Accordion

Toward Soft Near-threshold Voltage Computing

Ulya R. Karpuzcu, Ismail Akturk

UNIVERSITY OF MINNESOTA

Nam Sung Kim

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

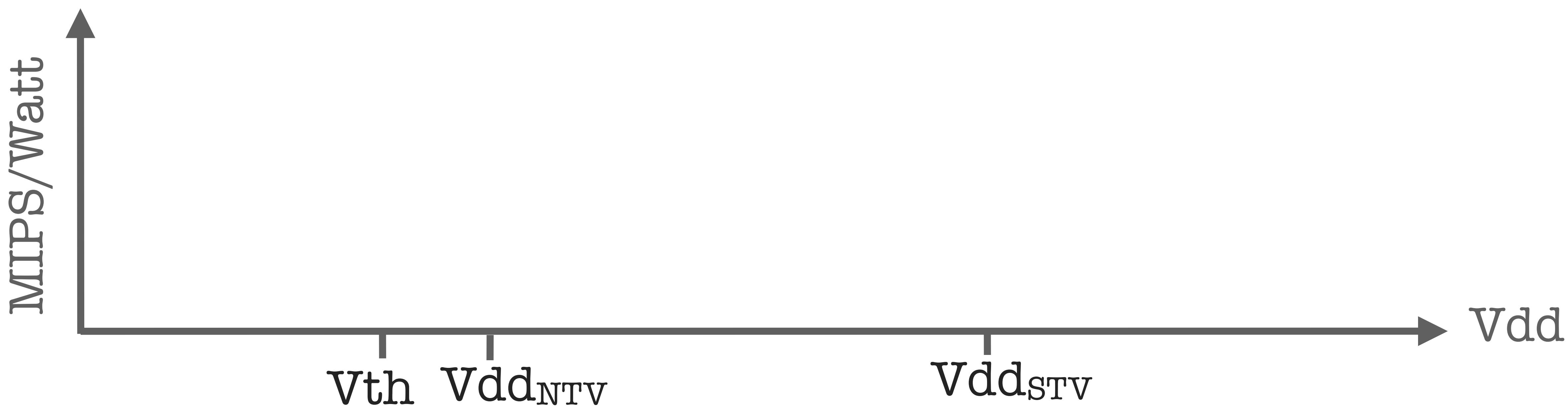
Near-threshold Voltage Computing (NTC)

Near-threshold Voltage Computing (NTC)

- Supply voltage Vdd remains slightly above threshold voltage Vth

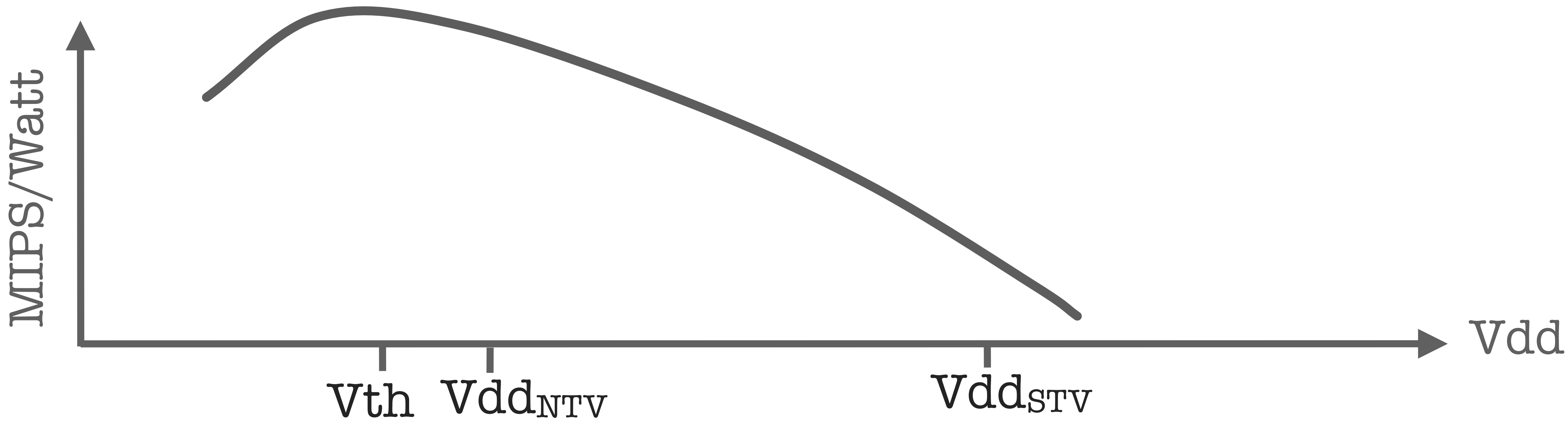
Near-threshold Voltage Computing (NTC)

- Supply voltage Vdd remains slightly above threshold voltage Vth



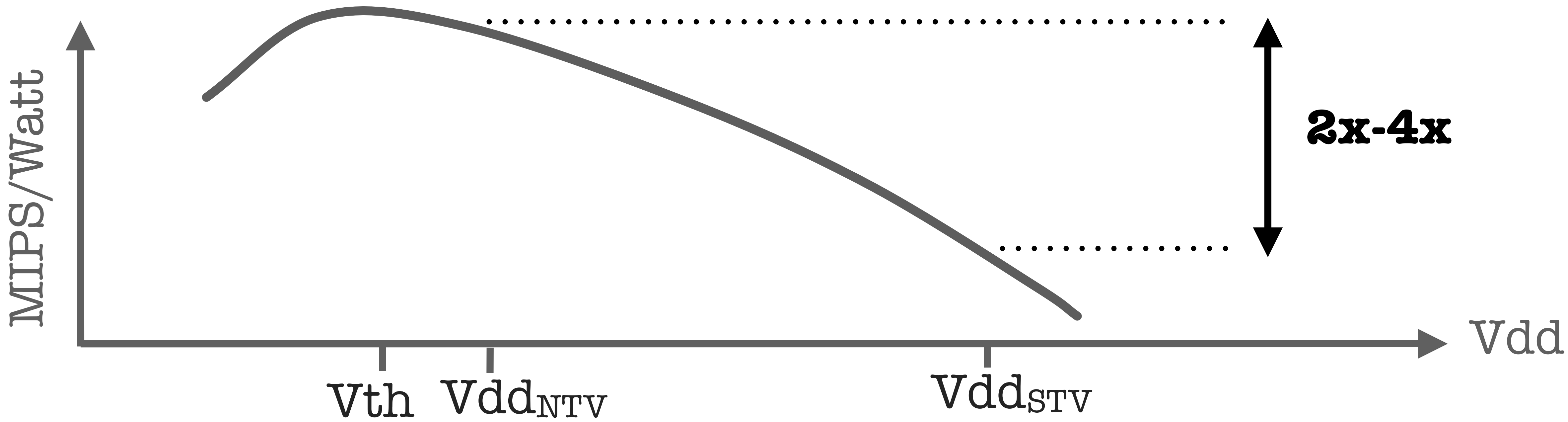
Near-threshold Voltage Computing (NTC)

- Supply voltage Vdd remains slightly above threshold voltage Vth
- Energy efficiency increases as Vdd reaches Vth



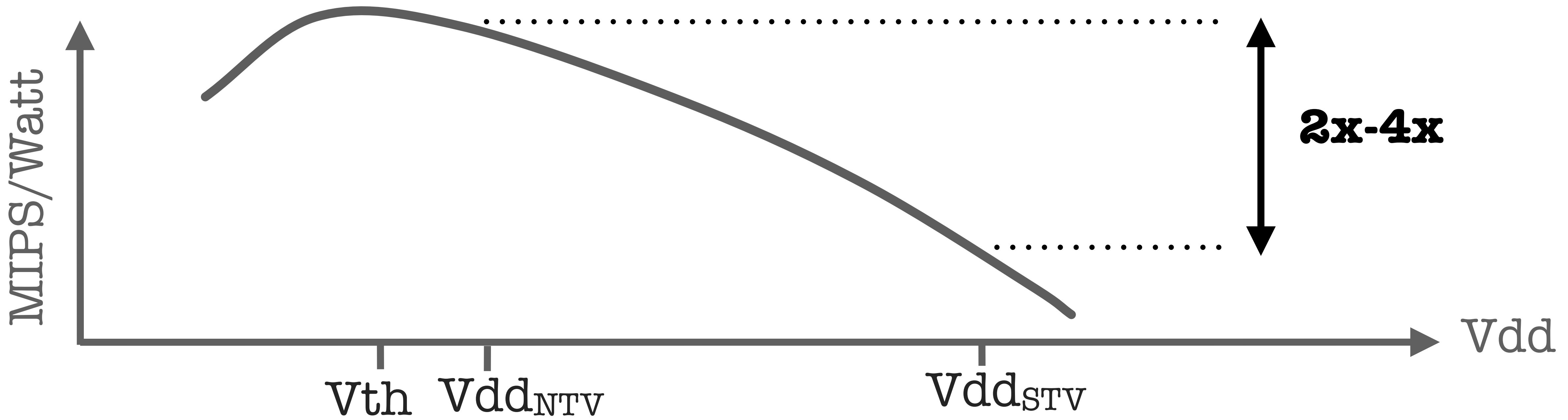
Near-threshold Voltage Computing (NTC)

- Supply voltage Vdd remains slightly above threshold voltage Vth
- Energy efficiency increases as Vdd reaches Vth
  - 2-4x more energy efficient than super-threshold (STV) operation



Near-threshold Voltage Computing (NTC)

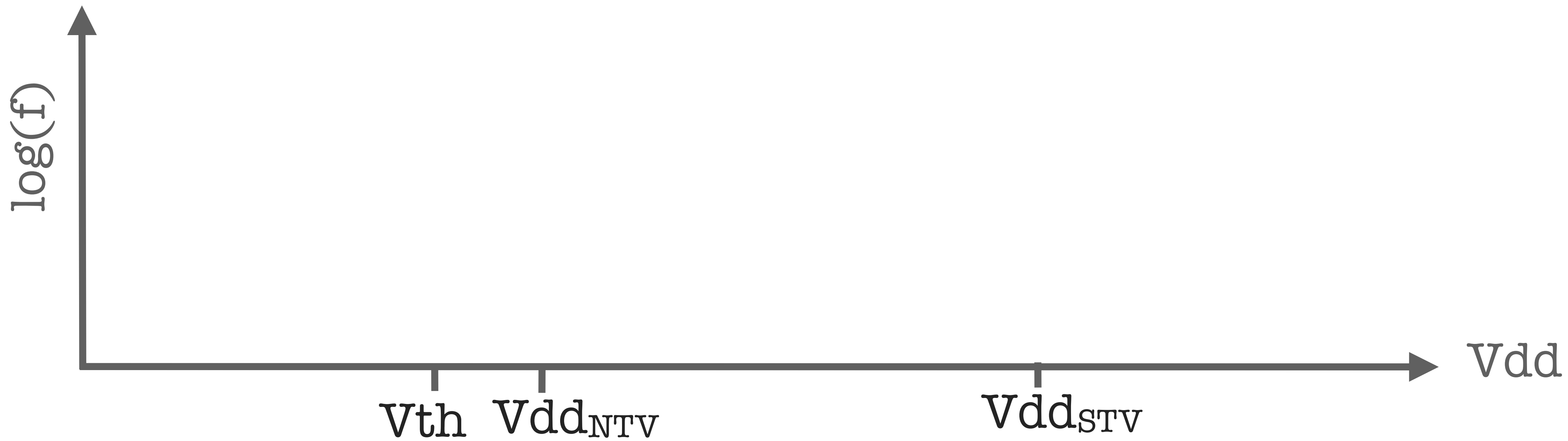
- Supply voltage Vdd remains slightly above threshold voltage Vth
- Energy efficiency increases as Vdd reaches Vth
  - 2-4x more energy efficient than super-threshold (STV) operation



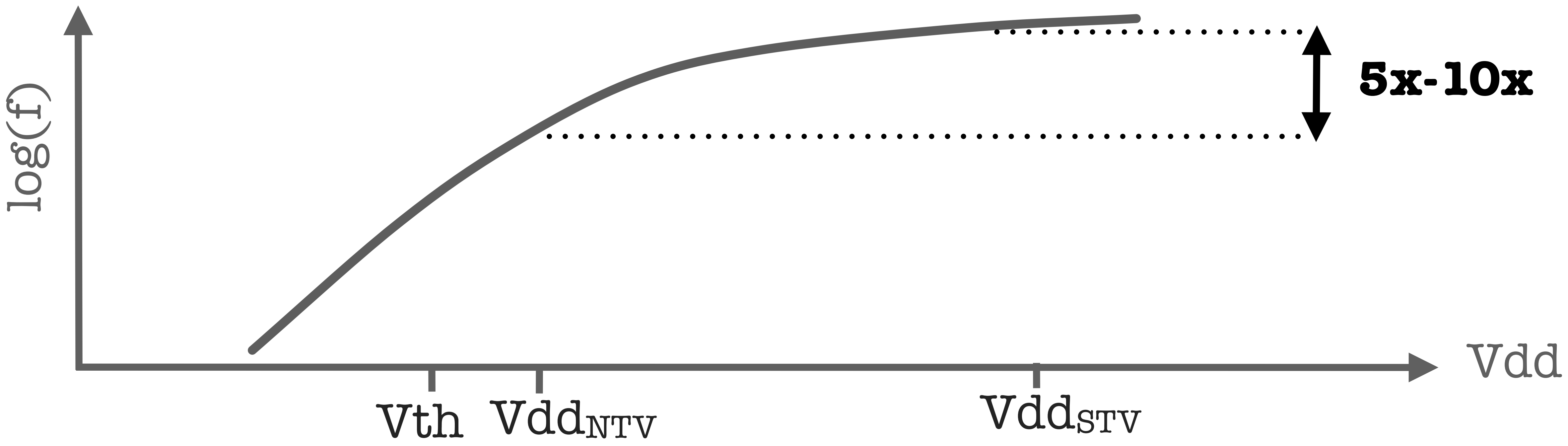
How close to Vdd can Vth get?

NTC Basics: How close to Vdd can Vth get?

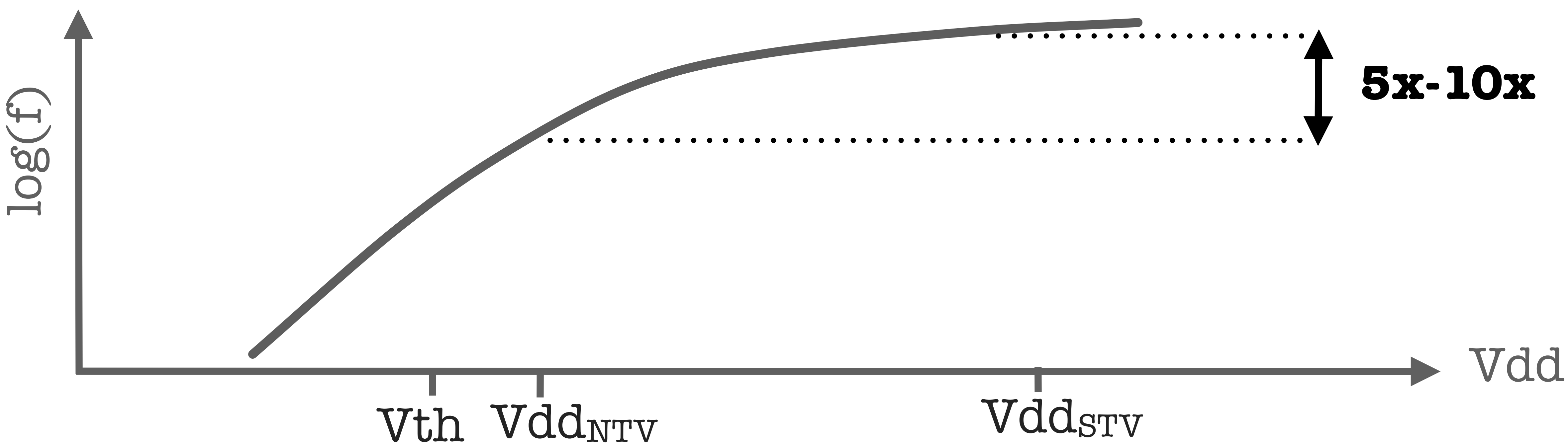
NTC Basics: How close to Vdd can Vth get?



NTC Basics: How close to Vdd can Vth get?

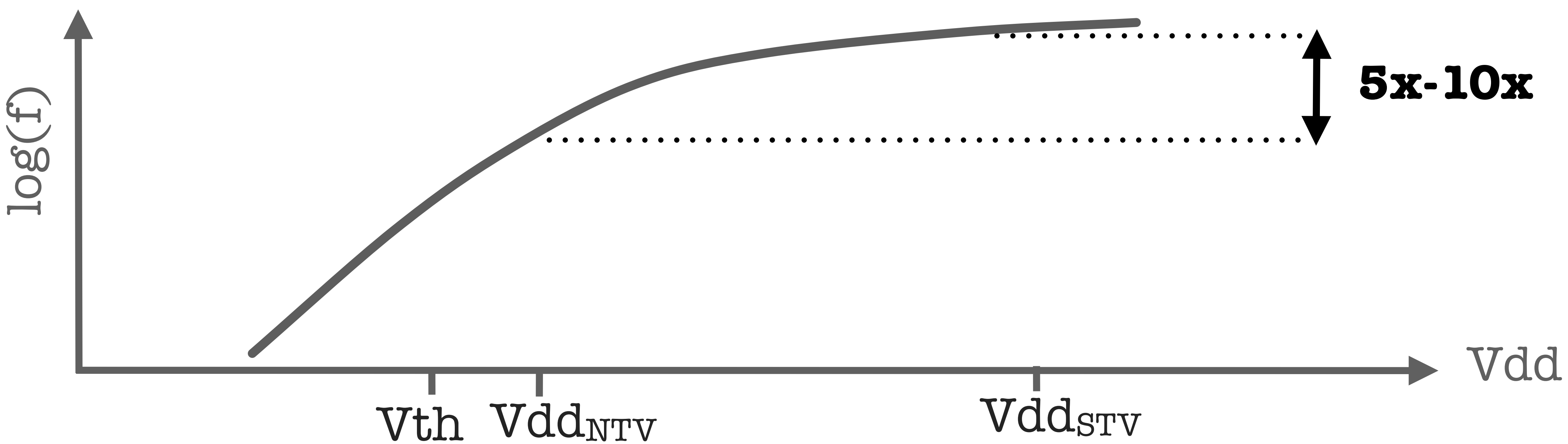


NTC Basics: How close to Vdd can Vth get?



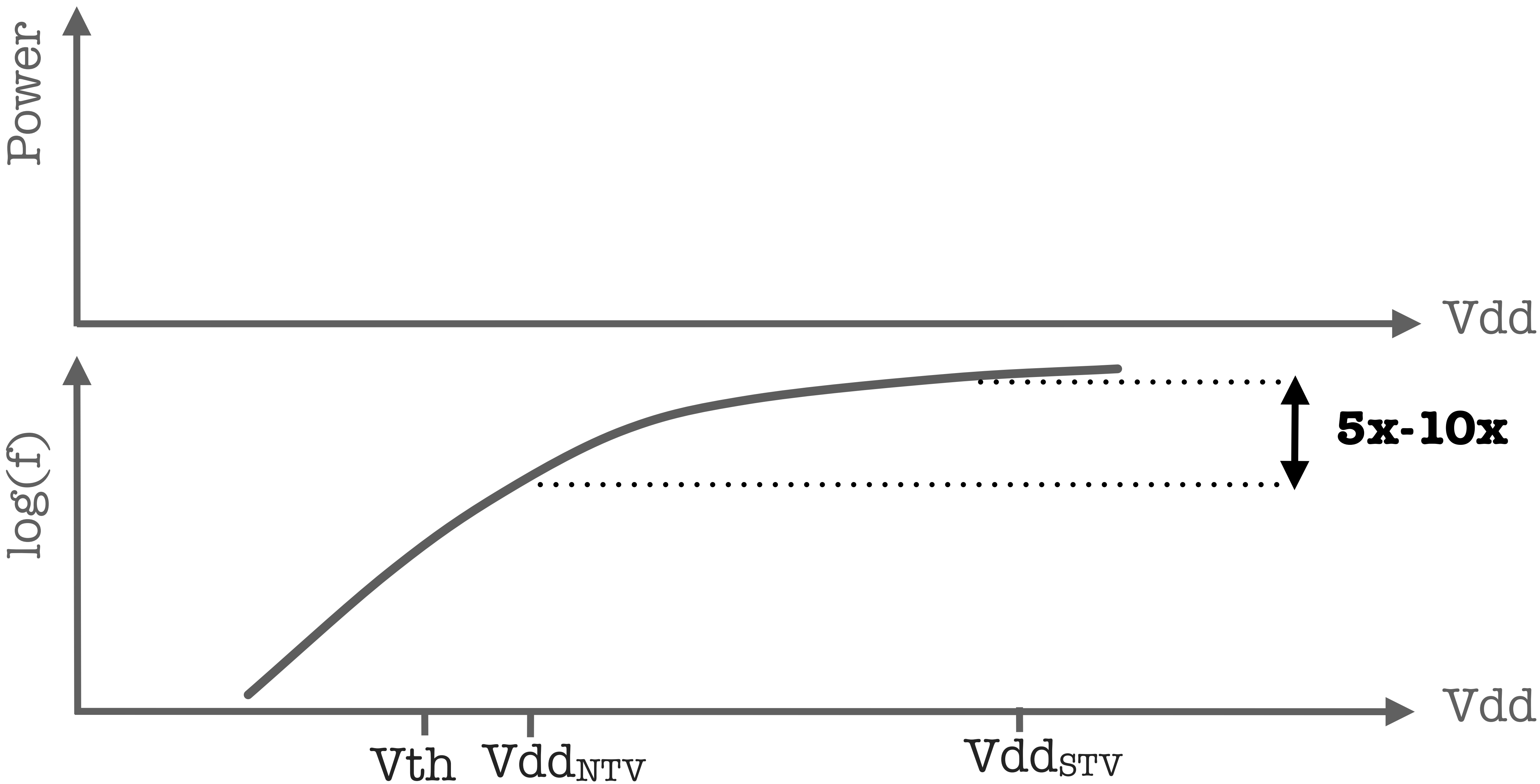
- Execution time is proportional to work per parallel task x f

NTC Basics: How close to Vdd can Vth get?



