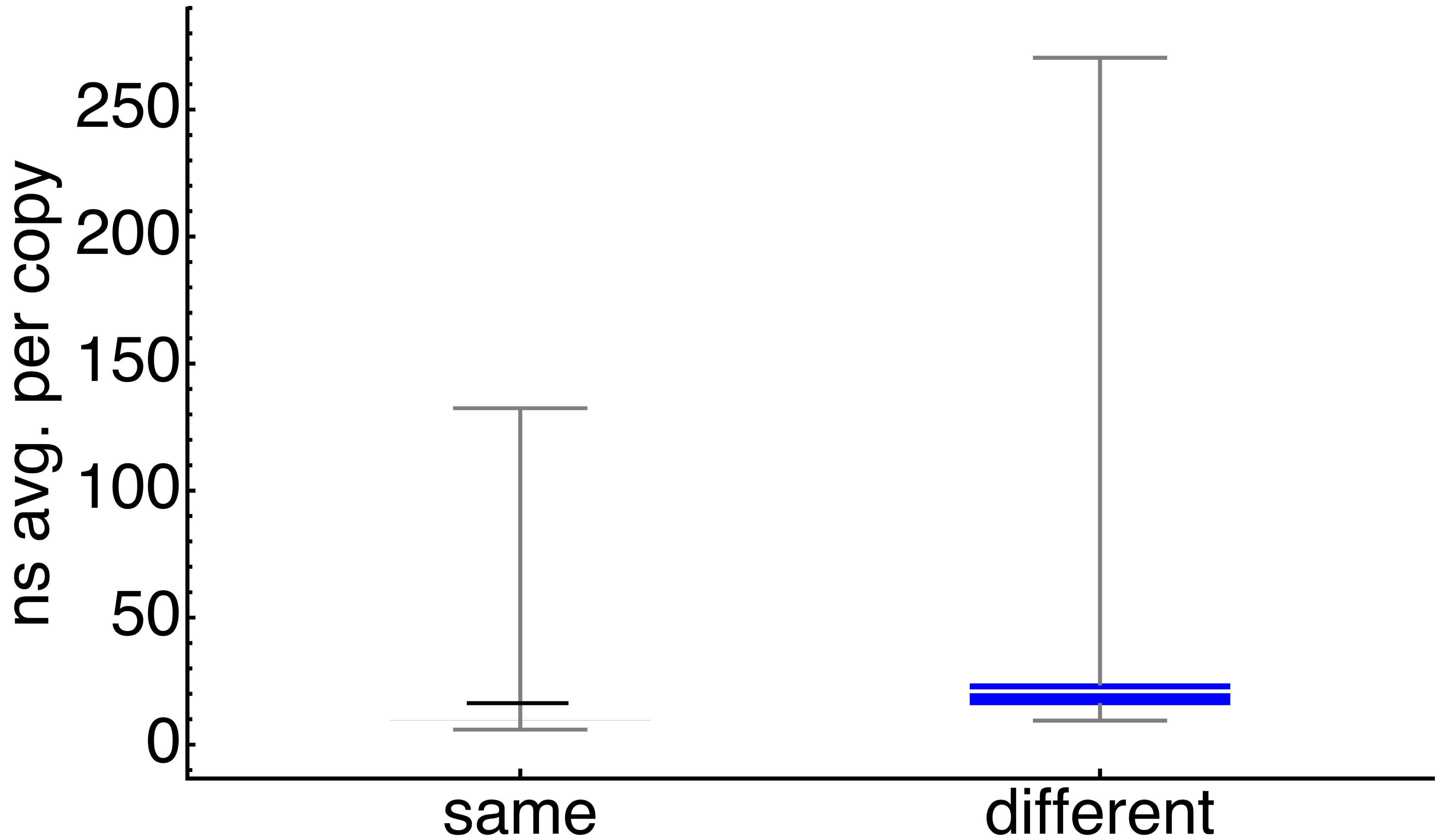
`clock_gettime`

- POSIX standard
- relatively accurate
- portable
- slower than `rdtsc`

Two Timing Choices



NUMA Node Variations



Stream Based
Supercomputing Lab
http://sbs.wustl.edu
Washington University in St. Louis

Minimize Variation

- Restricting timer to single core
- Use the x86 **`rdtsc`** instruction with processor recommended serializers for each processor type
- Keeping processes under test on the same NUMA node as timer
- Run timer thread with altered priority to minimize core context swaps

Best **C**ase **E**xecution **T**ime **V**ariation

- no-op instruction implemented in most processors
- usually takes exactly 1 cycle
- no real functional units are involved, so least taxing
- variation observed in execution time should be external to process