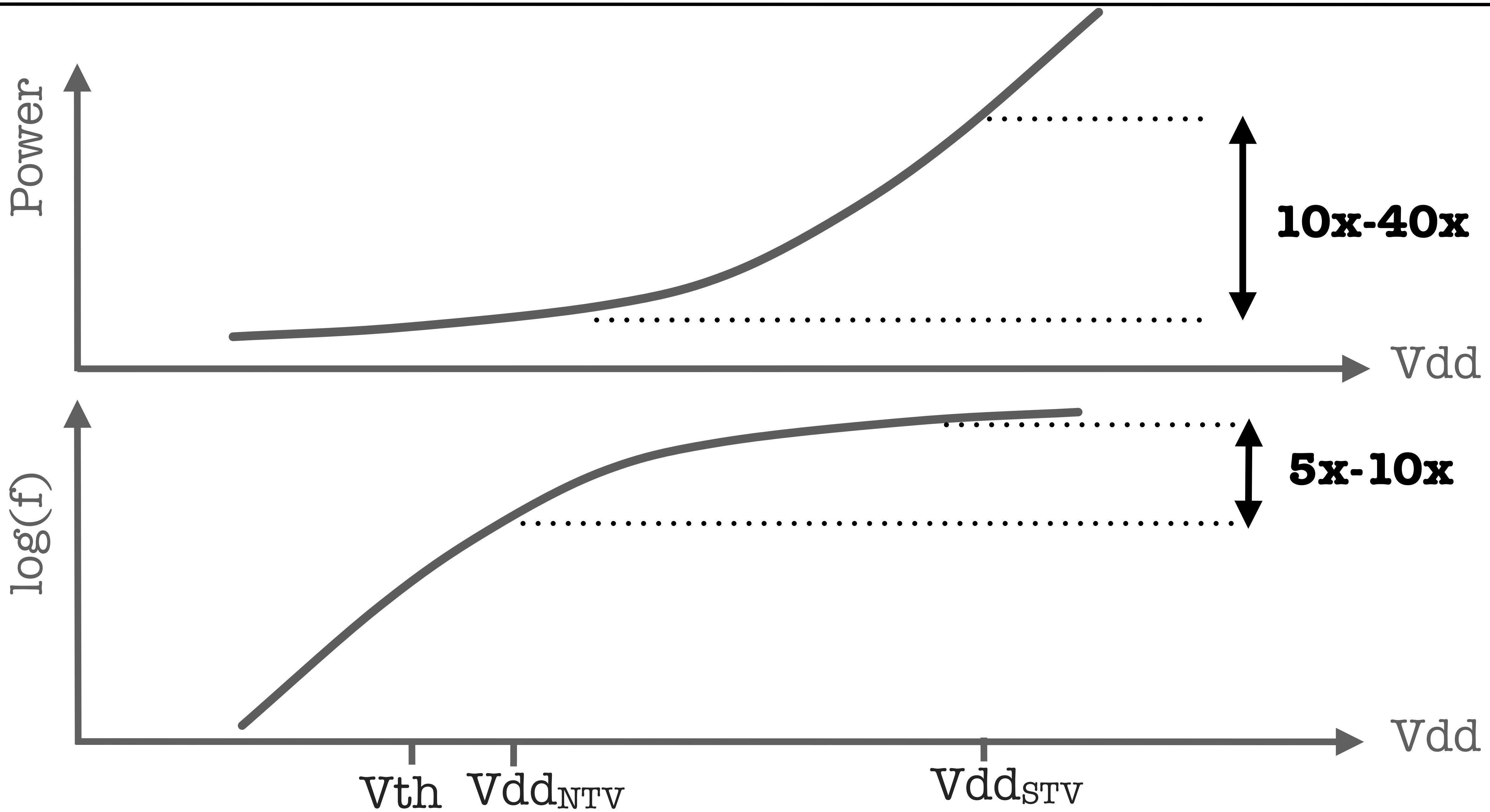
- Execution time is proportional to work per parallel task x f
  - No degradation if 5-10x more cores engaged in computation

NTC Basics: How close to Vdd can Vth get?



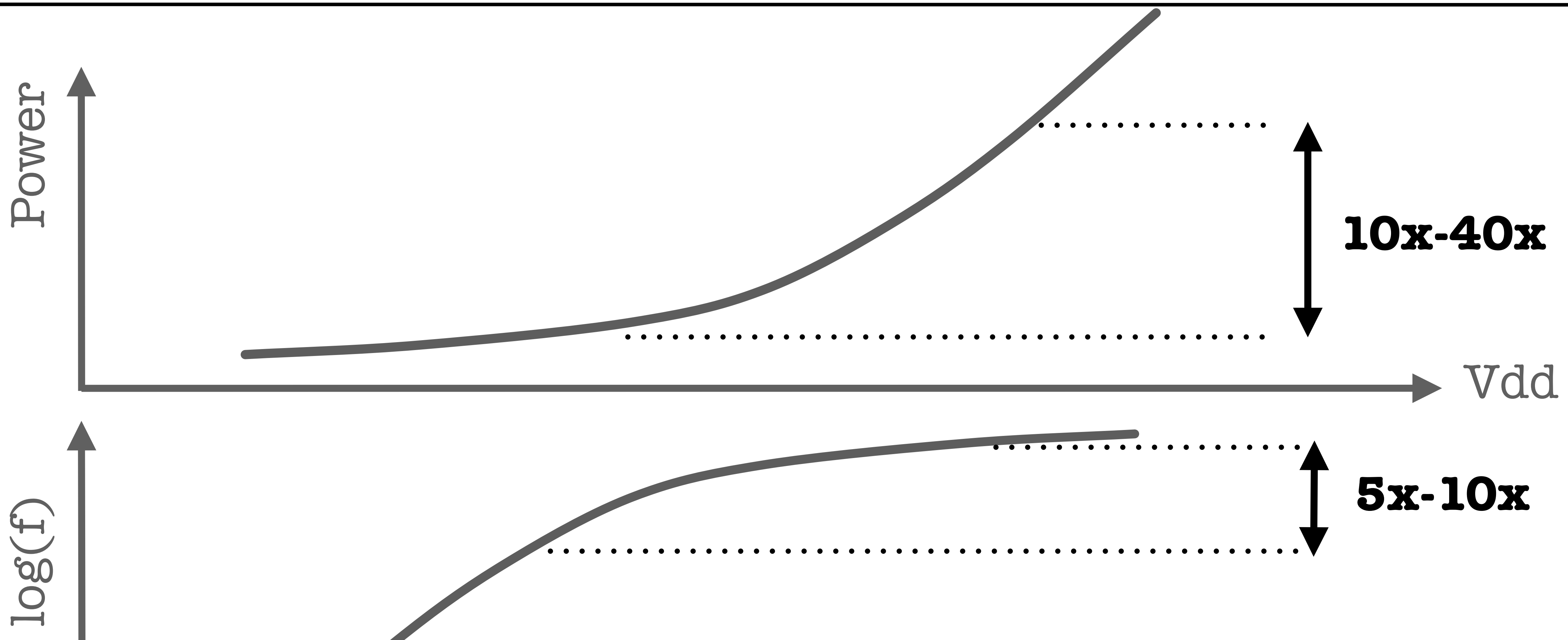
- Execution time is proportional to work per parallel task x f
  - No degradation if 5-10x more cores engaged in computation

NTC Basics: How close to Vdd can Vth get?



- Execution time is proportional to work per parallel task x f
  - No degradation if 5-10x more cores engaged in computation
  - 10-40x power savings per core can accommodate 5-10x more cores

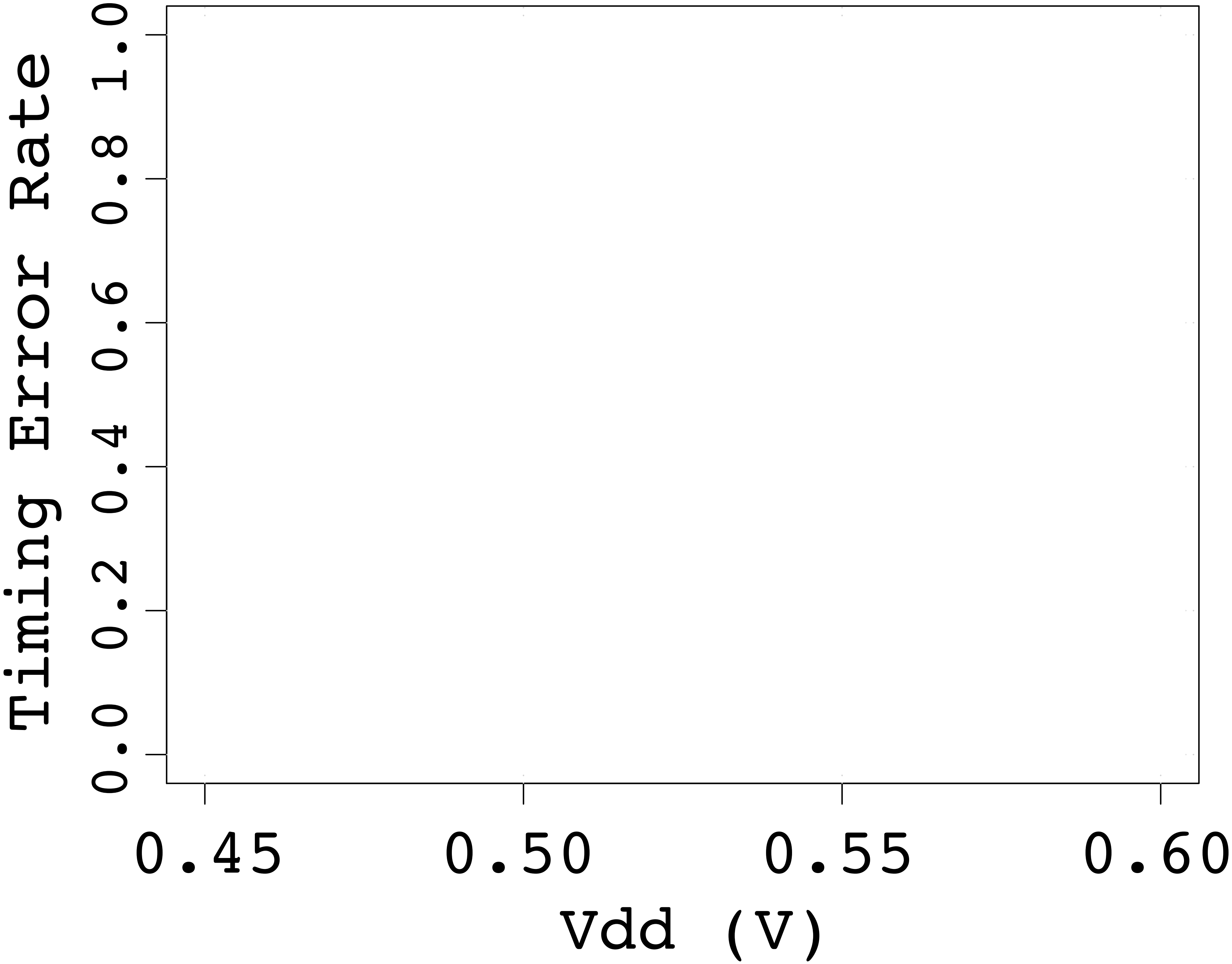
NTC Basics: How close to Vdd can Vth get?



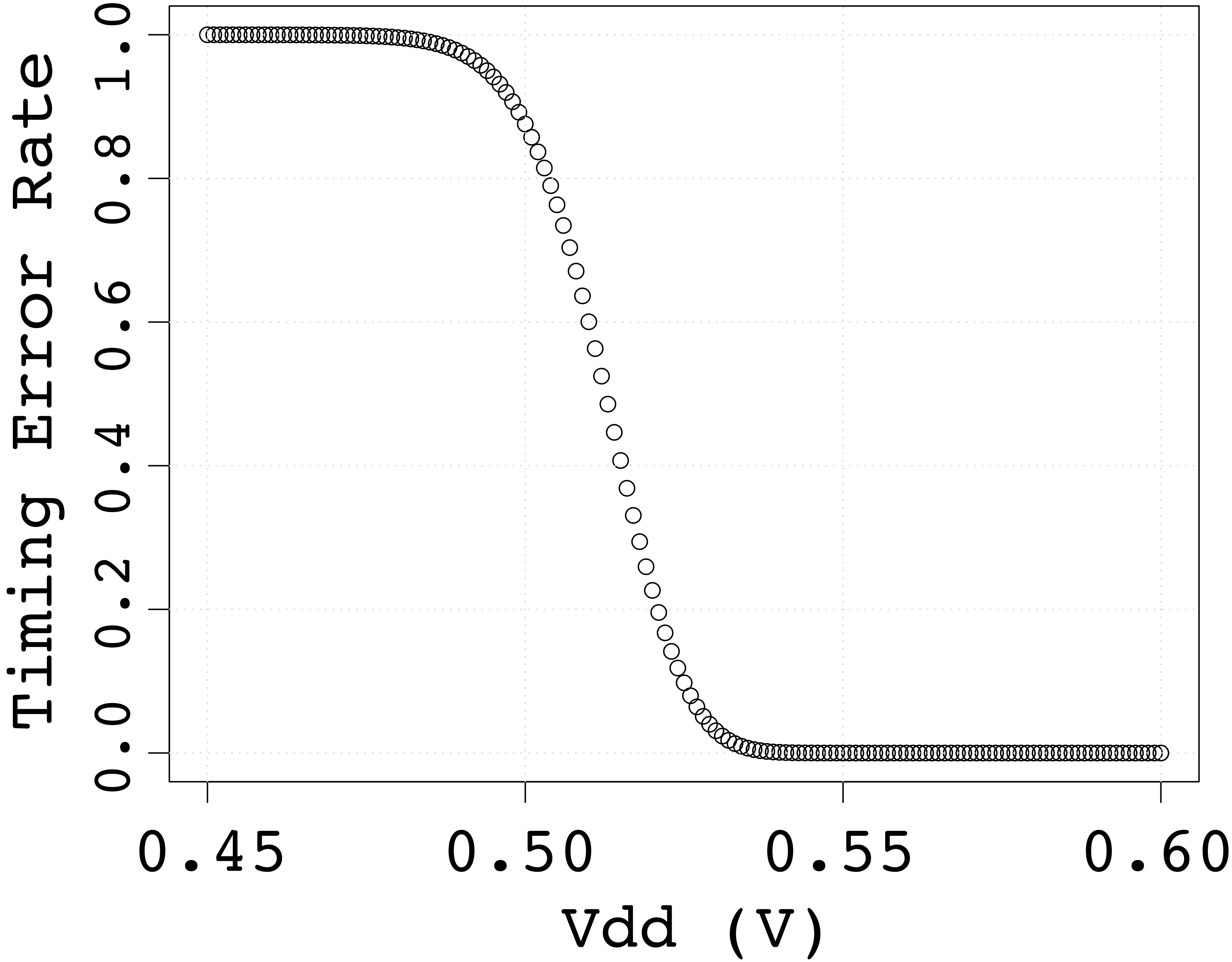
Limited by the degree of parallelism in application

- Execution time is proportional to work per parallel task x f
  - No degradation if 5-10x more cores engaged in computation
  - 10-40x power savings per core can accommodate 5-10x more cores

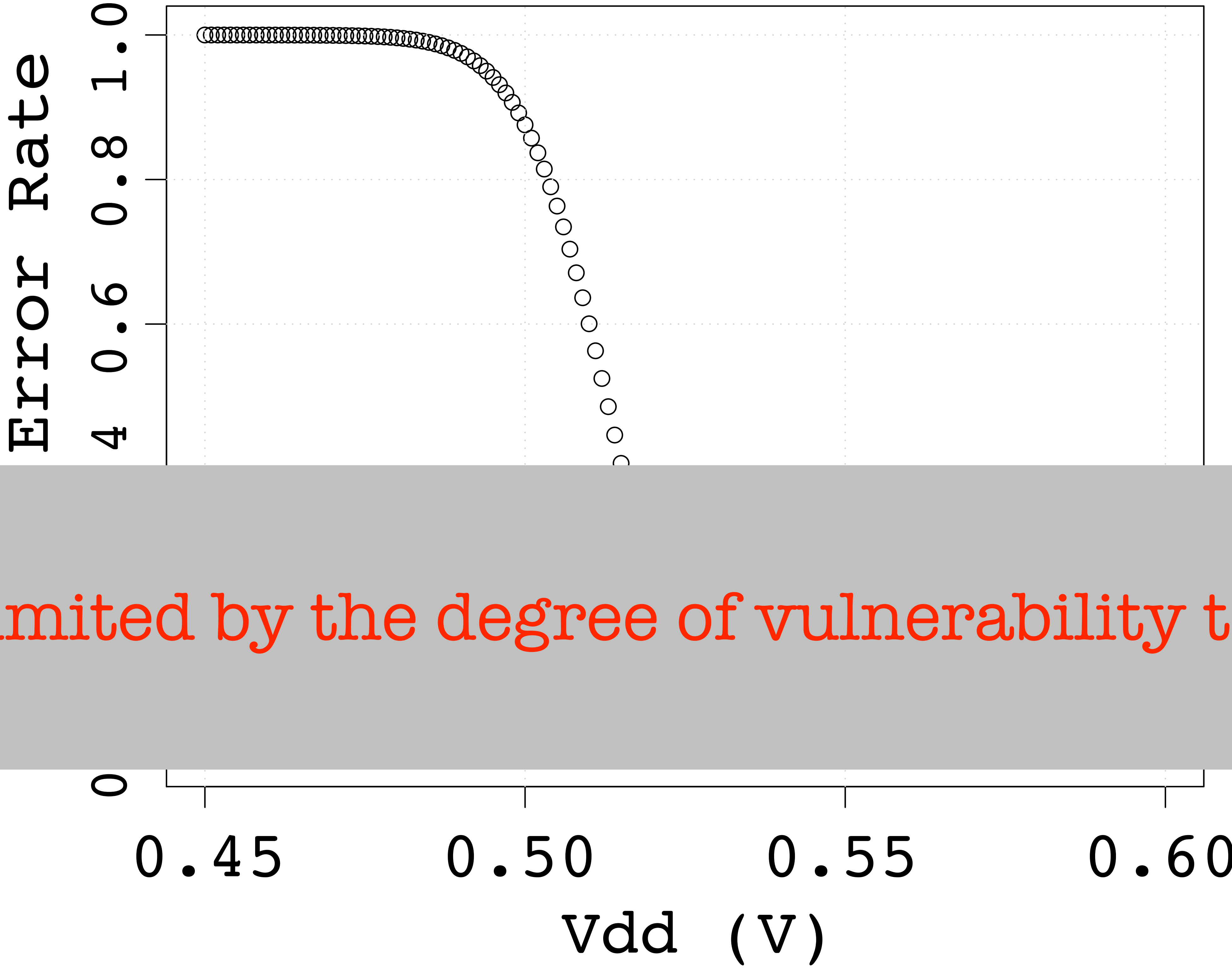
NTC Basics: How close to Vdd can Vth get?



NTC Basics: How close to Vdd can Vth get?



NTC Basics: How close to Vdd can Vth get?



Limited by the degree of vulnerability to variation

Accordion Basics

Accordion Basics

- How to close the gap between NTC and STC execution times?

Accordion Basics

- How to close the gap between NTC and STC execution times?

$$\texttt{Execution Time} \propto \frac{\texttt{Problem Size}}{\texttt{f} \times \texttt{Core Count}}$$

Accordion Basics

- How to close the gap between NTC and STC execution times?

$$\mathtt{Execution\ Time} \propto \frac{\mathtt{Problem\ Size}}{\mathtt{f} \times \mathtt{Core\ Count}}$$

$$\frac{\mathtt{Problem\ Size}_{\mathtt{NTV}}}{\mathtt{f}_{\mathtt{NTV}} \times \mathtt{Core\ Count}_{\mathtt{NTV}}} \rightarrow \frac{\mathtt{Problem\ Size}_{\mathtt{STV}}}{\mathtt{f}_{\mathtt{STV}} \times \mathtt{Core\ Count}_{\mathtt{STV}}}$$

Accordion Basics

- How to close the gap between NTC and STC execution times?

$$\texttt{Execution Time} \propto \frac{\texttt{Problem Size}}{\texttt{f} \times \texttt{Core Count}}$$

$$\frac{\texttt{Problem Size}_{\texttt{NTV}}}{\texttt{f}_{\texttt{NTV}} \times \texttt{Core Count}_{\texttt{NTV}}} \rightarrow \frac{\texttt{Problem Size}_{\texttt{STV}}}{\texttt{f}_{\texttt{STV}} \times \texttt{Core Count}_{\texttt{STV}}}$$

$$\texttt{f}_{\texttt{NTV}} < \texttt{f}_{\texttt{STV}}$$

Accordion Basics

- How to close the gap between NTC and STC execution times?

$$\texttt{Execution Time} \propto \frac{\texttt{Problem Size}}{\texttt{f} \times \texttt{Core Count}}$$

$$\frac{\texttt{Problem Size}_{\texttt{NTV}}}{\texttt{f}_{\texttt{NTV}} \times \texttt{Core Count}_{\texttt{NTV}}} \rightarrow \frac{\texttt{Problem Size}_{\texttt{STV}}}{\texttt{f}_{\texttt{STV}} \times \texttt{Core Count}_{\texttt{STV}}}$$

$$\texttt{f}_{\texttt{NTV}} < \texttt{f}_{\texttt{STV}} \qquad \texttt{Core Count}_{\texttt{NTV}} > \texttt{Core Count}_{\texttt{STV}}$$

Accordion Basics

- How to close the gap between NTC and STC execution times?

$$\texttt{Execution Time} \propto \frac{\texttt{Problem Size}}{\texttt{f} \times \texttt{Core Count}}$$

$$\frac{\texttt{Problem Size}_{\texttt{NTV}}}{\texttt{f}_{\texttt{NTV}} \times \texttt{Core Count}_{\texttt{NTV}}} \rightarrow \frac{\texttt{Problem Size}_{\texttt{STV}}}{\texttt{f}_{\texttt{STV}} \times \texttt{Core Count}_{\texttt{STV}}}$$

$$\texttt{f}_{\texttt{NTV}} < \texttt{f}_{\texttt{STV}} \qquad \texttt{Core Count}_{\texttt{NTV}} > \texttt{Core Count}_{\texttt{STV}}$$

- Designate the problem size as the main knob to adjust

Accordion Basics

- How to close the gap between NTC and STC execution times?

$$\texttt{Execution Time} \propto \frac{\texttt{Problem Size}}{\texttt{f} \times \texttt{Core Count}}$$

$$\frac{\texttt{Problem Size}_{\texttt{NTV}}}{\texttt{f}_{\texttt{NTV}} \times \texttt{Core Count}_{\texttt{NTV}}} \rightarrow \frac{\texttt{Problem Size}_{\texttt{STV}}}{\texttt{f}_{\texttt{STV}} \times \texttt{Core Count}_{\texttt{STV}}}$$

$$\texttt{f}_{\texttt{NTV}} < \texttt{f}_{\texttt{STV}} \qquad \texttt{Core Count}_{\texttt{NTV}} > \texttt{Core Count}_{\texttt{STV}}$$

- Designate the problem size as the main knob to adjust
  - the degree of parallelism

Accordion Basics

- How to close the gap between NTC and STC execution times?

$$\texttt{Execution Time} \propto \frac{\texttt{Problem Size}}{\texttt{f} \times \texttt{Core Count}}$$

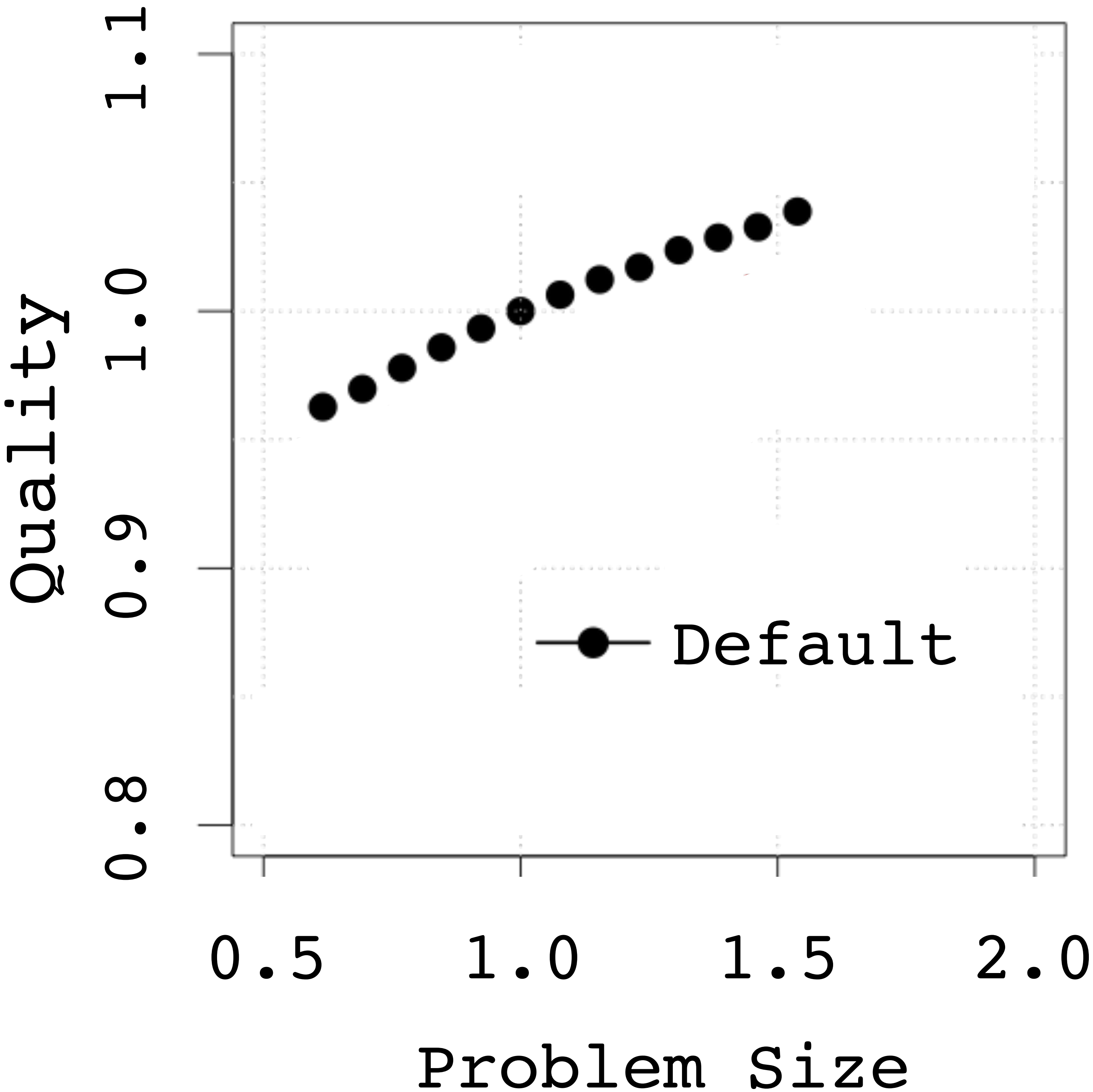
$$\frac{\texttt{Problem Size}_{\texttt{NTV}}}{\texttt{f}_{\texttt{NTV}} \times \texttt{Core Count}_{\texttt{NTV}}} \rightarrow \frac{\texttt{Problem Size}_{\texttt{STV}}}{\texttt{f}_{\texttt{STV}} \times \texttt{Core Count}_{\texttt{STV}}}$$

$$\texttt{f}_{\texttt{NTV}} < \texttt{f}_{\texttt{STV}} \qquad \texttt{Core Count}_{\texttt{NTV}} > \texttt{Core Count}_{\texttt{STV}}$$

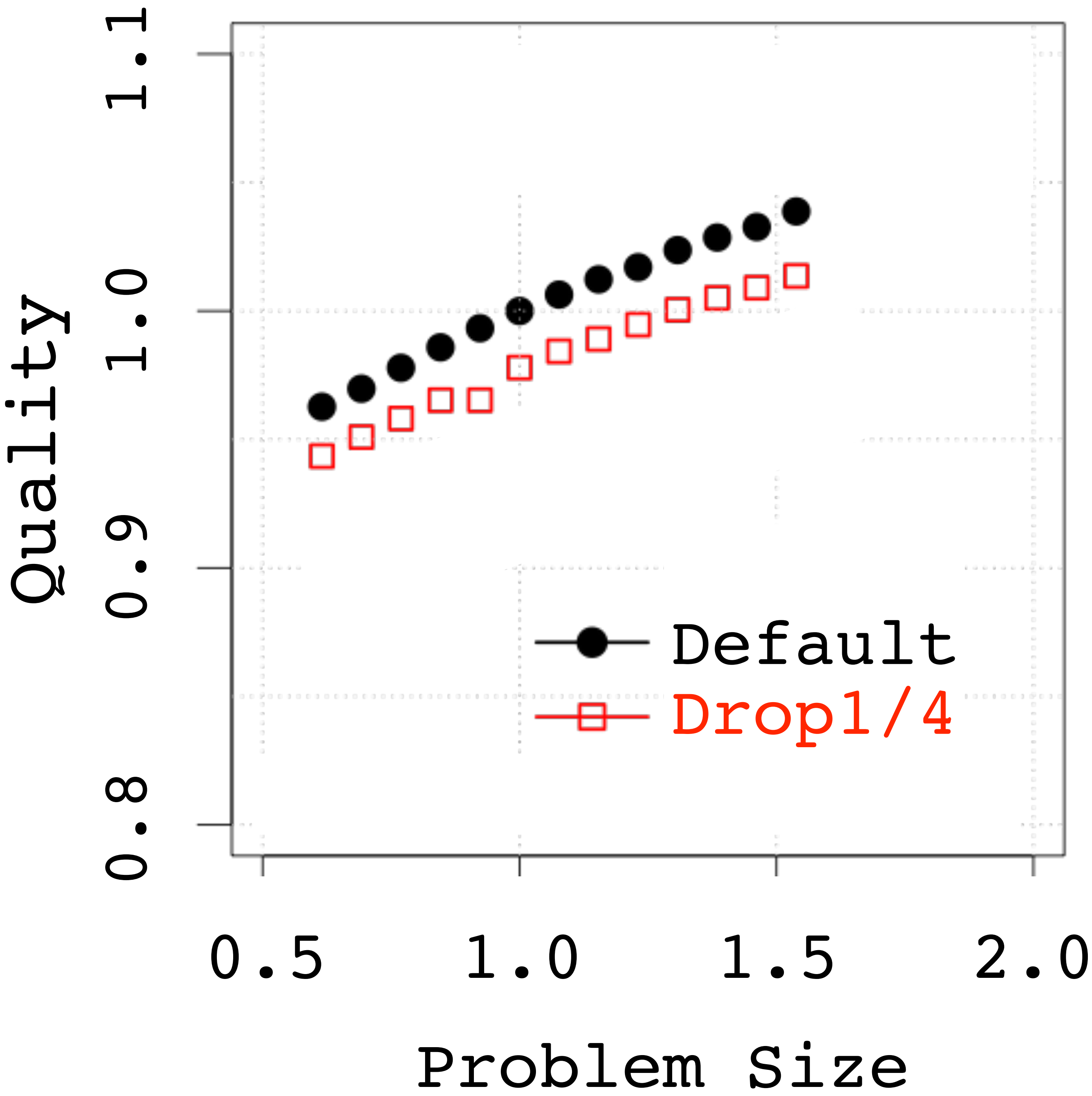
- Designate the problem size as the main knob to adjust
  - the degree of parallelism
  - the degree of vulnerability to variation