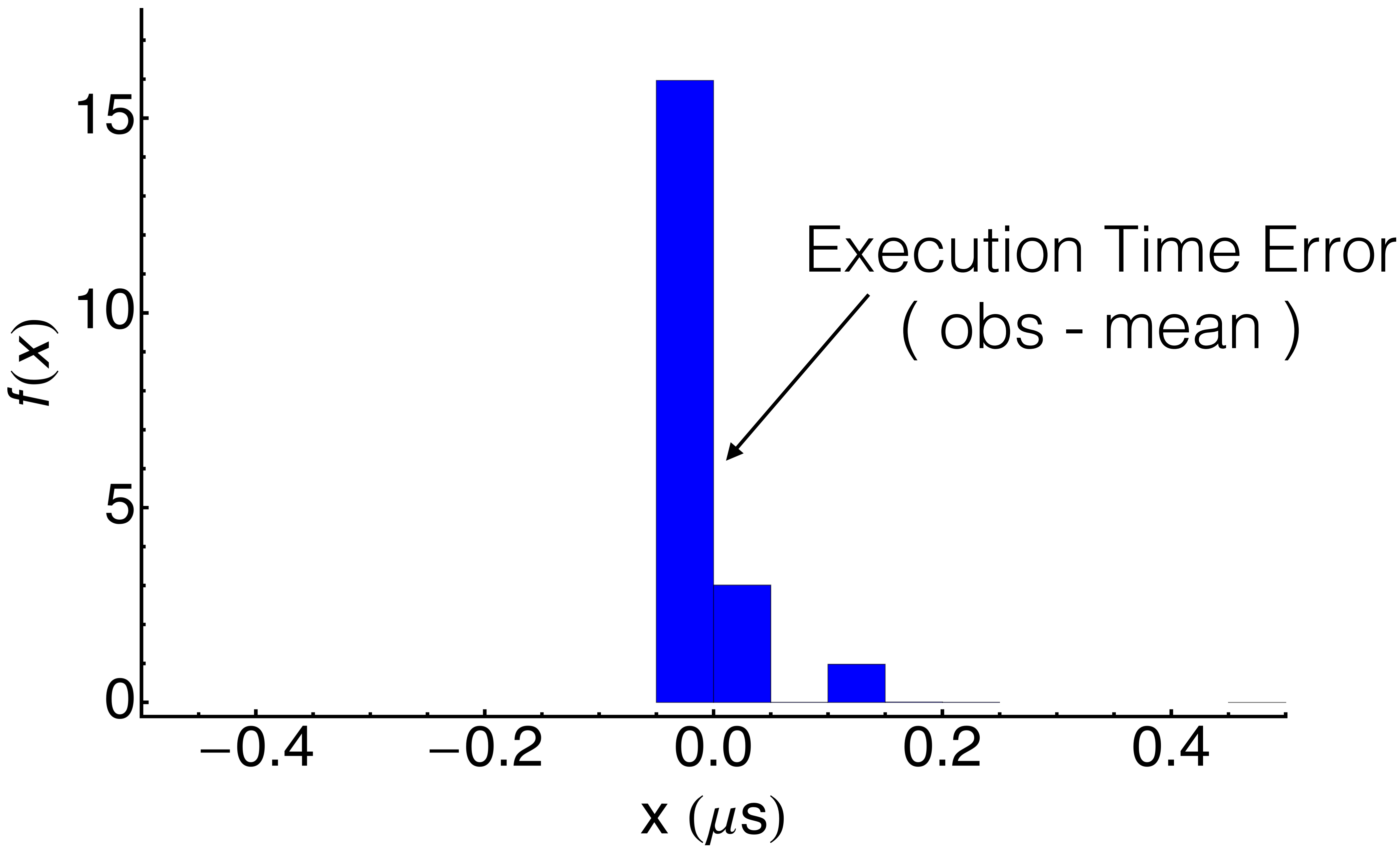
Data Collection

- no-op loops calibrated for various nominal times, tied to a single core and run thousands of times

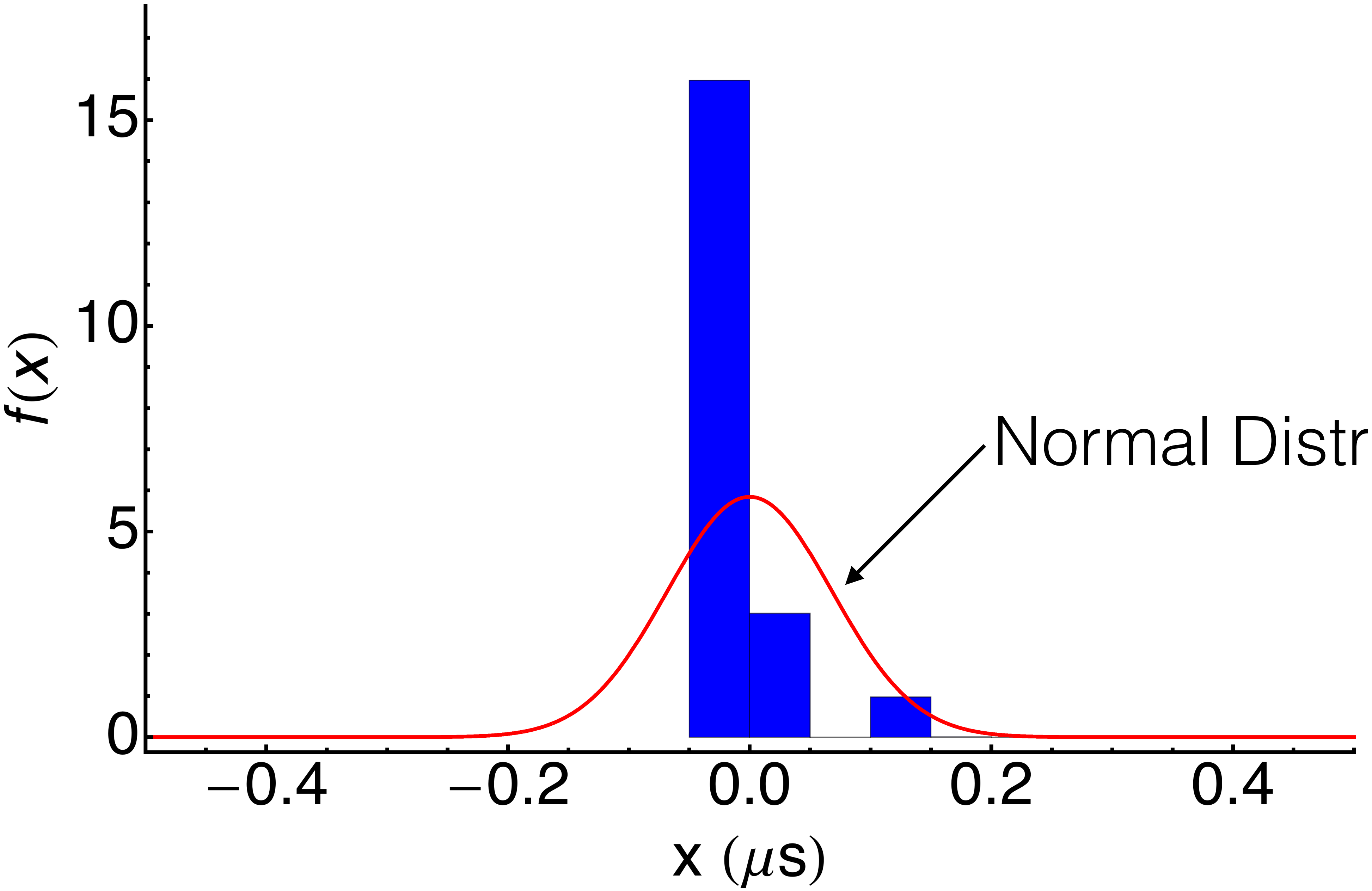
- Execution time measured end to end for each run, environment collected

- Parameters include:
    Number of processes executing on core
    Number of context swaps (voluntary, involuntary)
    Many others