Problem Size vs. Quality of Computing

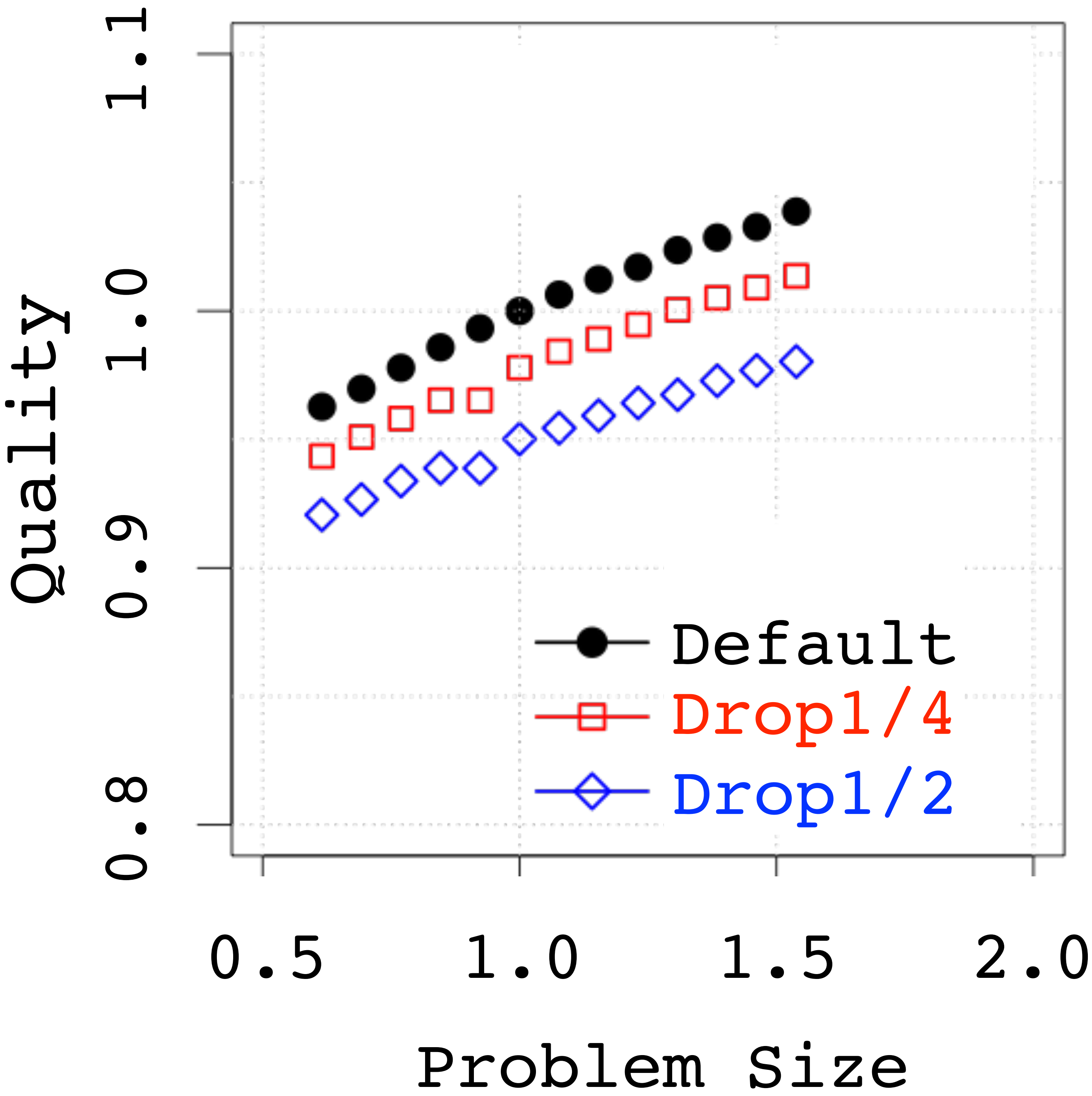
Problem Size vs. Quality of Computing



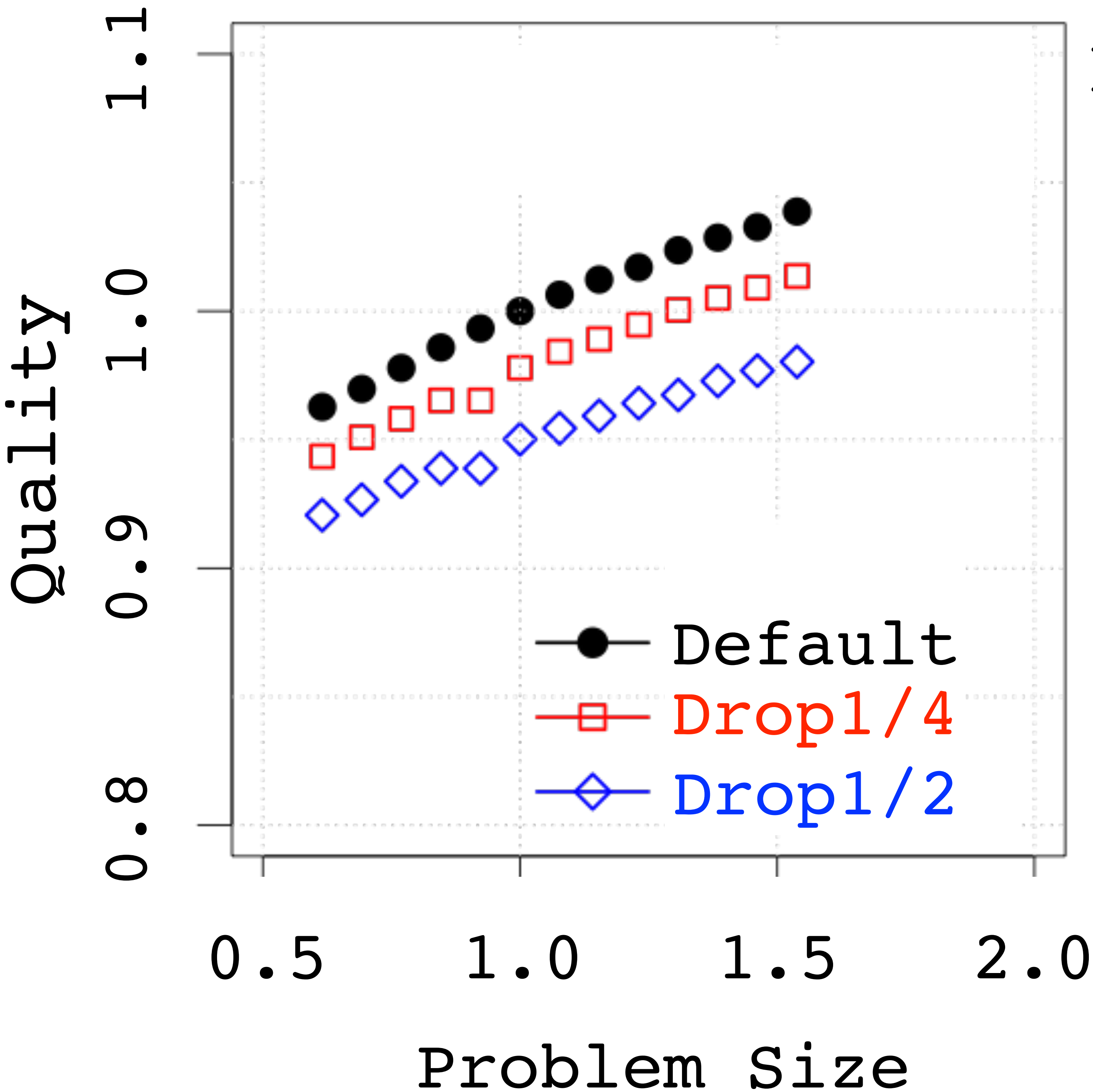
Problem Size vs. Quality of Computing



Problem Size vs. Quality of Computing



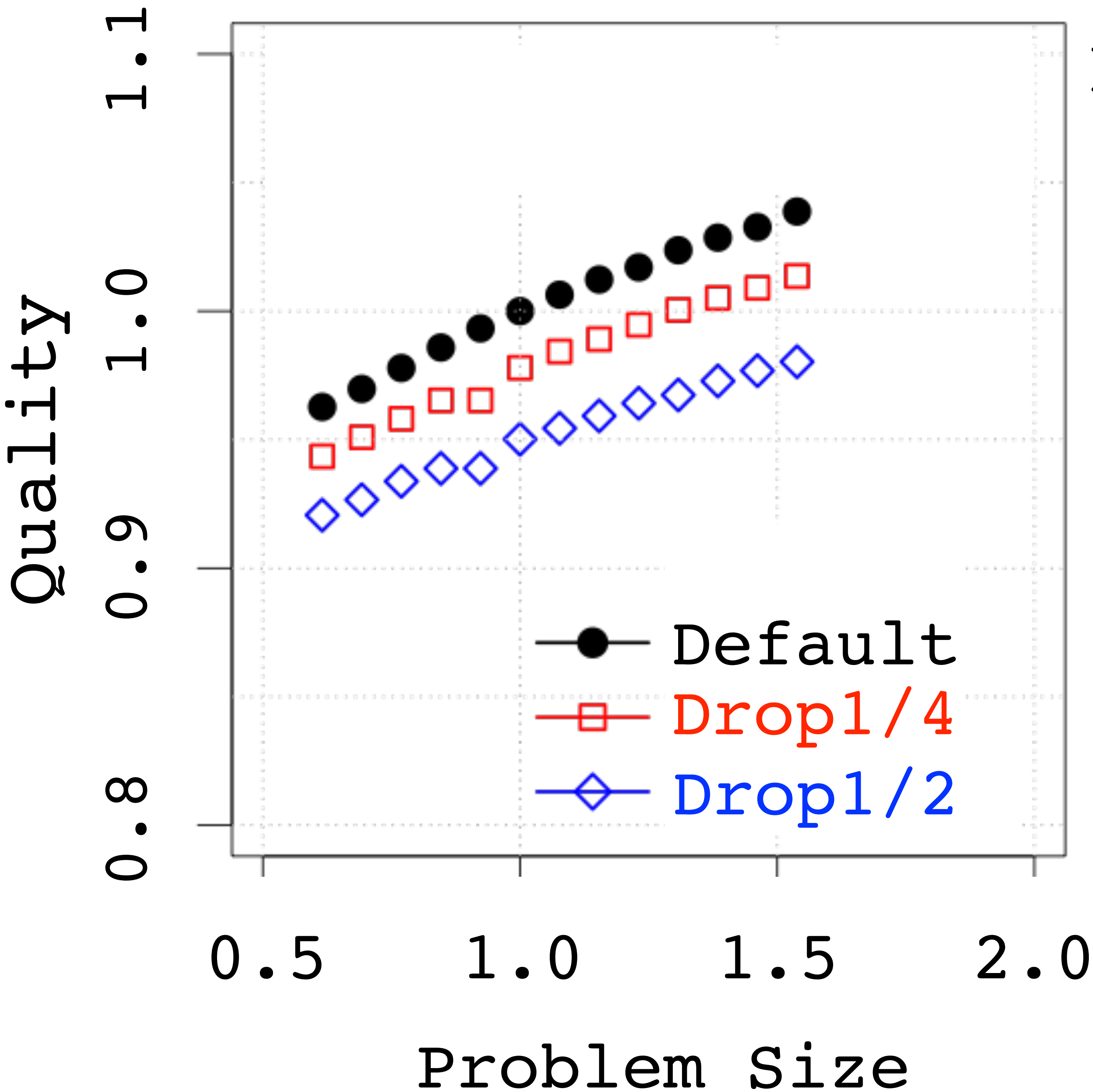
Problem Size vs. Quality of Computing



Execution Time

$$\propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

Problem Size vs. Quality of Computing

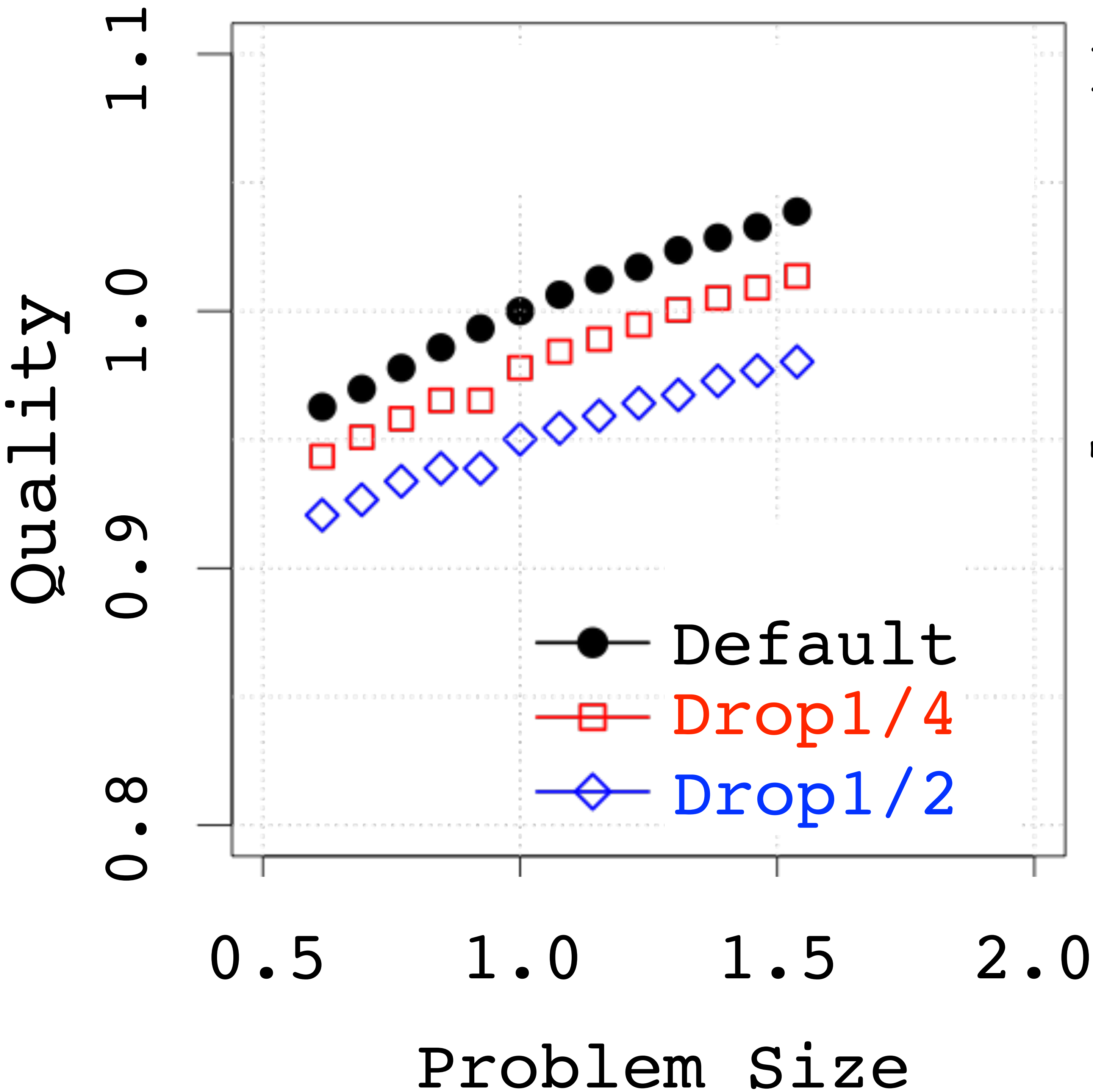


Execution Time

$$\propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

⬇

Problem Size vs. Quality of Computing



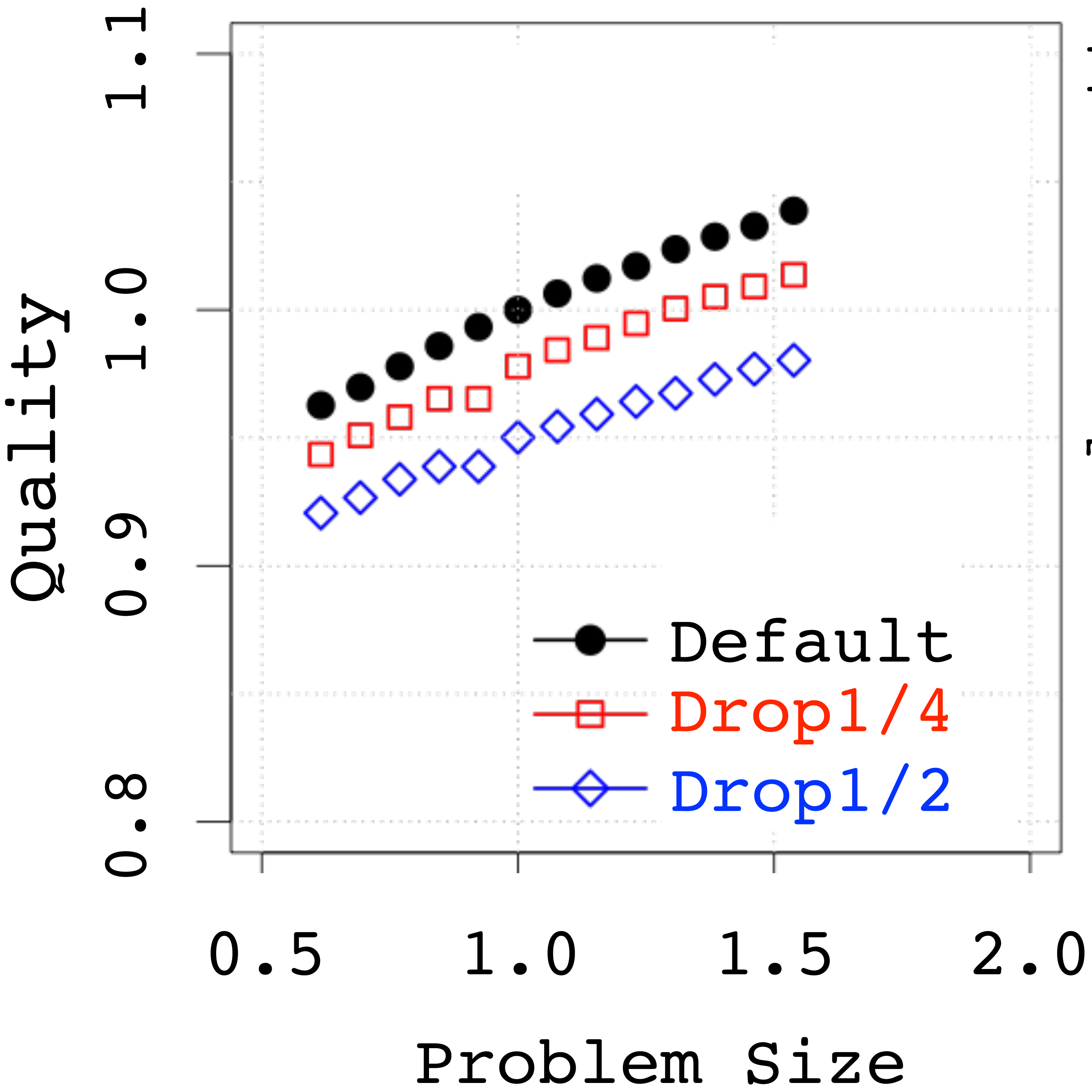
Execution Time

$$\propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

⬇

To make up for f degradation:

Problem Size vs. Quality of Computing



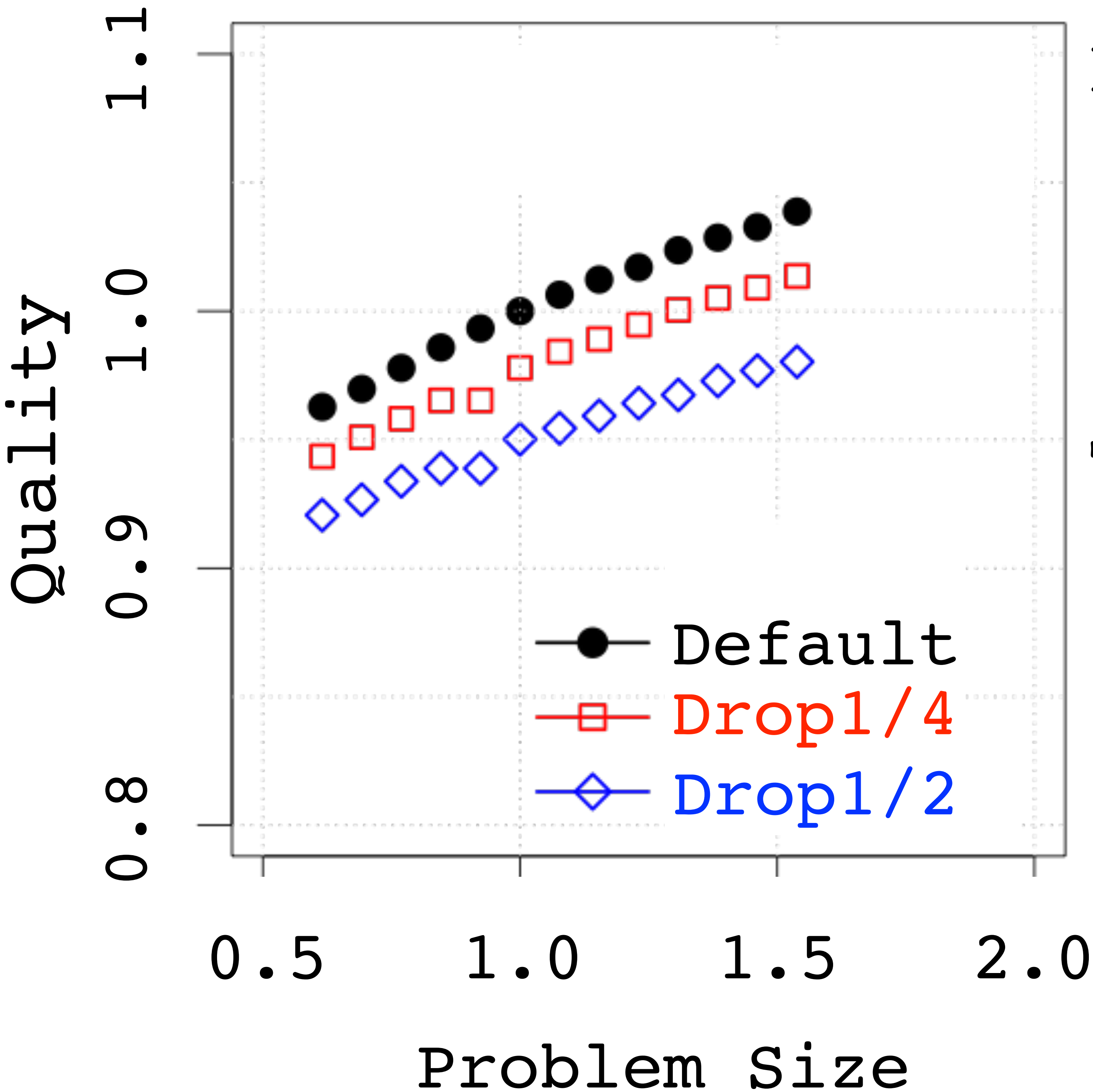
Execution Time

$$\propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

To make up for f degradation:

- Compress problem size

Problem Size vs. Quality of Computing



Execution Time

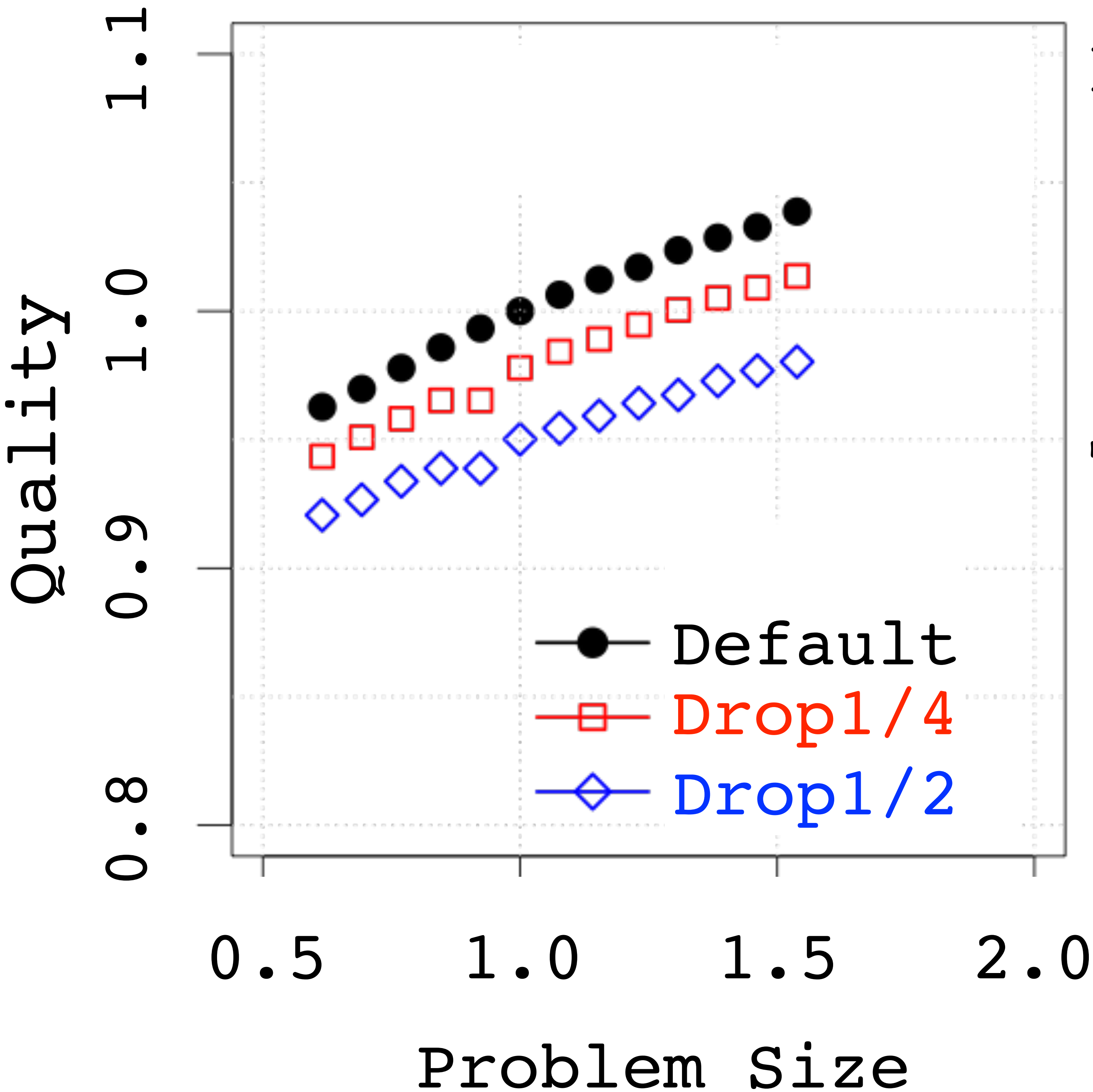
$$\propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

⬇

To make up for f degradation:

- Compress problem size
- Expand problem size

Problem Size vs. Quality of Computing



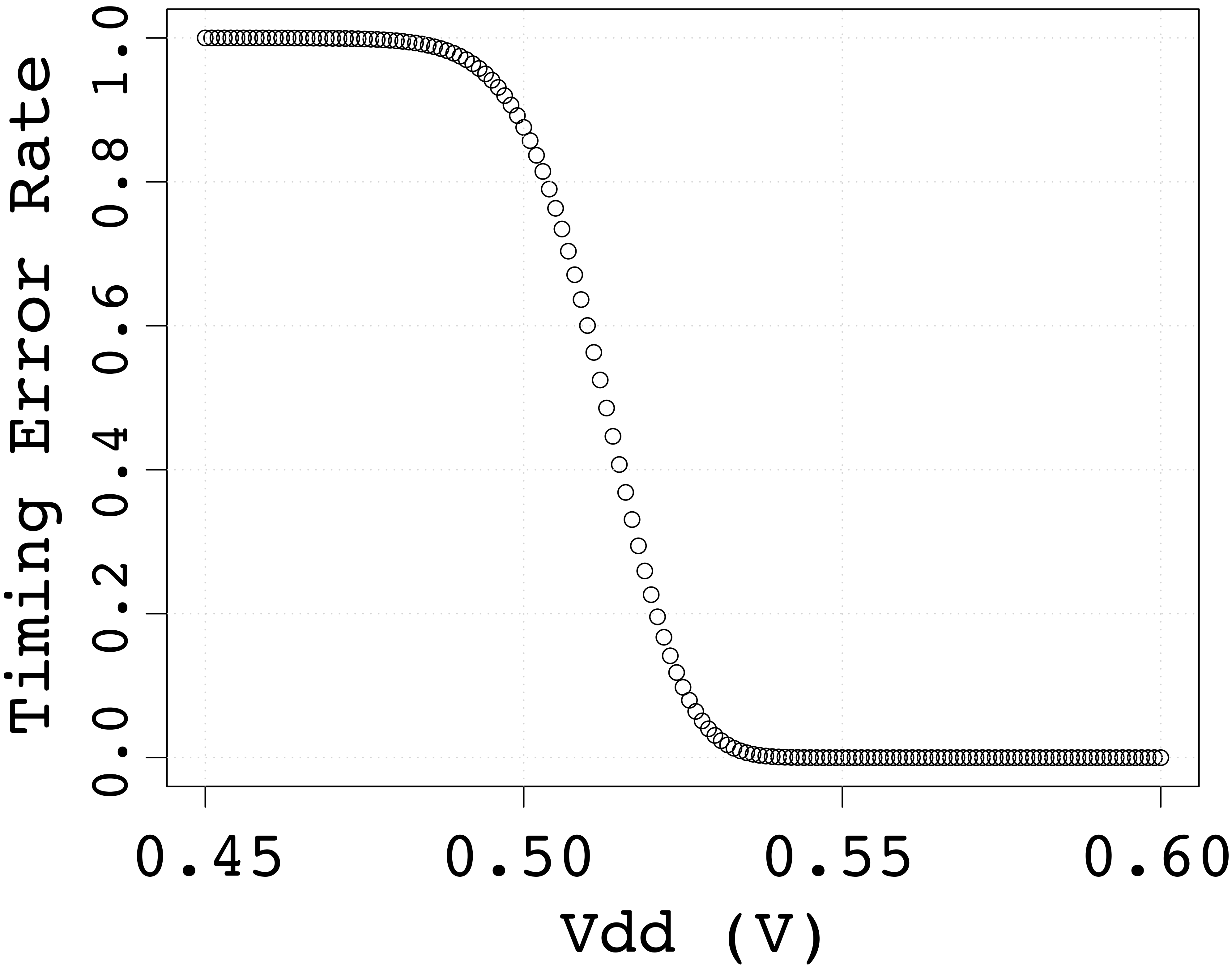
$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

⬇

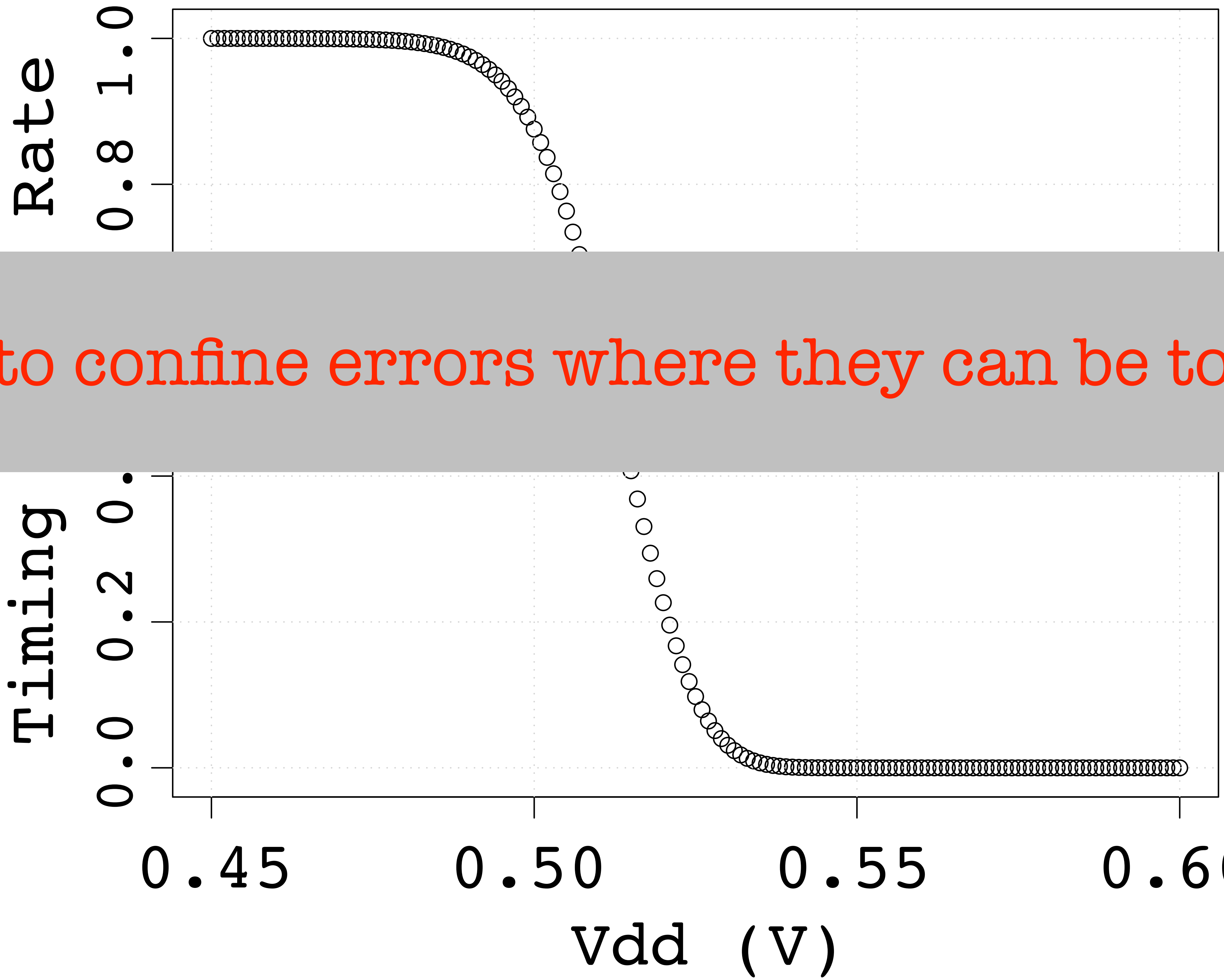
To make up for f degradation:

- Compress problem size
- Expand problem size
  - Core Count should expand

Variation Induced Errors

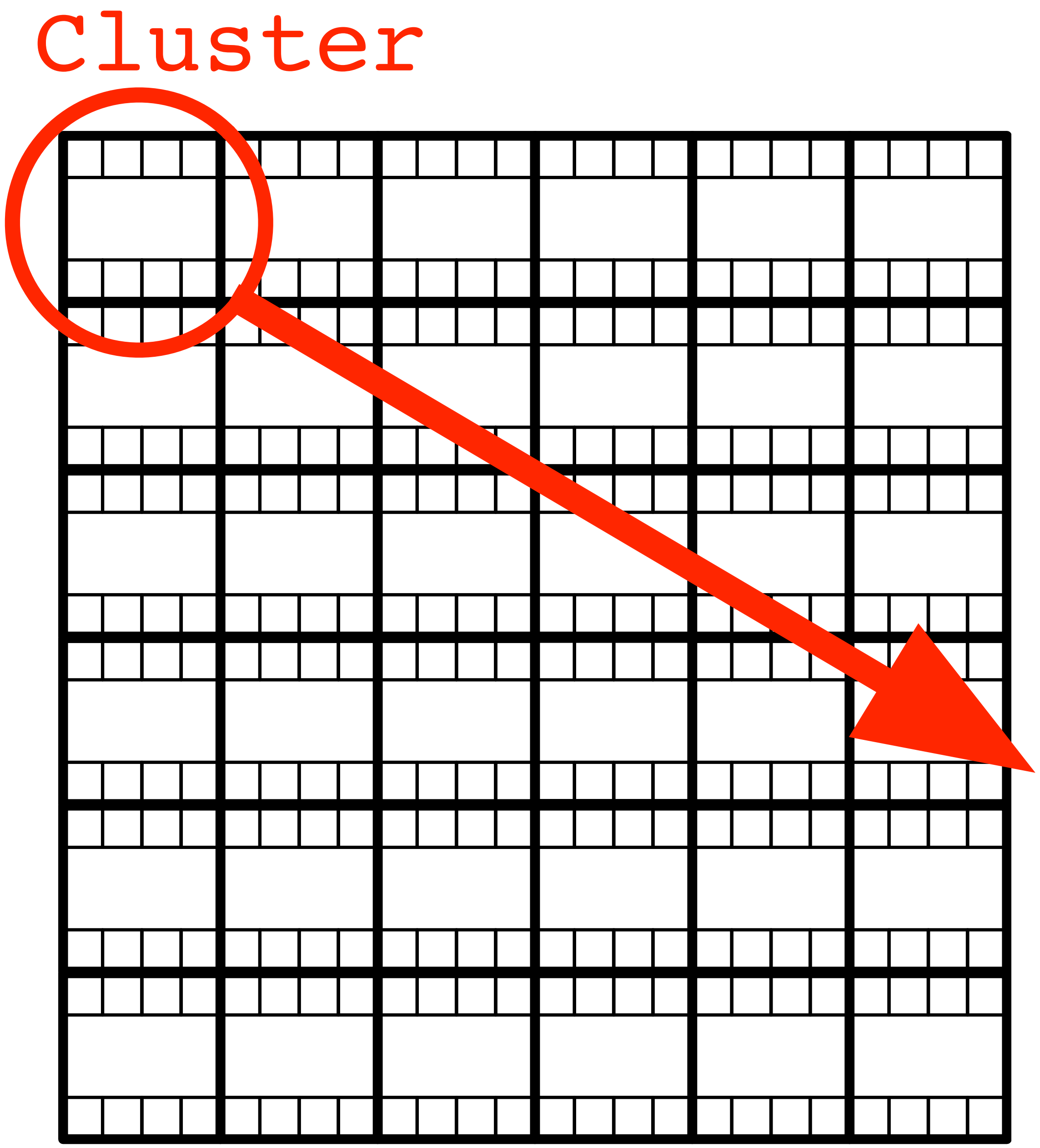


Variation Induced Errors

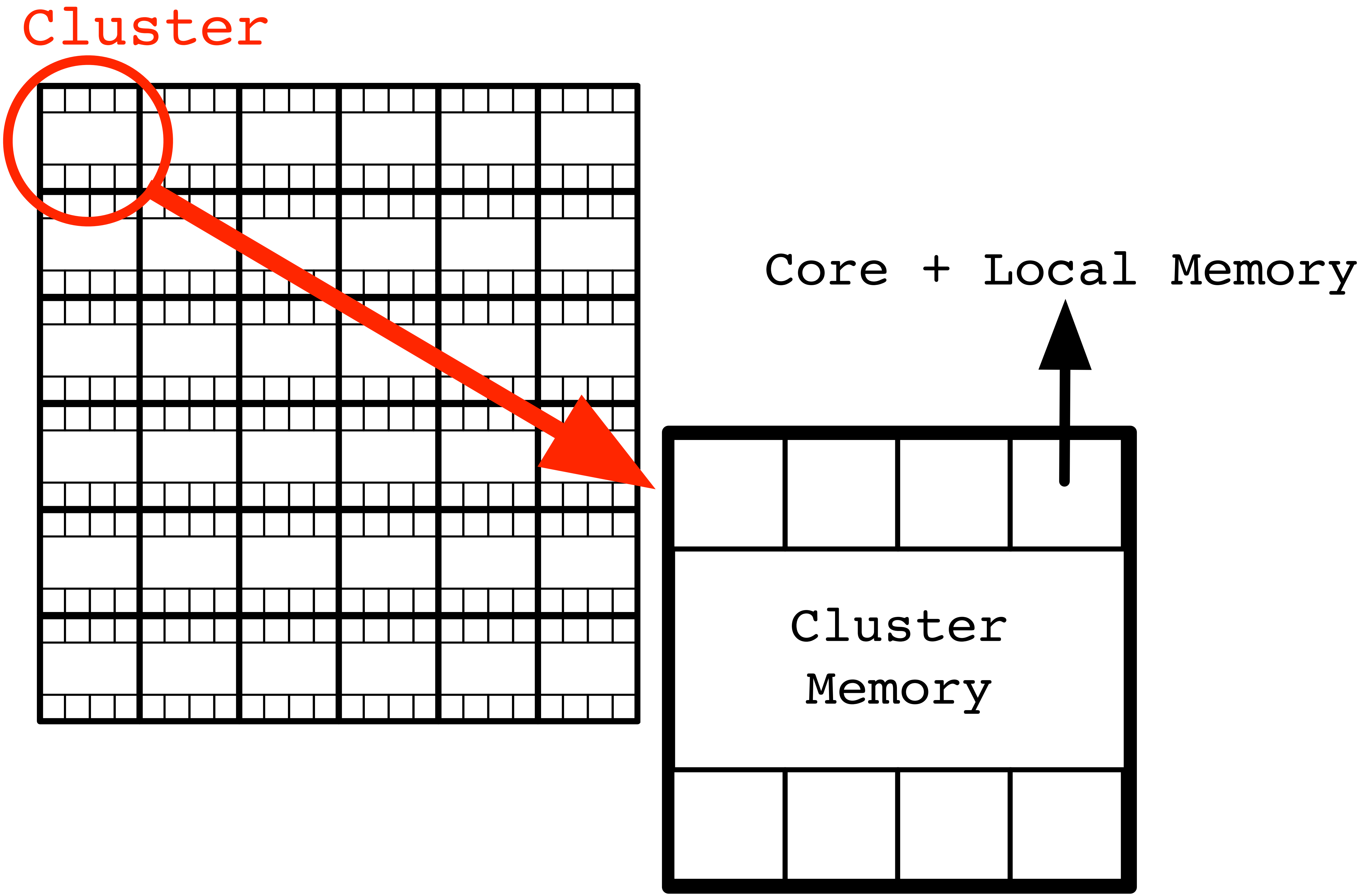


How to confine errors where they can be tolerated?