Levy Distribution



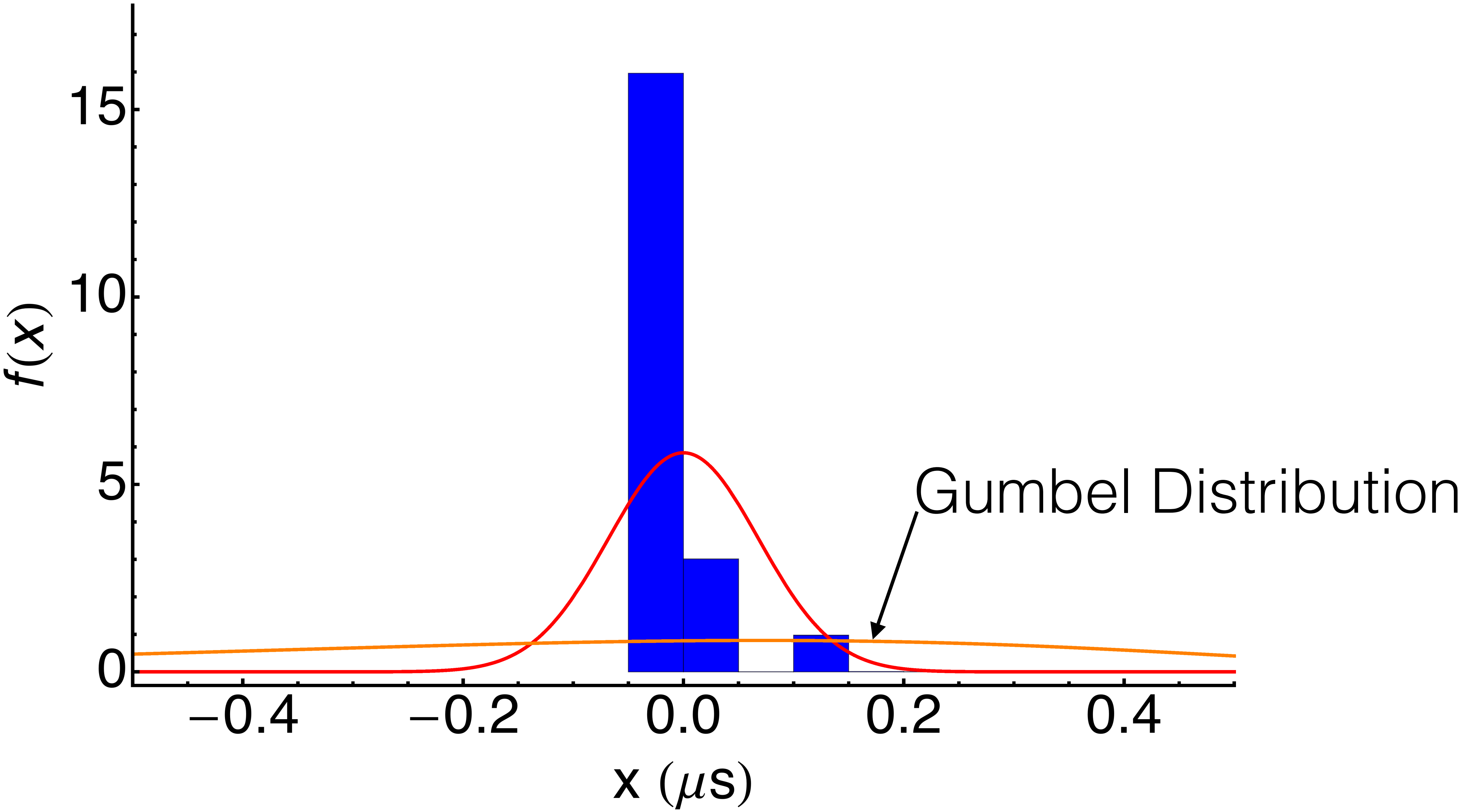
Levy Distribution