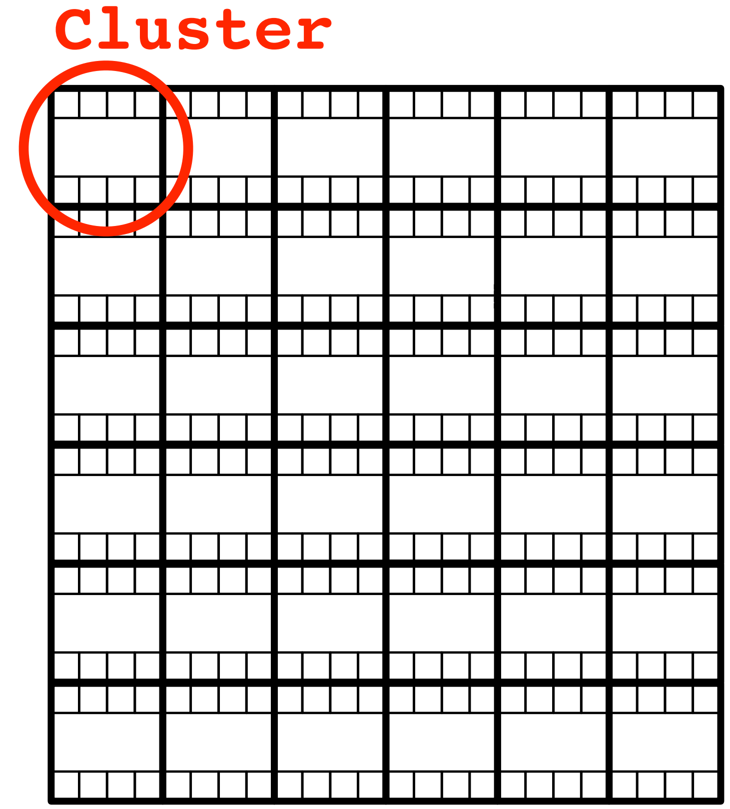
Accordion Organization



Accordion Organization

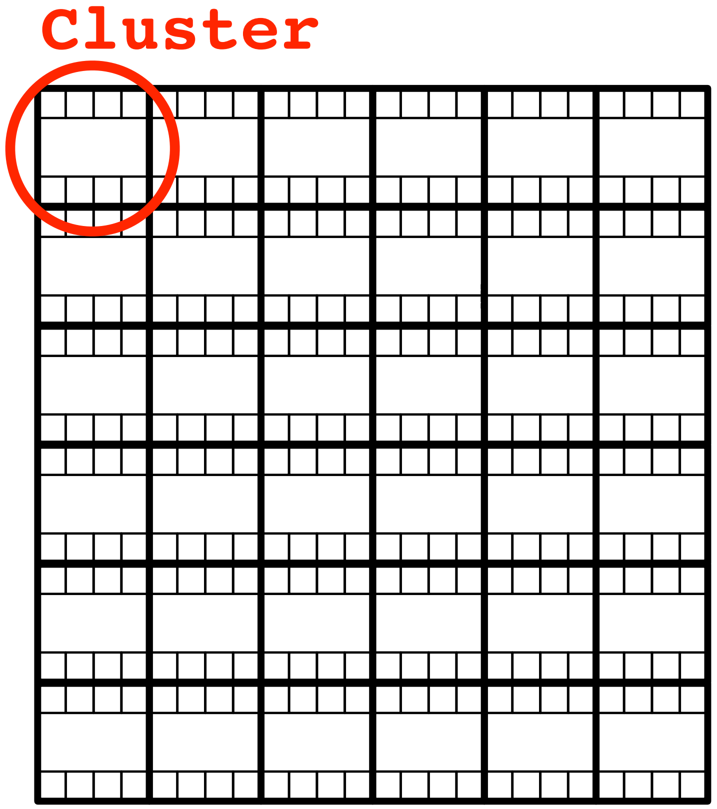


Accordion Design Space

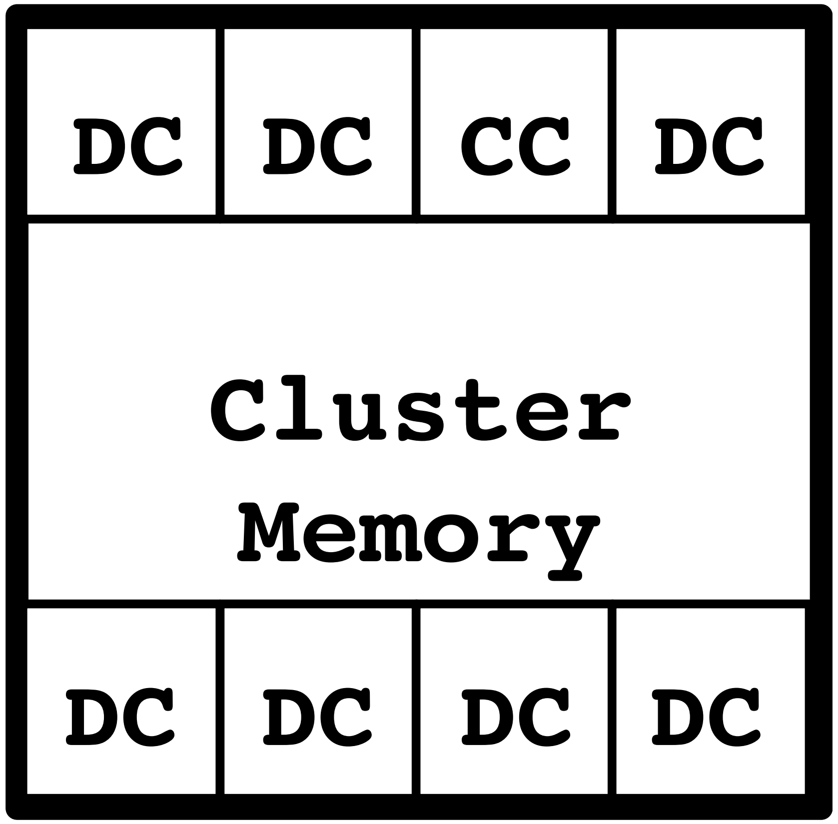


Accordion Design Space

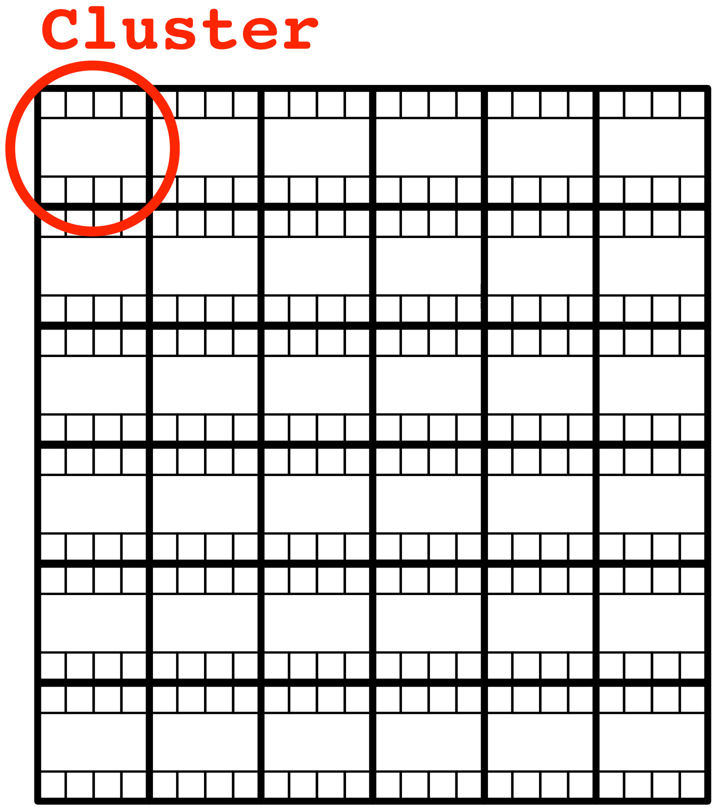
CC: Control Core
DC: Data Core



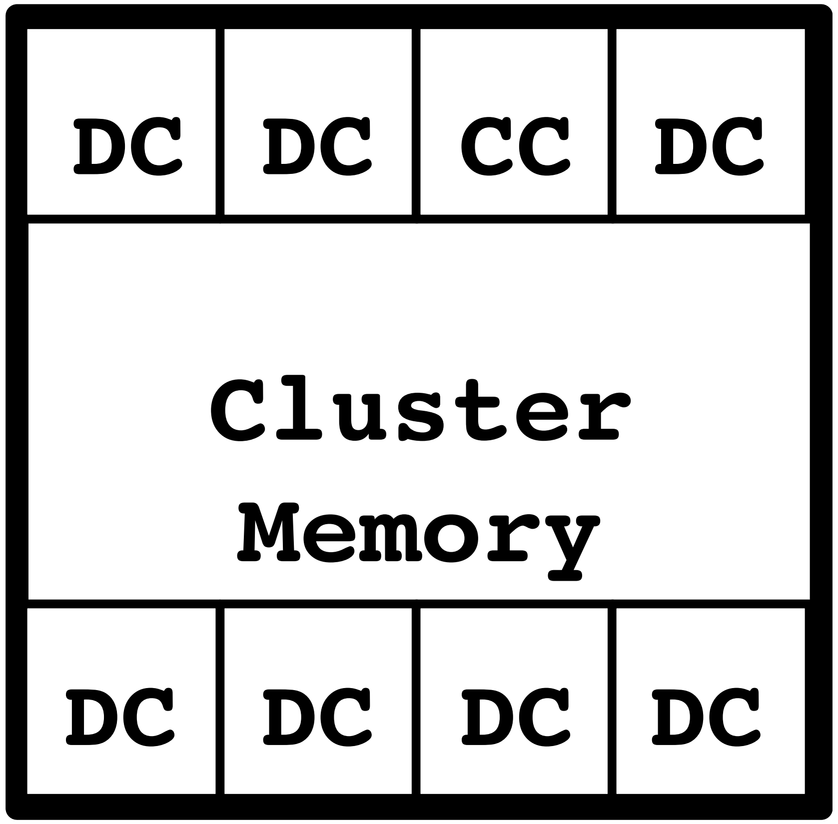
Accordion Design Space

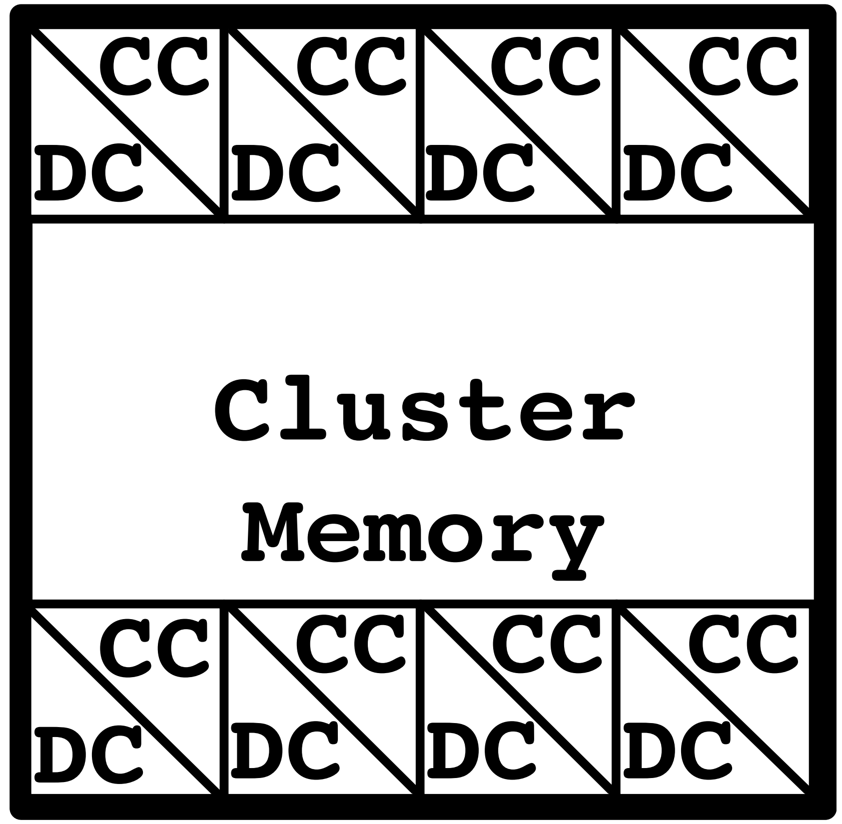


CC: Control Core
DC: Data Core

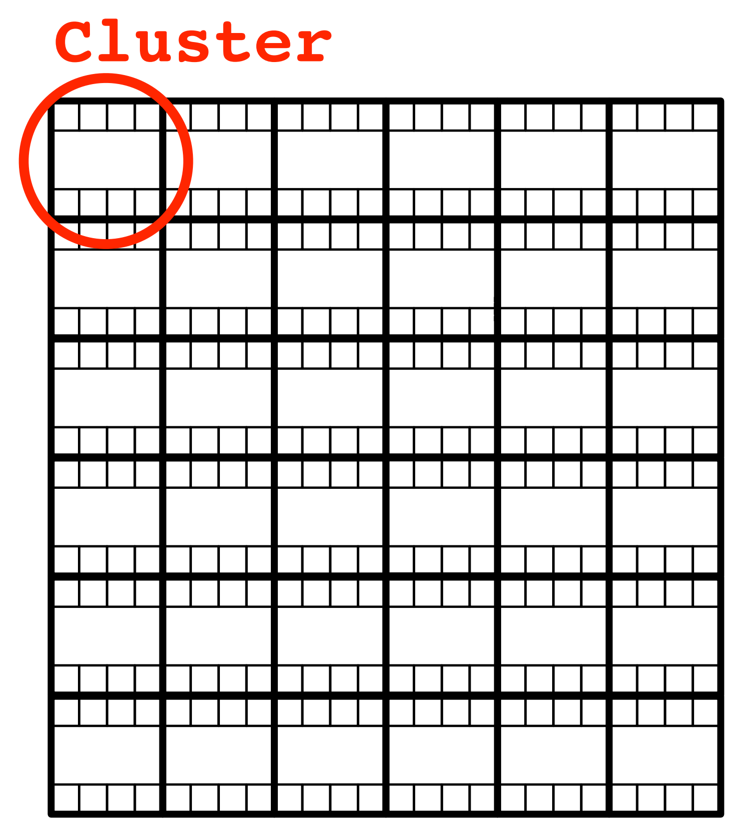


Accordion Design Space

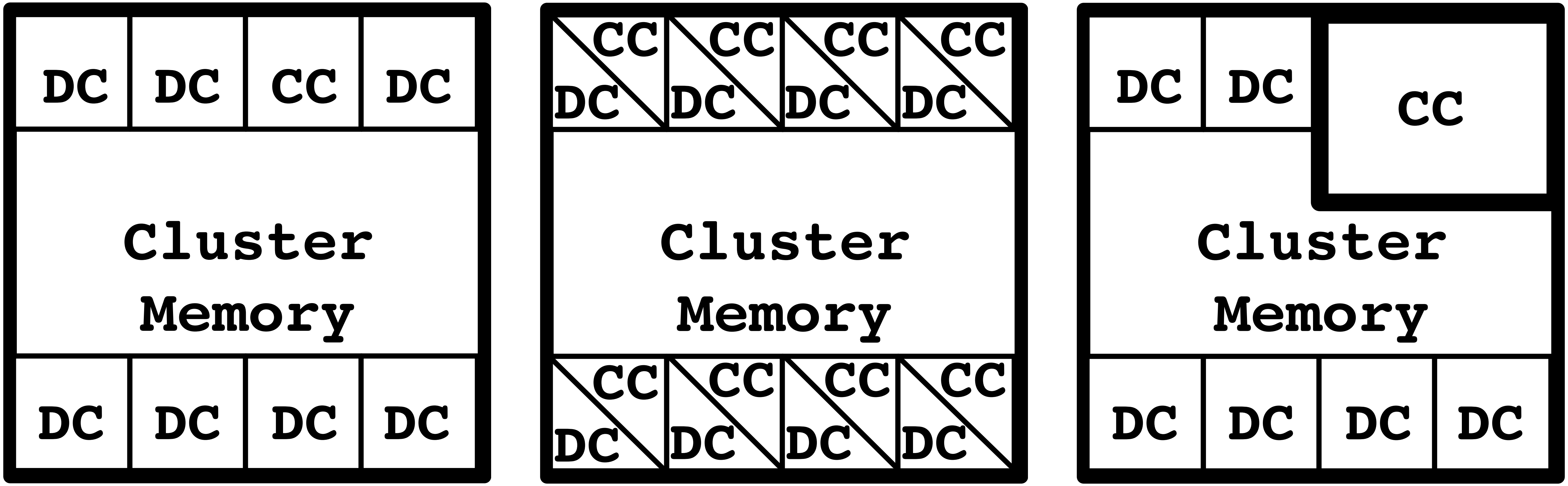




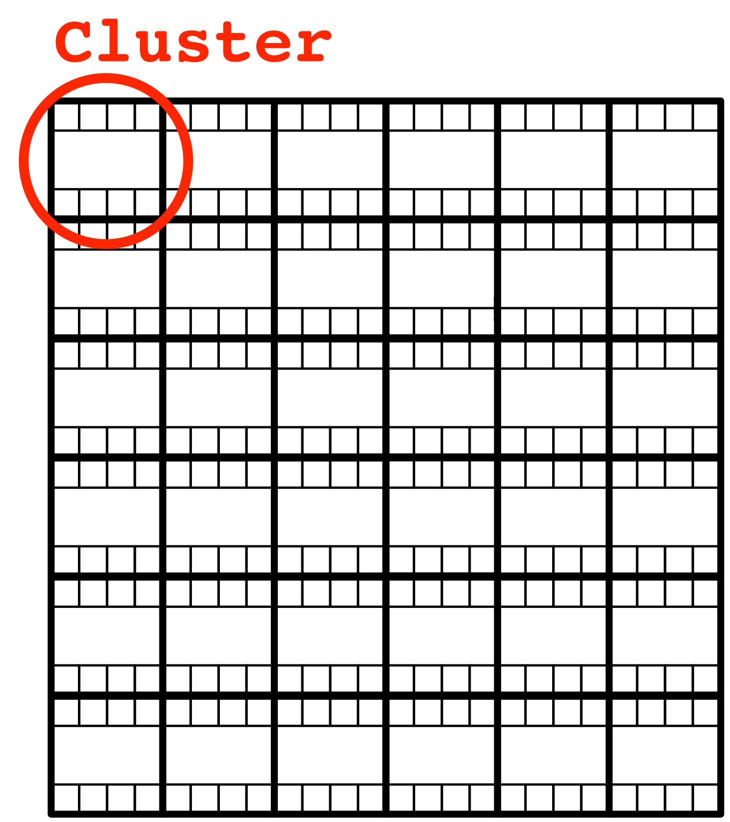
CC: Control Core
DC: Data Core



Accordion Design Space



CC: Control Core
DC: Data Core



Accordion Modes of Operation

$$\frac{\text{Problem Size}_{\text{NTV}}}{f_{\text{NTV}} \times \text{Core Count}_{\text{NTV}}} \rightarrow \frac{\text{Problem Size}_{\text{STV}}}{f_{\text{STV}} \times \text{Core Count}_{\text{STV}}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV	Quality

$$\frac{\text{Problem Size}_{\text{NTV}}}{f_{\text{NTV}} \times \text{Core Count}_{\text{NTV}}} \rightarrow \frac{\text{Problem Size}_{\text{STV}}}{f_{\text{STV}} \times \text{Core Count}_{\text{STV}}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV	Quality
Still				

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV	Quality
Still	= STV			

$$\frac{\text{Problem Size}_{\text{NTV}}}{f_{\text{NTV}} \times \text{Core Count}_{\text{NTV}}} \rightarrow \frac{\text{Problem Size}_{\text{STV}}}{f_{\text{STV}} \times \text{Core Count}_{\text{STV}}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV	Quality
Still	= STV	= STV		

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV	Quality
Still	= STV	= STV	≤ NTV	

Safe

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality
Still	= STV	= STV	≤ NTV	> NTV	

Safe

Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality
Still	= STV	= STV	≤ NTV	> NTV	= STV

Safe

Speculative

Safe

$$\frac{\text{Problem Size}_{\text{NTV}}}{f_{\text{NTV}} \times \text{Core Count}_{\text{NTV}}} \rightarrow \frac{\text{Problem Size}_{\text{STV}}}{f_{\text{STV}} \times \text{Core Count}_{\text{STV}}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV

Safe Speculative Safe Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress						
			Safe	Speculative	Safe	Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	**Problem Size**	**Core Count**	**f < STV**		**Quality**	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV					

Safe Speculative Safe Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction				

Safe Speculative Safe Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction	≤ NTV			

Safe Speculative Safe Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction	≤ NTV	> NTV		

Safe Speculative Safe Speculative

$$\frac{\text{Problem Size}_{\text{NTV}}}{f_{\text{NTV}} \times \text{Core Count}_{\text{NTV}}} \rightarrow \frac{\text{Problem Size}_{\text{STV}}}{f_{\text{STV}} \times \text{Core Count}_{\text{STV}}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction	≤ NTV	> NTV	≤ STV	

Safe Speculative Safe Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction	≤ NTV	> NTV	≤ STV	≤ STV

Safe Speculative Safe Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction	≤ NTV	> NTV	≤ STV	≤ STV
Expand						

Safe Speculative Safe Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction	≤ NTV	> NTV	≤ STV	≤ STV
Expand	> STV					
			Safe	Speculative	Safe	Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction	≤ NTV	> NTV	≤ STV	≤ STV
Expand	> STV	> STV				
			Safe	Speculative	Safe	Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction	≤ NTV	> NTV	≤ STV	≤ STV
Expand	> STV	> STV	≤ NTV			

Safe

Speculative

Safe

Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction	≤ NTV	> NTV	≤ STV	≤ STV
Expand	> STV	> STV	≤ NTV	> NTV		

Safe | Speculative | Safe | Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction	≤ NTV	> NTV	≤ STV	≤ STV
Expand	> STV	> STV	≤ NTV	> NTV	> STV	

Safe Speculative Safe Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Modes of Operation

Mode	Problem Size	Core Count	f < STV		Quality	
Still	= STV	= STV	≤ NTV	> NTV	= STV	≤ STV
Compress	< STV	No restriction	≤ NTV	> NTV	≤ STV	≤ STV
Expand	> STV	> STV	≤ NTV	> NTV	> STV	≤ STV
			Safe	Speculative	Safe	Speculative

$$\frac{\text{Problem Size}_{NTV}}{f_{NTV} \times \text{Core Count}_{NTV}} \rightarrow \frac{\text{Problem Size}_{STV}}{f_{STV} \times \text{Core Count}_{STV}}$$

Accordion Pros and Cons

Accordion Pros and Cons

- Accommodates a closer Vdd to Vth without compromising performance

Accordion Pros and Cons

- Accommodates a closer Vdd to Vth without compromising performance
- Tolerates exacerbated variation as Vdd reaches Vth

Accordion Pros and Cons

- Accommodates a closer Vdd to Vth without compromising performance
- Tolerates exacerbated variation as Vdd reaches Vth
- Highly application-specific:

Accordion Pros and Cons

- Accommodates a closer Vdd to Vth without compromising performance
- Tolerates exacerbated variation as Vdd reaches Vth
- Highly application-specific:
  - Inputs to dictate the problem size

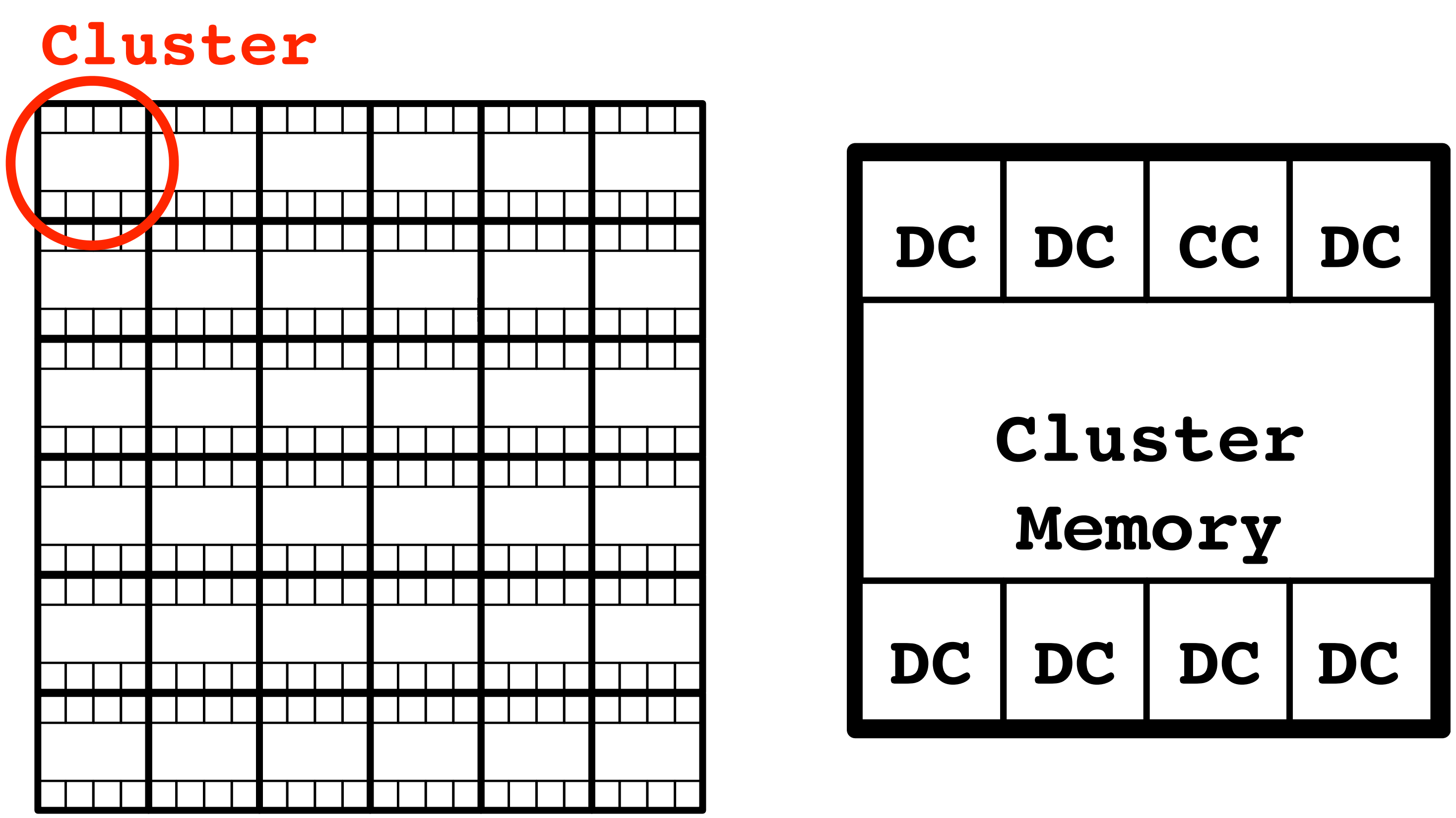
Accordion Pros and Cons

- Accommodates a closer Vdd to Vth without compromising performance
- Tolerates exacerbated variation as Vdd reaches Vth
- Highly application-specific:
  - Inputs to dictate the problem size
  - Quality metrics & thresholds

Evaluation Setup

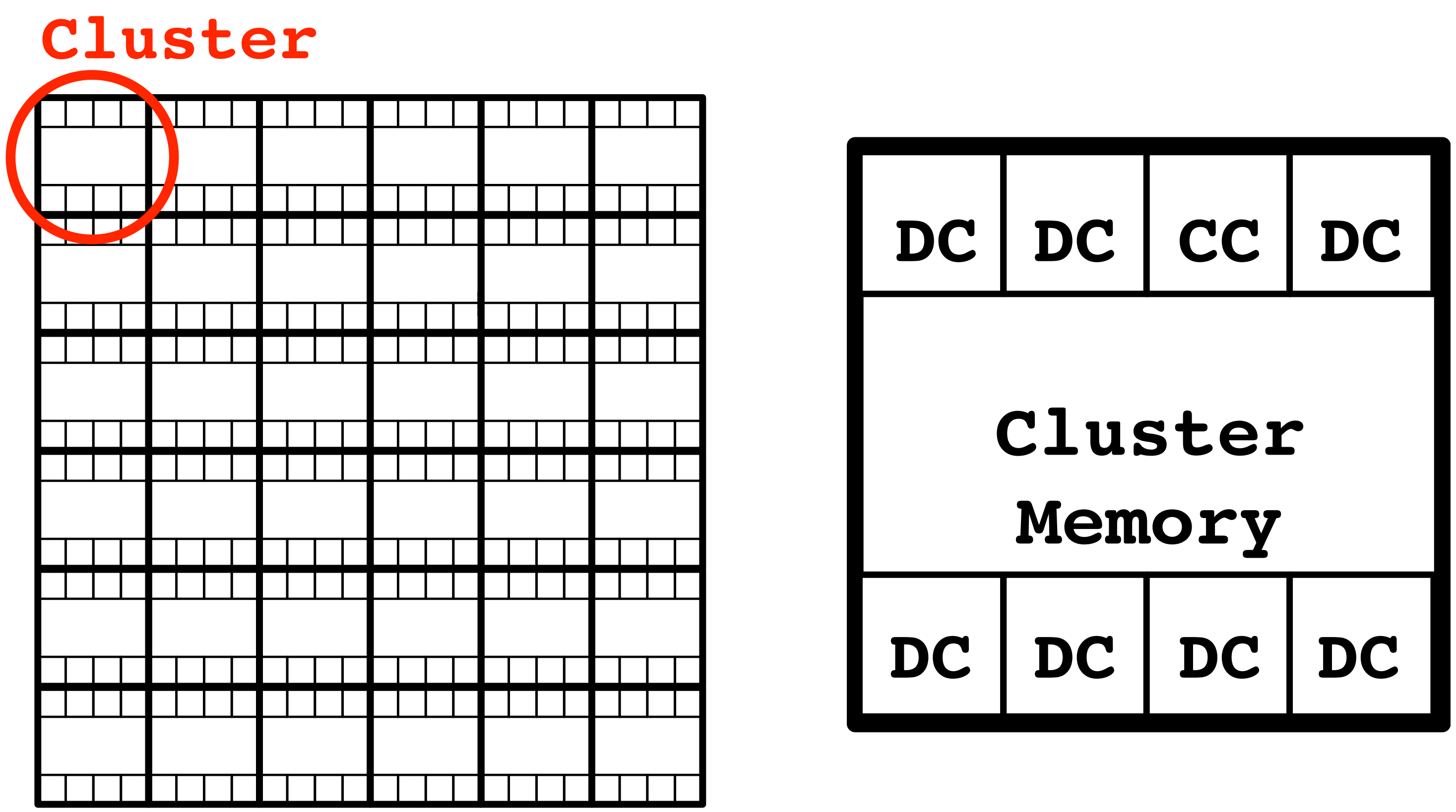
Evaluation Setup

- Simple, clustered hardware to exploit within die variation



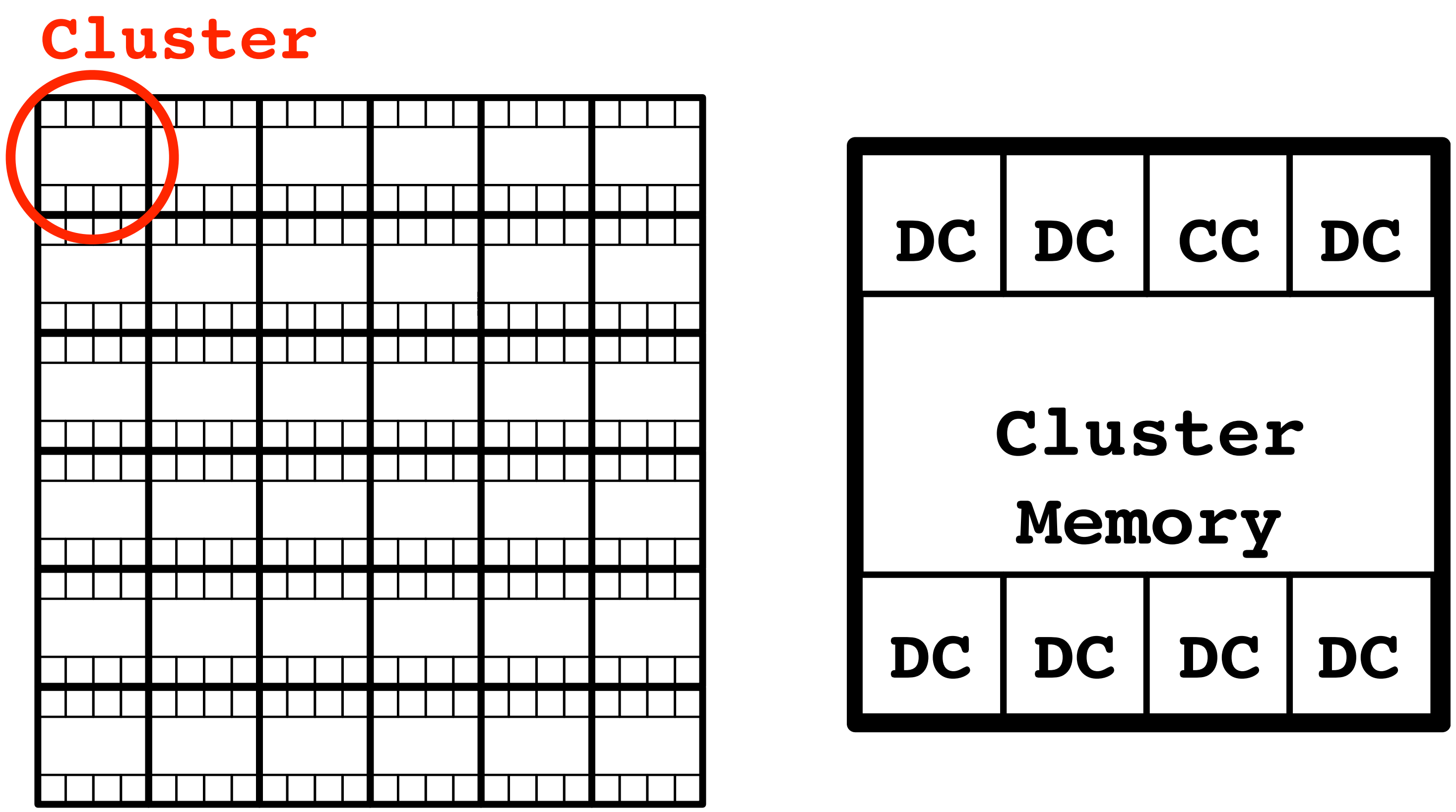
Evaluation Setup

- Simple, clustered hardware to exploit within die variation
  - Each cluster supports a different max f at chip-wide, single Vdd