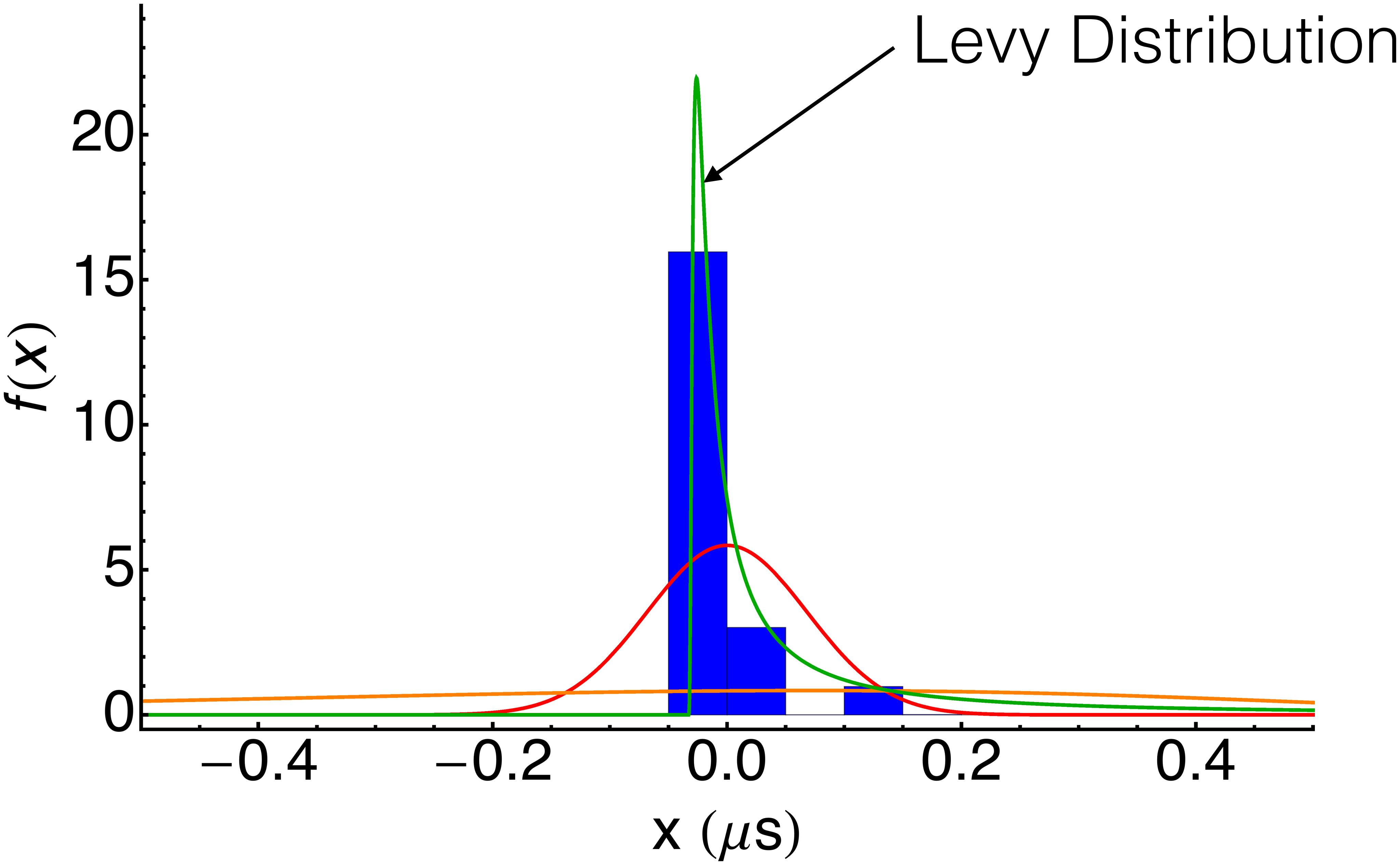
SBS
Stream Based
Supercomputing Lab
http://sbs.wustl.edu

Washington
University in St. Louis

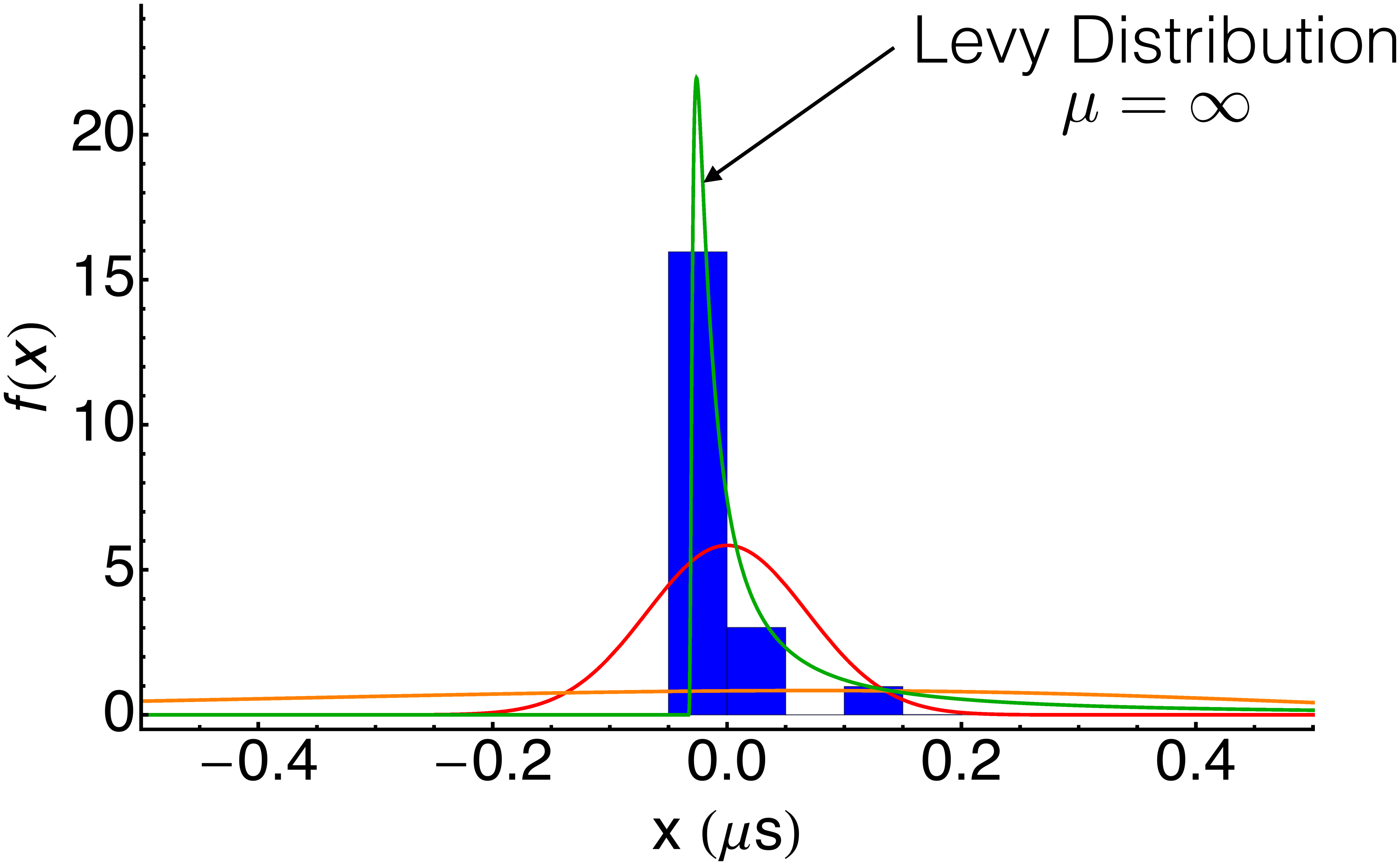
Levy Distribution



Levy Distribution



Levy Distribution



SBS
Stream Based
Supercomputing Lab
http://sbs.wustl.edu

Washington
University in St. Louis

Levy Distribution

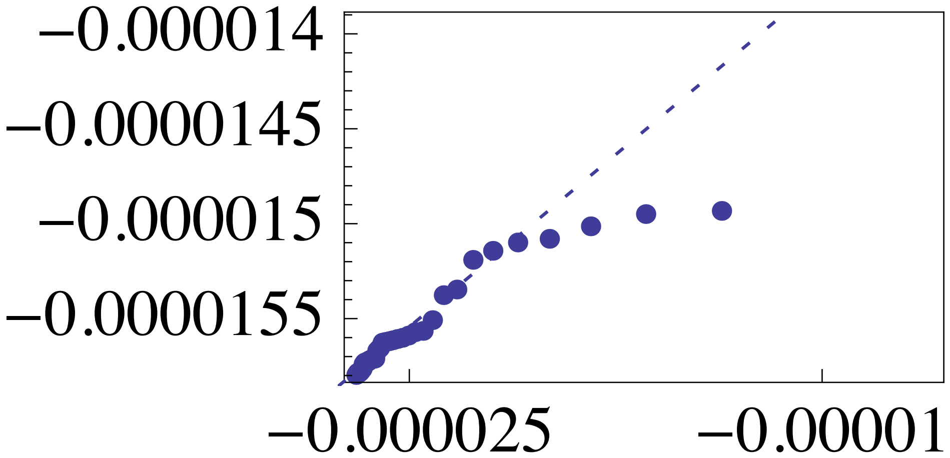
- Truncation enables mean calculation, but requires fitting to each dataset to find where to truncate

- The truncation parameters are correlated to both the number of processes per core and the expected execution time

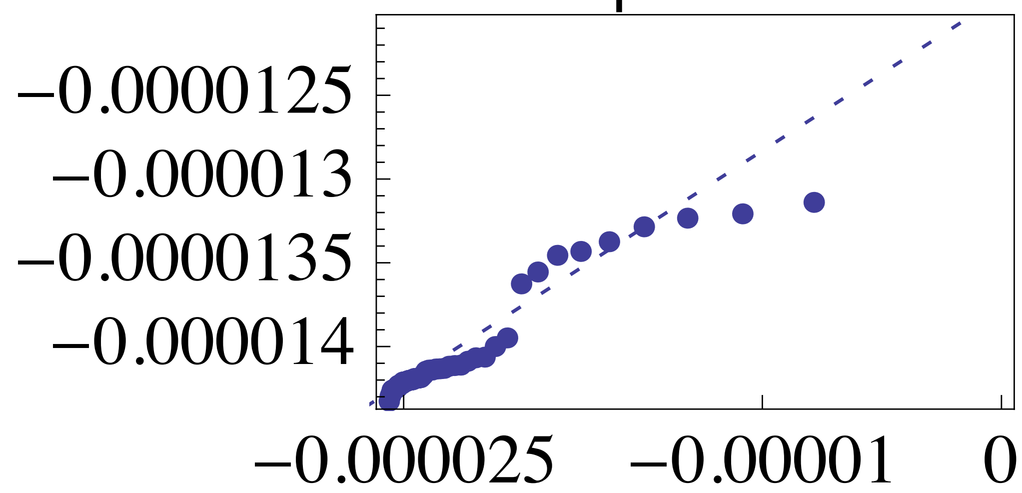
- Roughly linear relationship gives an approximate solution to truncation parameters without refitting