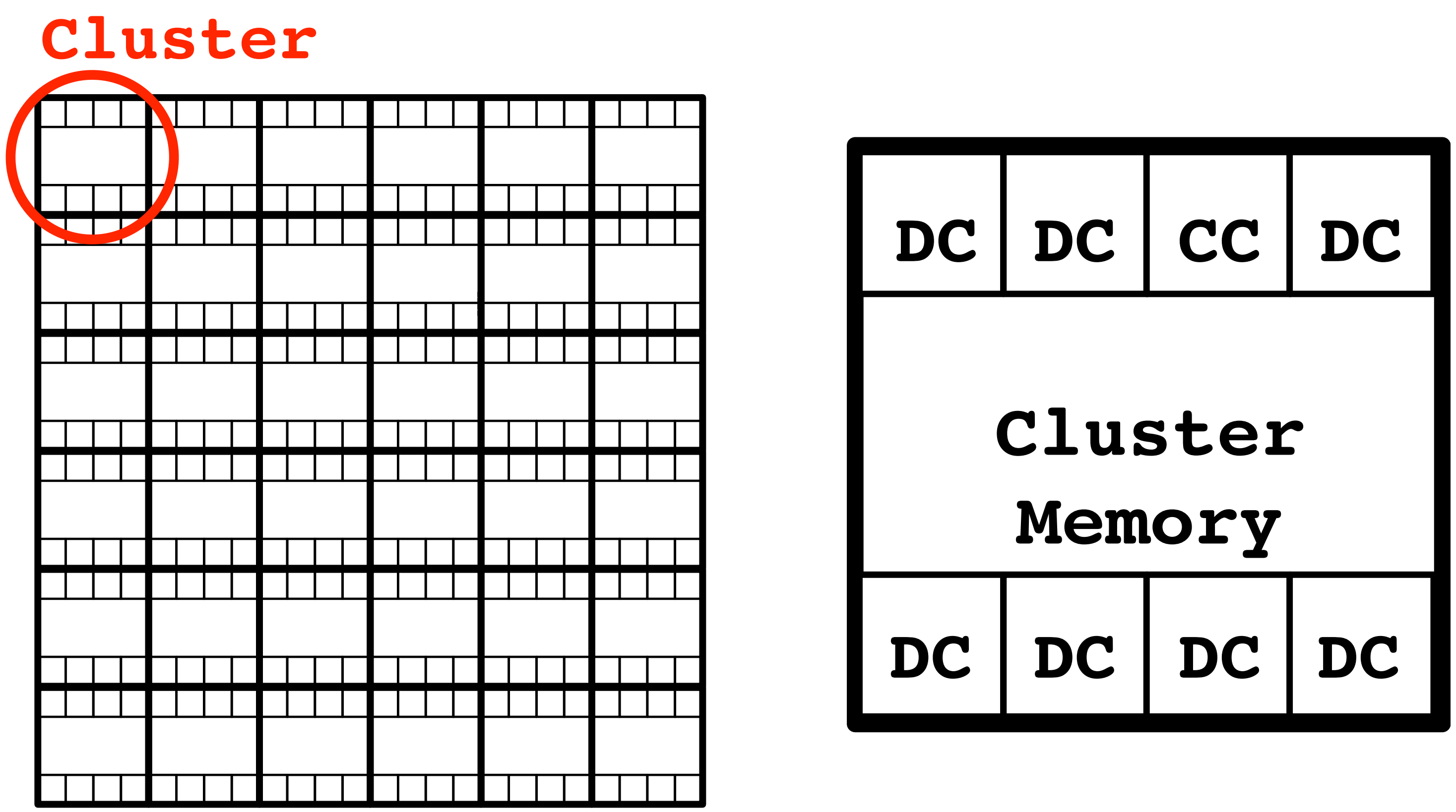
Evaluation Setup

- Simple, clustered hardware to exploit within die variation
  - Each cluster supports a different max f at chip-wide, single Vdd
  - All clusters assigned to a task cycle at the f of slowest cluster



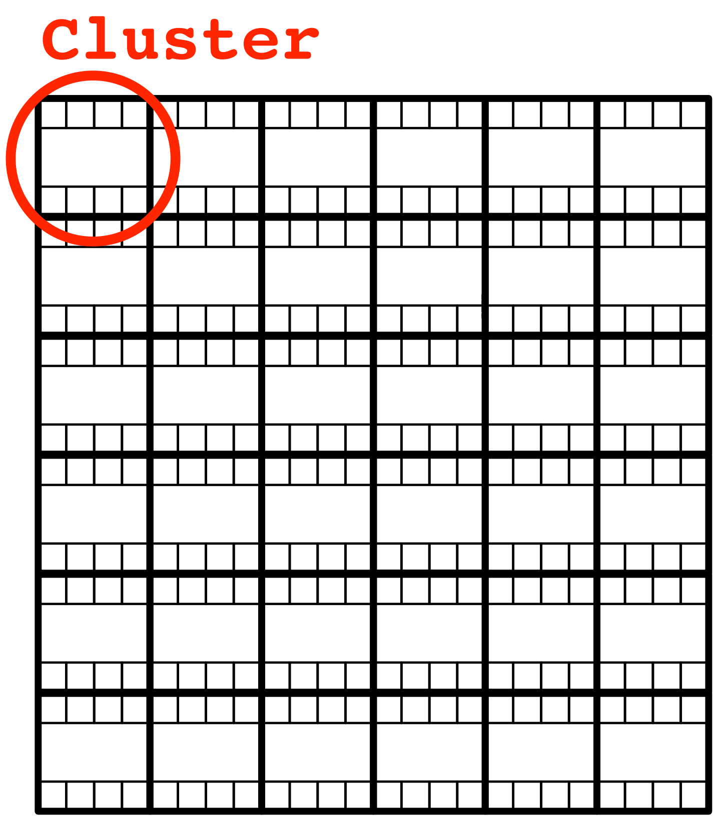
Evaluation Setup

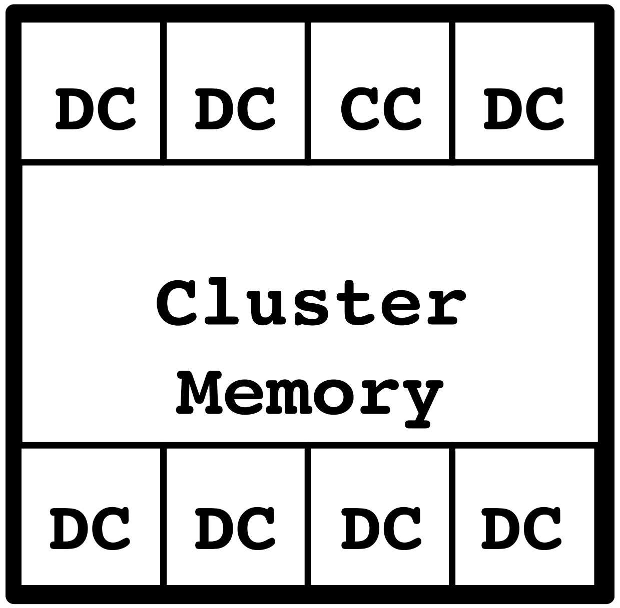
- Simple, clustered hardware to exploit within die variation
  - Each cluster supports a different max f at chip-wide, single Vdd
  - All clusters assigned to a task cycle at the f of slowest cluster
- Simulated 20mm x 20mm 288 core chip at 11nm



Evaluation Setup

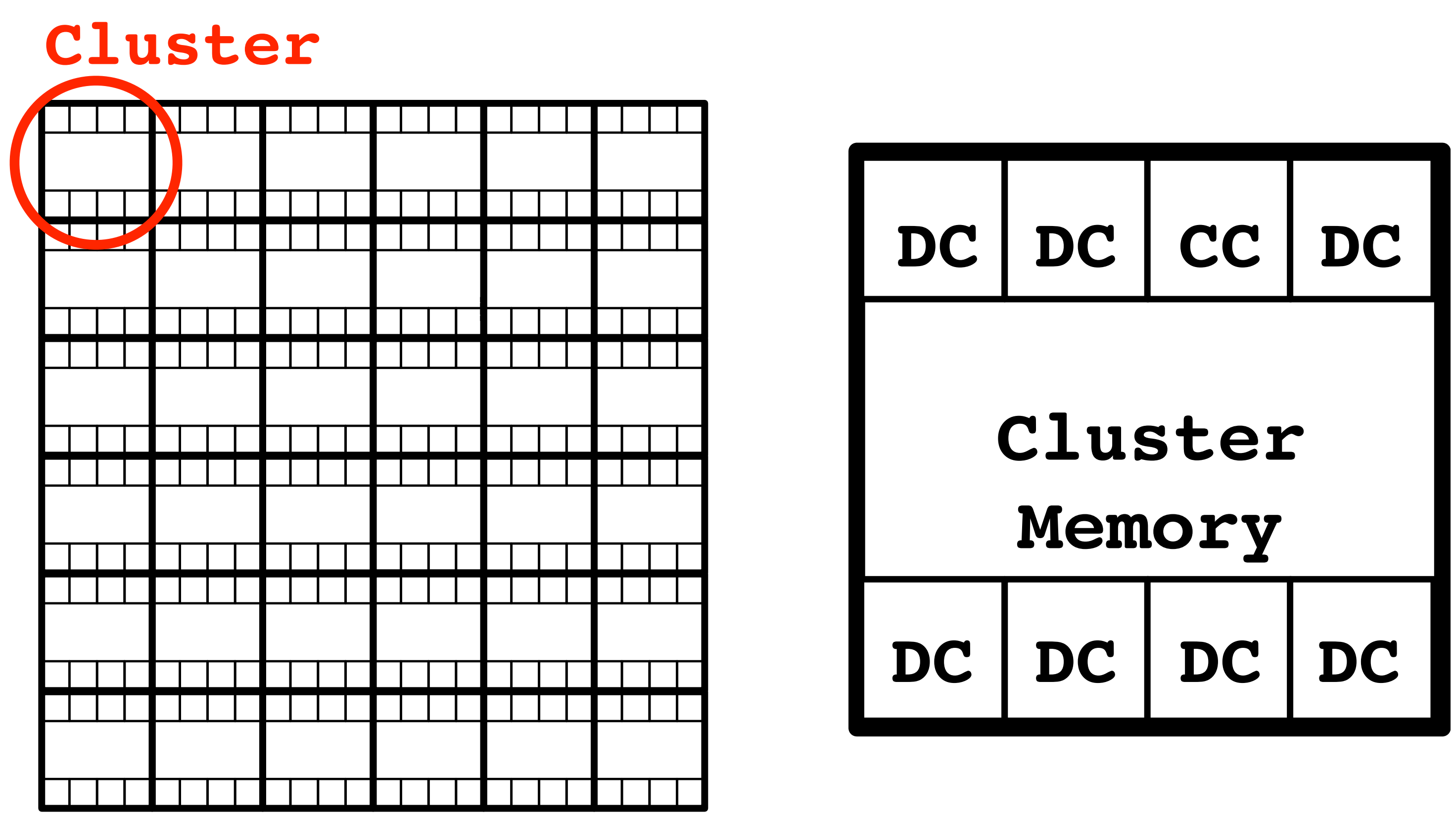
- Simple, clustered hardware to exploit within die variation
  - Each cluster supports a different max f at chip-wide, single Vdd
  - All clusters assigned to a task cycle at the f of slowest cluster
- Simulated 20mm x 20mm 288 core chip at 11nm
  - 36 clusters, 8 cores per cluster





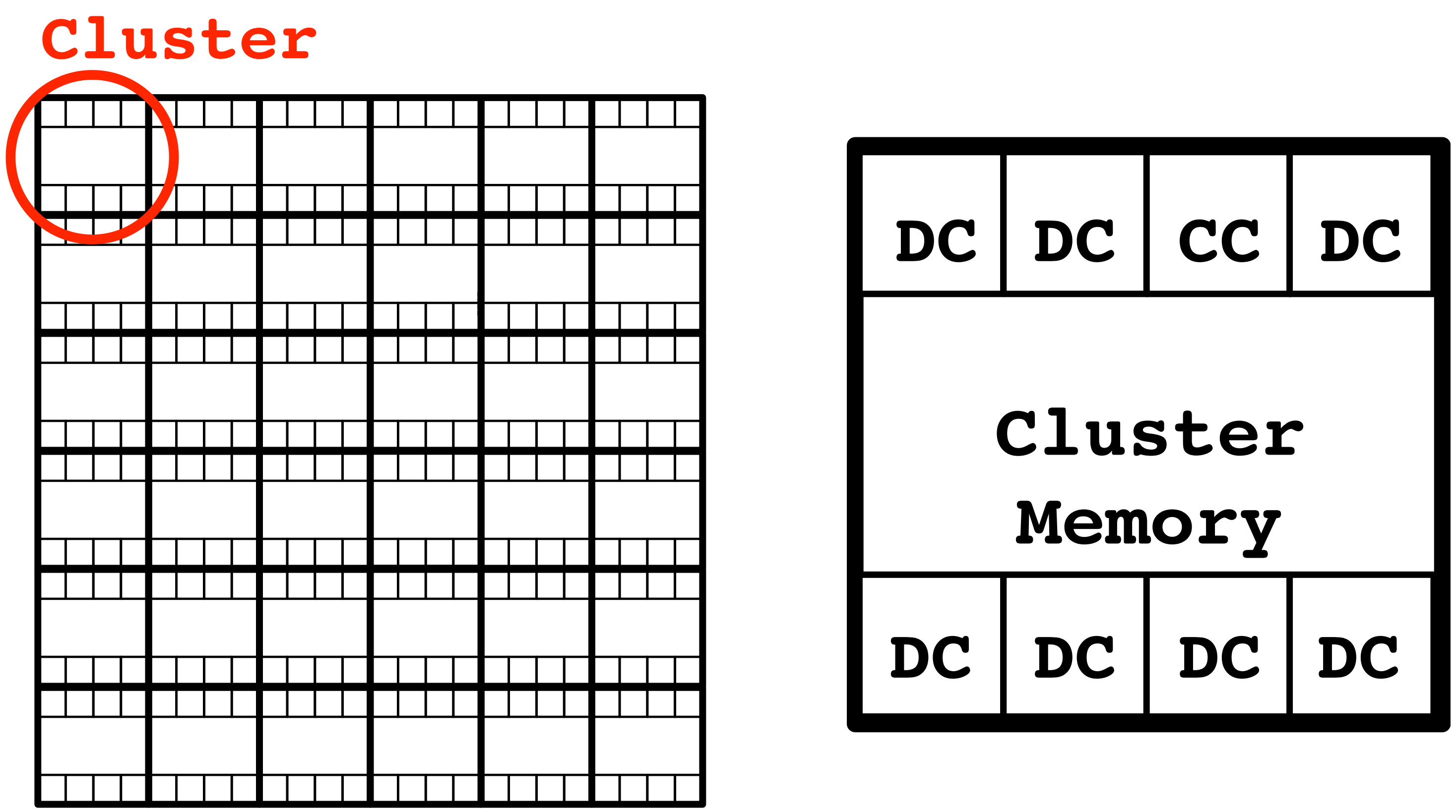
Evaluation Setup

- Simple, clustered hardware to exploit within die variation
  - Each cluster supports a different max f at chip-wide, single Vdd
  - All clusters assigned to a task cycle at the f of slowest cluster
- Simulated 20mm x 20mm 288 core chip at 11nm
  - 36 clusters, 8 cores per cluster
  - Core: Single issue in-order



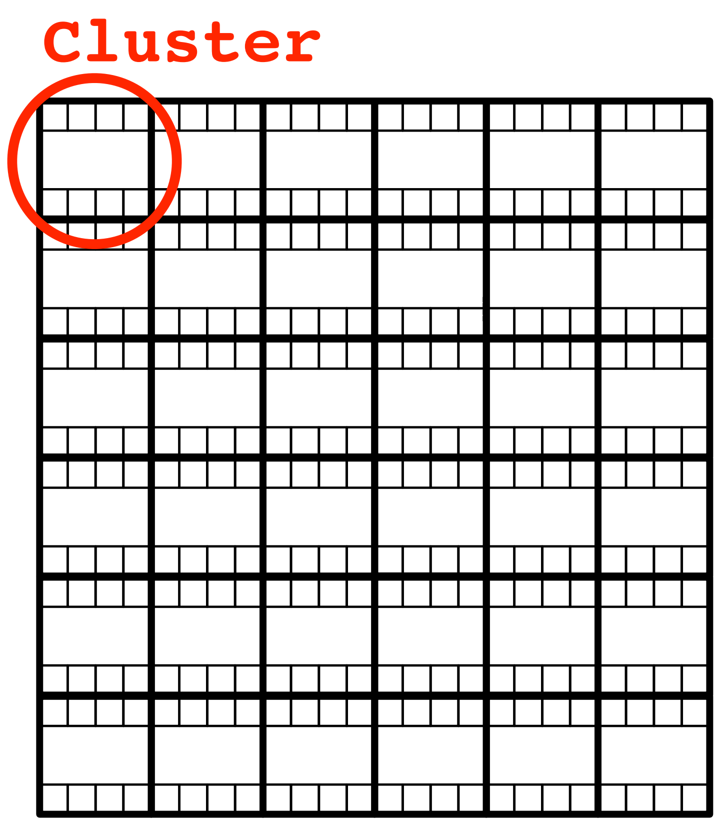
Evaluation Setup

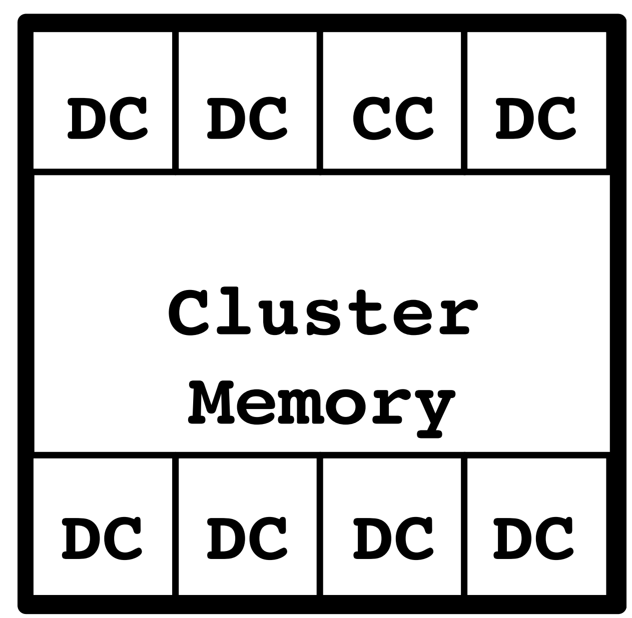
- Simple, clustered hardware to exploit within die variation
  - Each cluster supports a different max f at chip-wide, single Vdd
  - All clusters assigned to a task cycle at the f of slowest cluster
- Simulated 20mm x 20mm 288 core chip at 11nm
  - 36 clusters, 8 cores per cluster
  - Core: Single issue in-order
- VARIUS-NTV to extract per cluster min Vdd and max f



Evaluation Setup

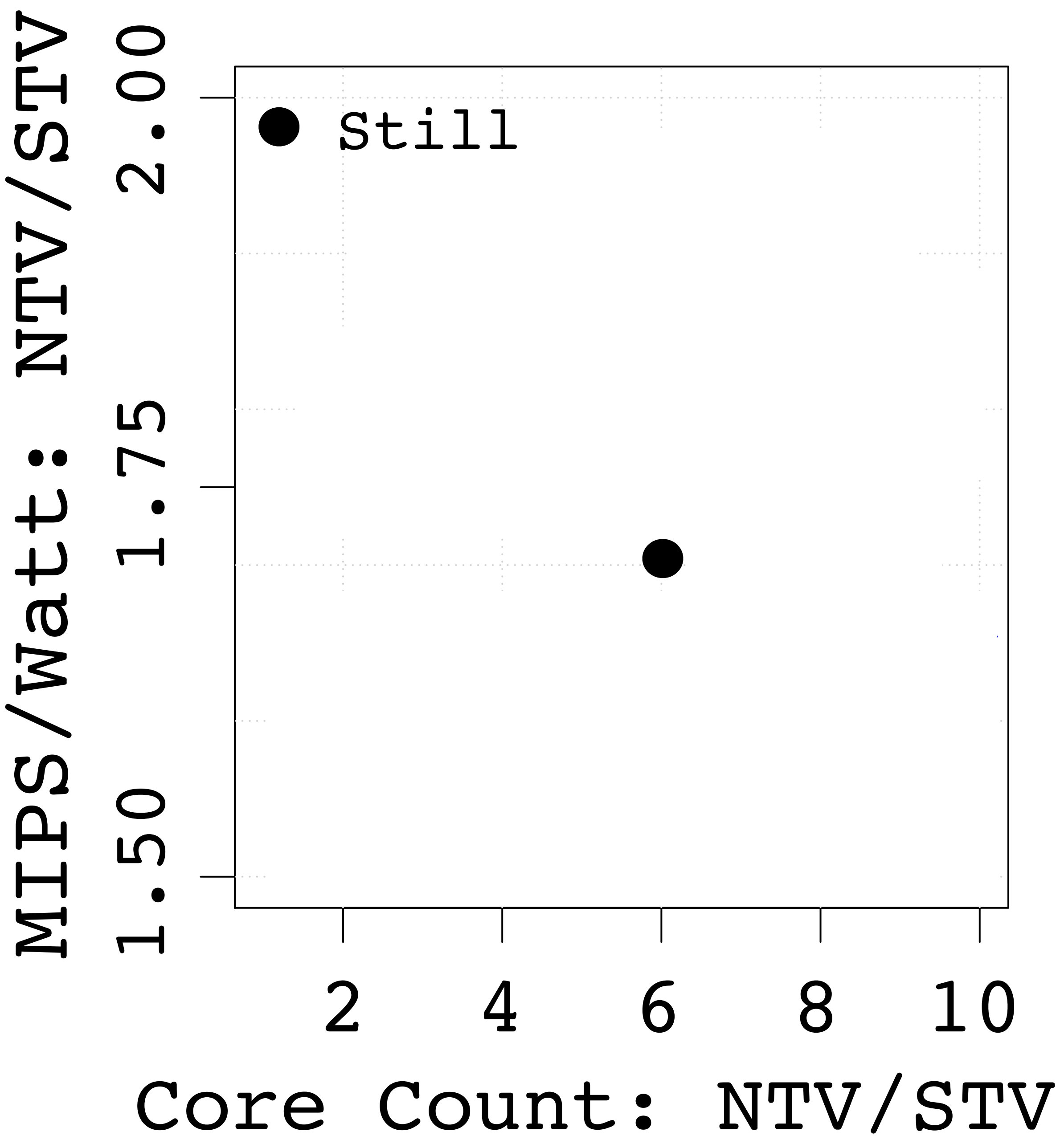
- Simple, clustered hardware to exploit within die variation
  - Each cluster supports a different max f at chip-wide, single Vdd
  - All clusters assigned to a task cycle at the f of slowest cluster
- Simulated 20mm x 20mm 288 core chip at 11nm
  - 36 clusters, 8 cores per cluster
  - Core: Single issue in-order
- VARIUS-NTV to extract per cluster min Vdd and max f
- RMS applications from PARSEC and Rodinia suites