Levy Fit

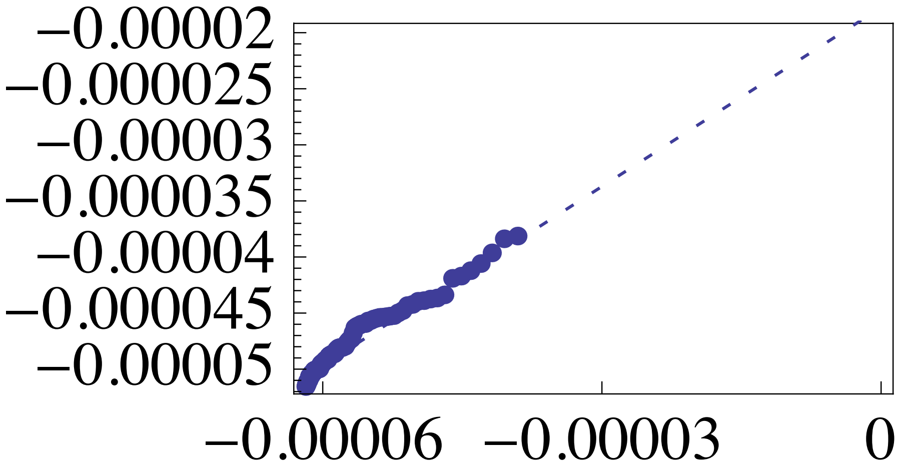
1 - 5 processes



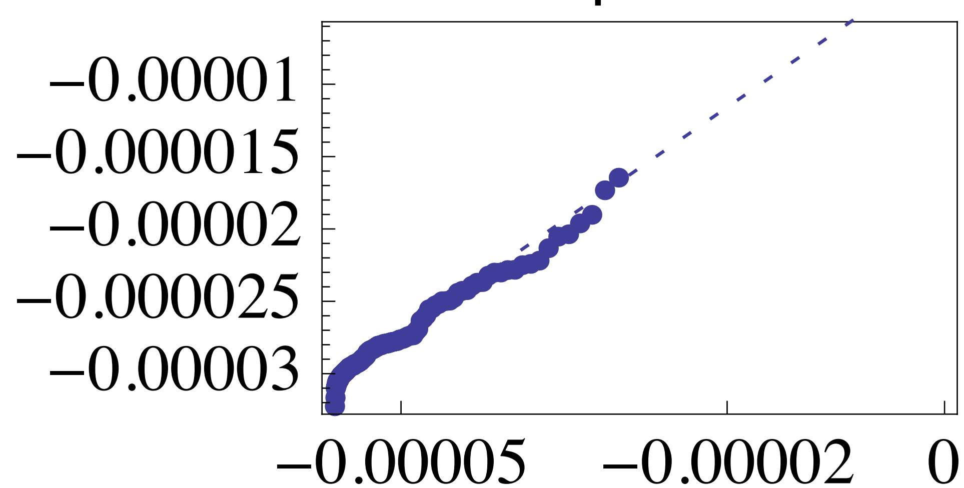
6 - 10 processes



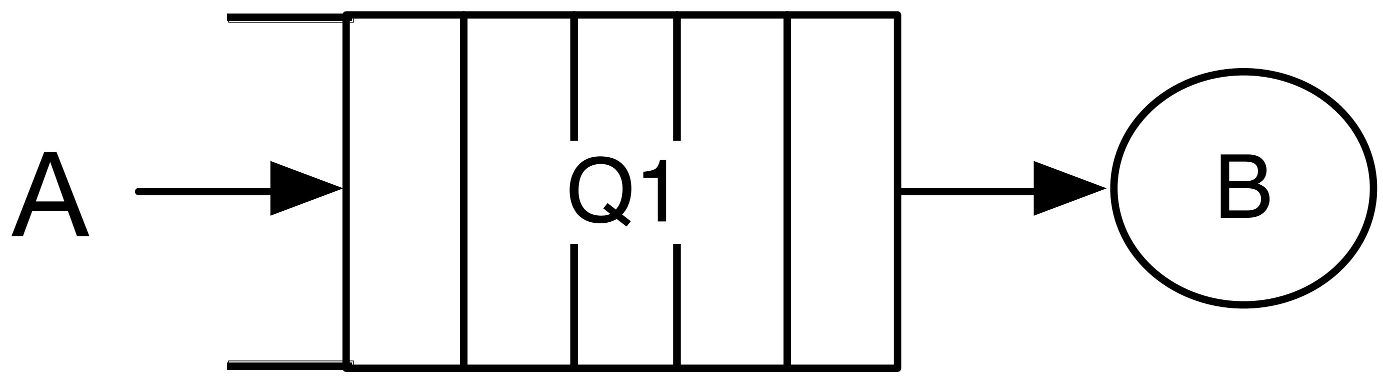
11 - 15 processes



16 - 20 processes



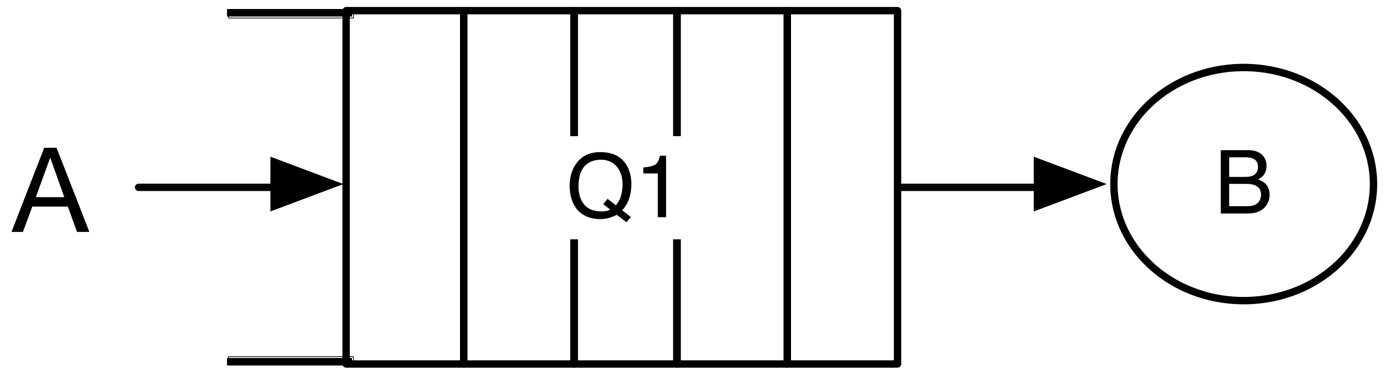
Test Setup



Question: Can we use an M/M/1 queueing model to estimate the mean queue occupancy of this system?

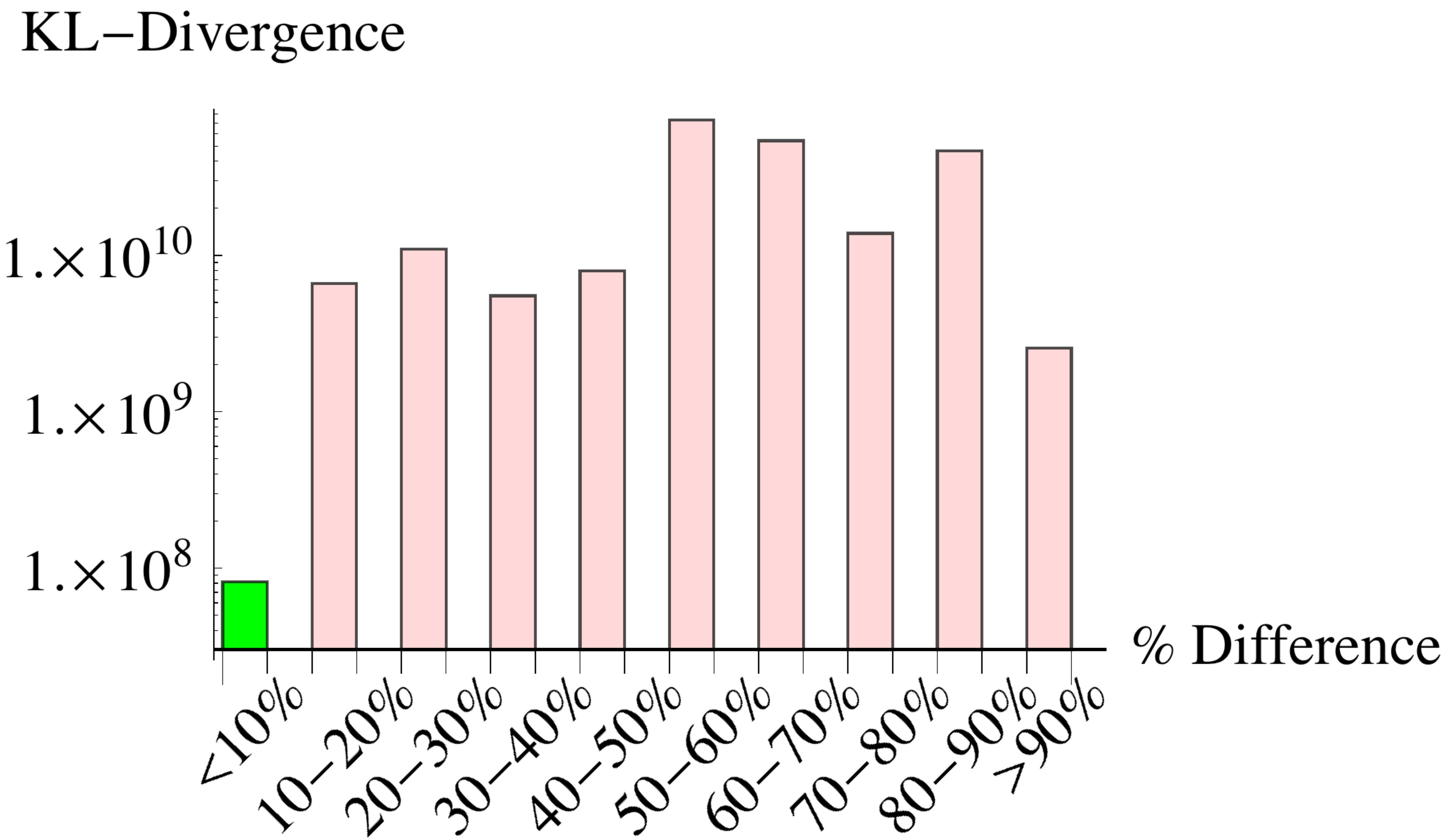
Hypothesis: Lower Kullback-Leibler (KL) divergence between expected and realized distribution is associated with higher model accuracy.

Test Setup



1. Dedicated thread of execution monitors queue occupancy
2. Calculate the estimated mean queue occupancy using the M/M/1 model
3. Calculate KL Divergence for the arrival process distribution using the truncated Levy distribution noise model

Convolution with Exponential



Conclusions

- The truncated Levy distribution can be used to approximate BCETV
- The distribution of BCETV can be used as a tool to accept or reject a stochastic queueing model based on distributional assumptions
- KL divergence between the expected and convolved distribution highly correlates with queue model accuracy

Parting Notes

Slides available here:
sbs.wust.edu

Timer C++ template code:
http://goo.gl/ItJ3jP

Test harness used to collect data:
http://goo.gl/U1VG6N