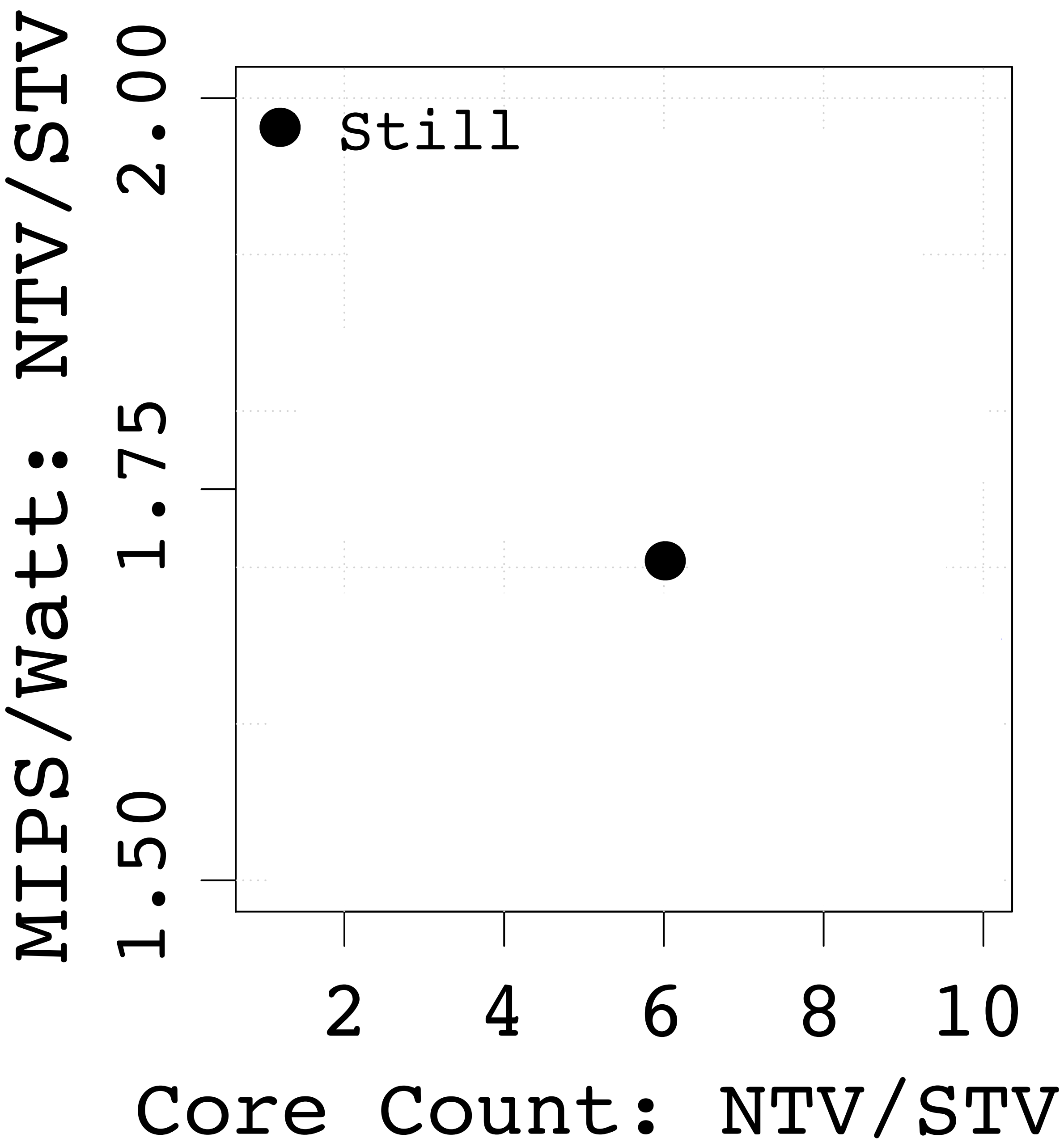


Iso-execution time front (canneal)

Iso-execution time front (canneal)

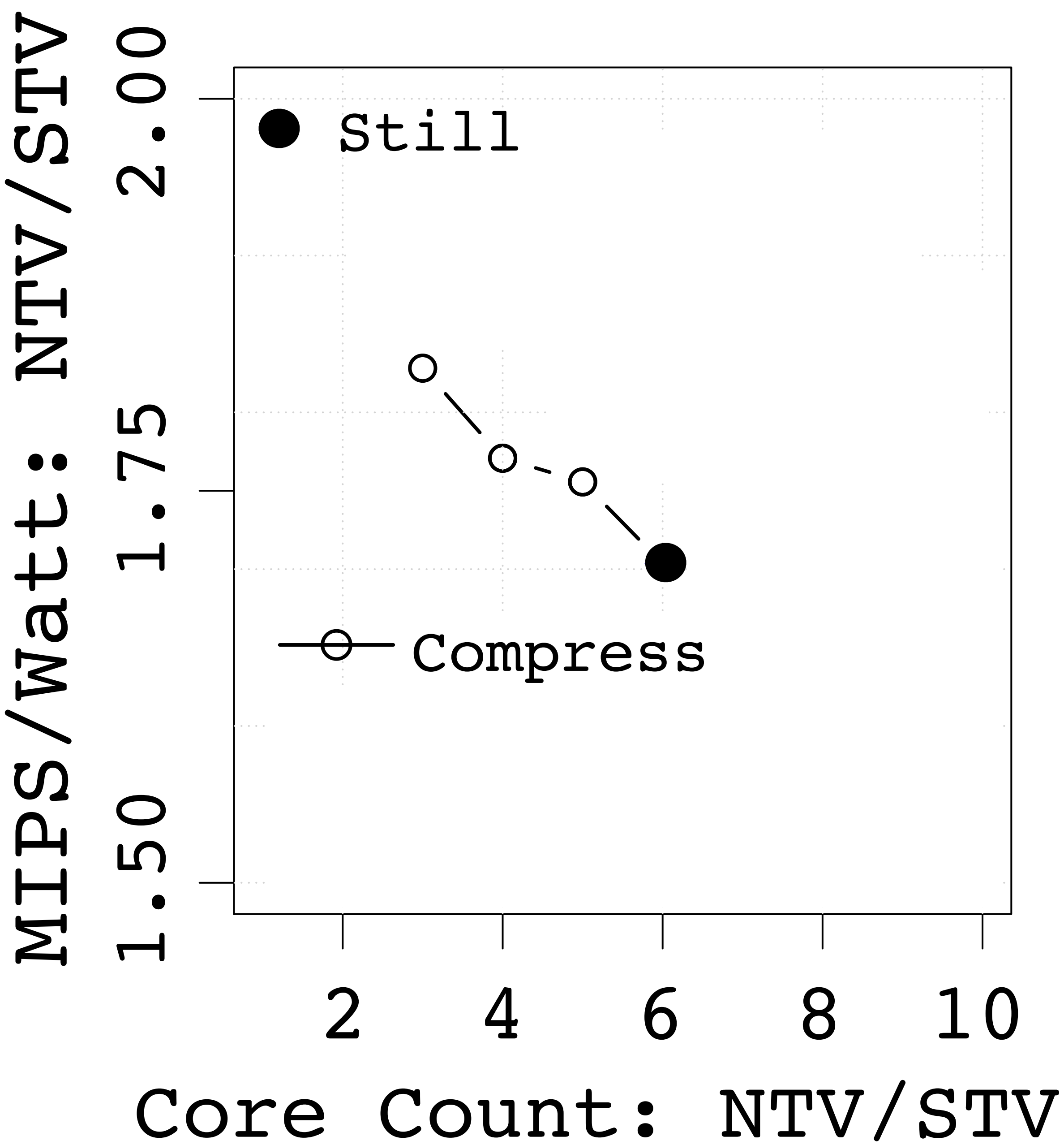


Iso-execution time front (canneal)



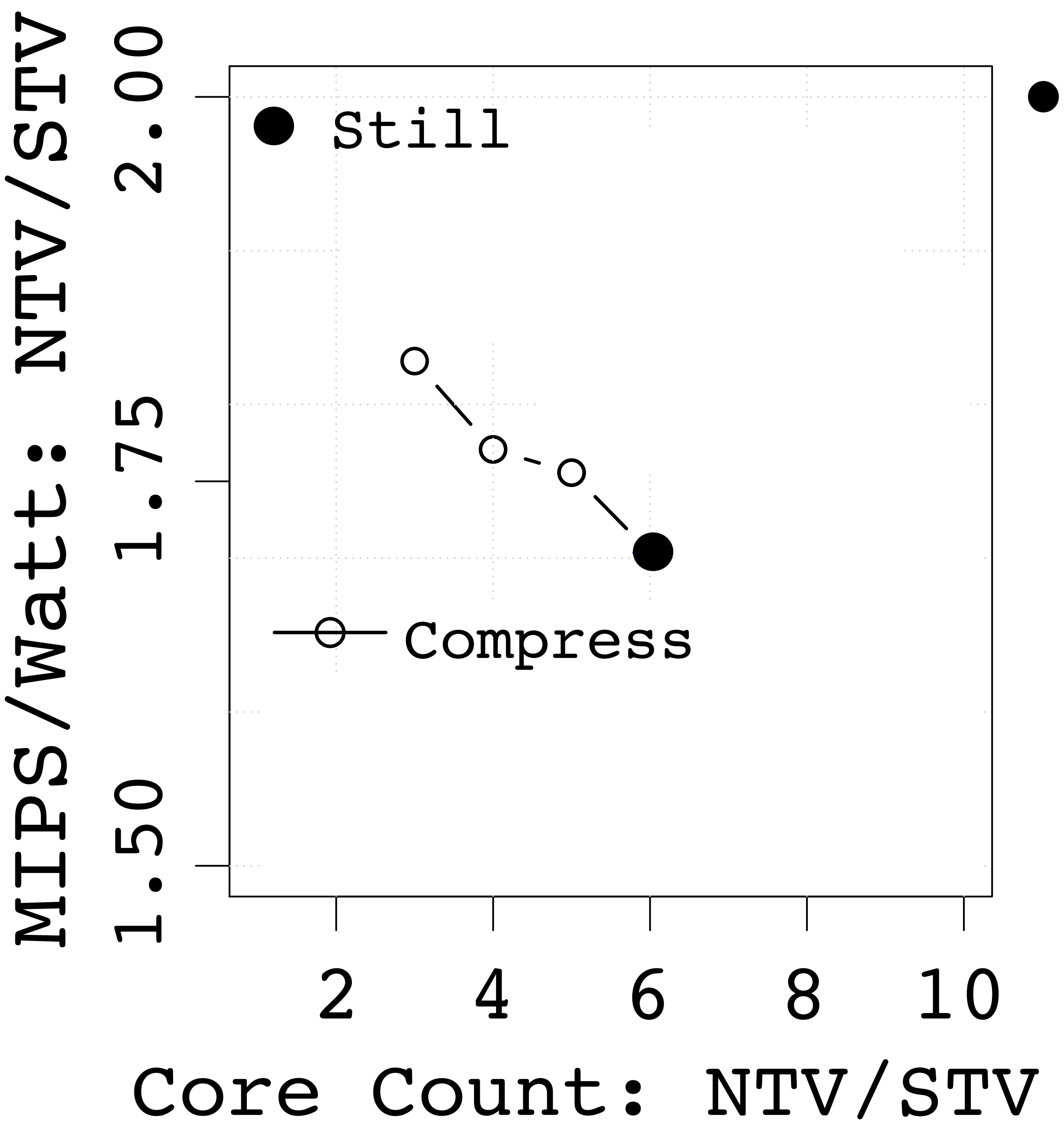
$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

Iso-execution time front (canneal)



$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

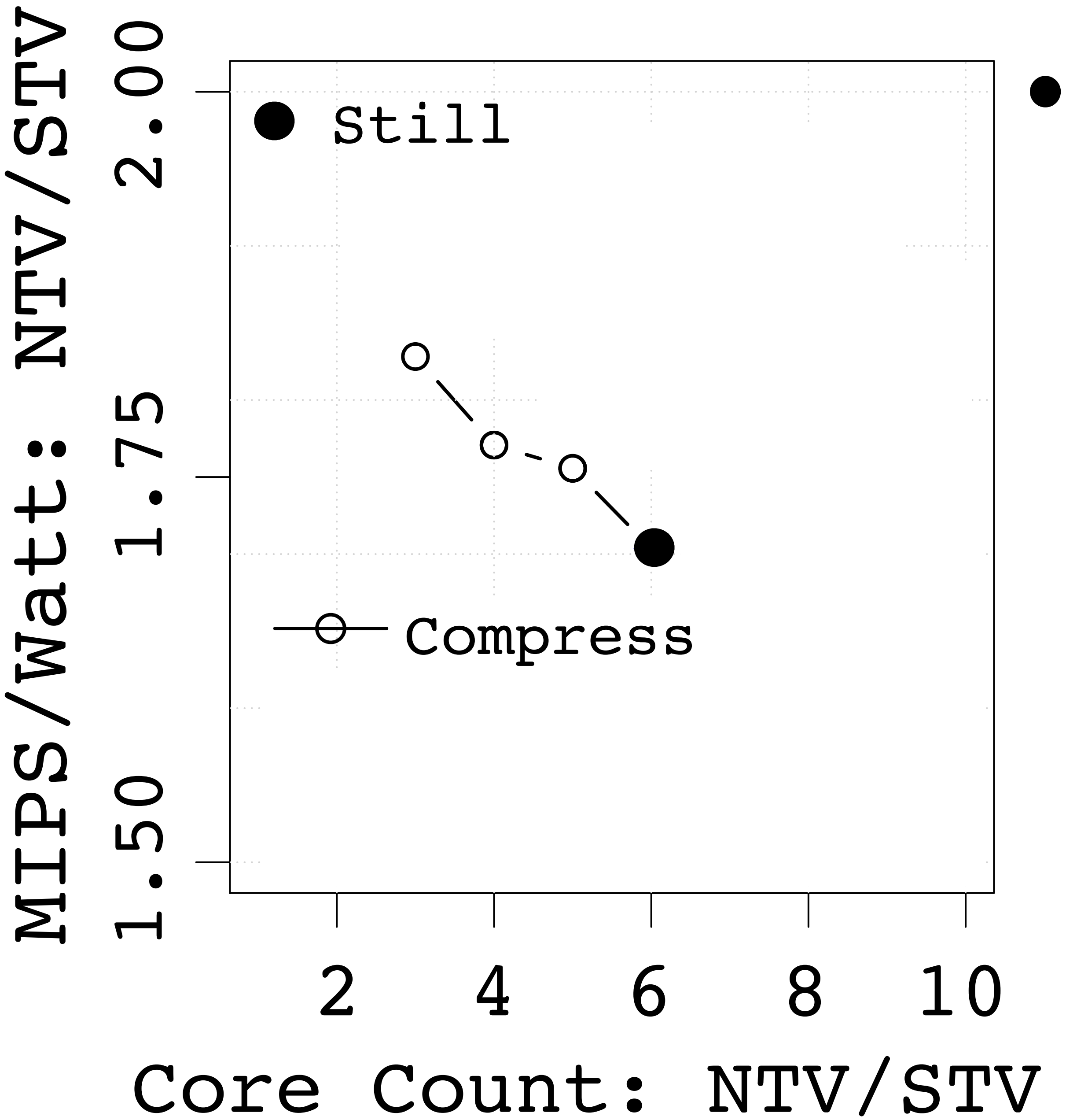
Iso-execution time front (canneal)



- As core count increases

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

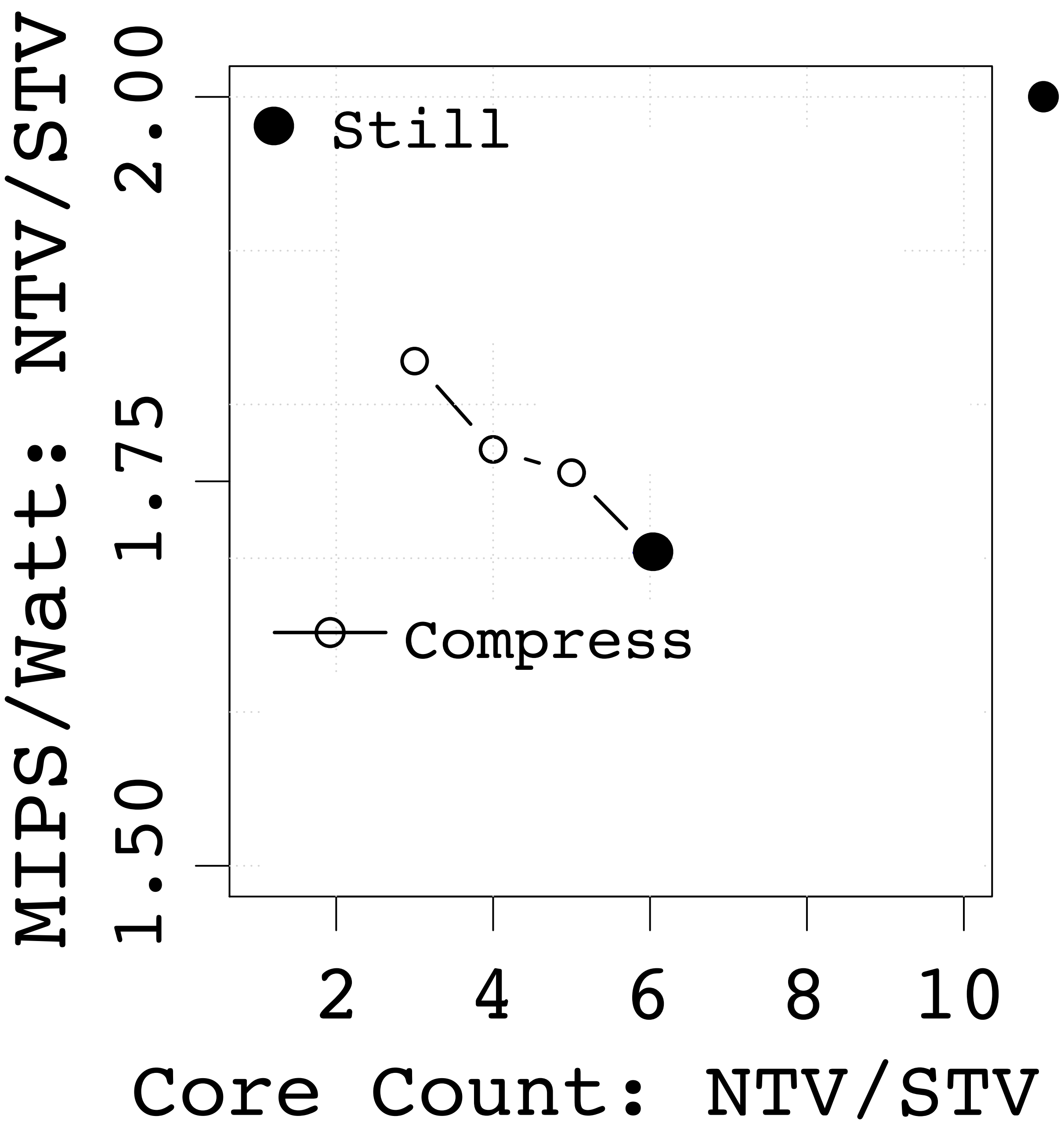
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

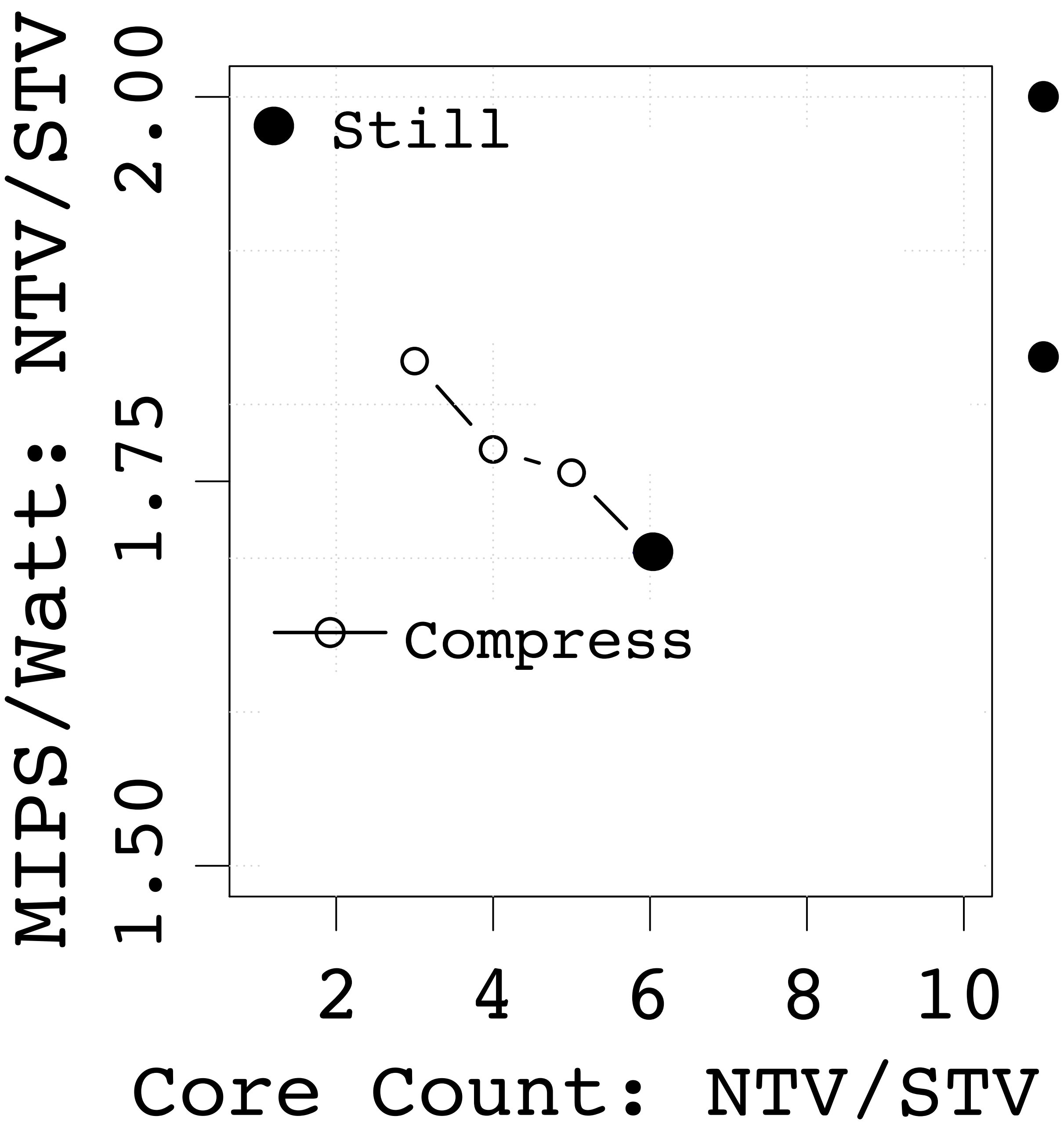
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

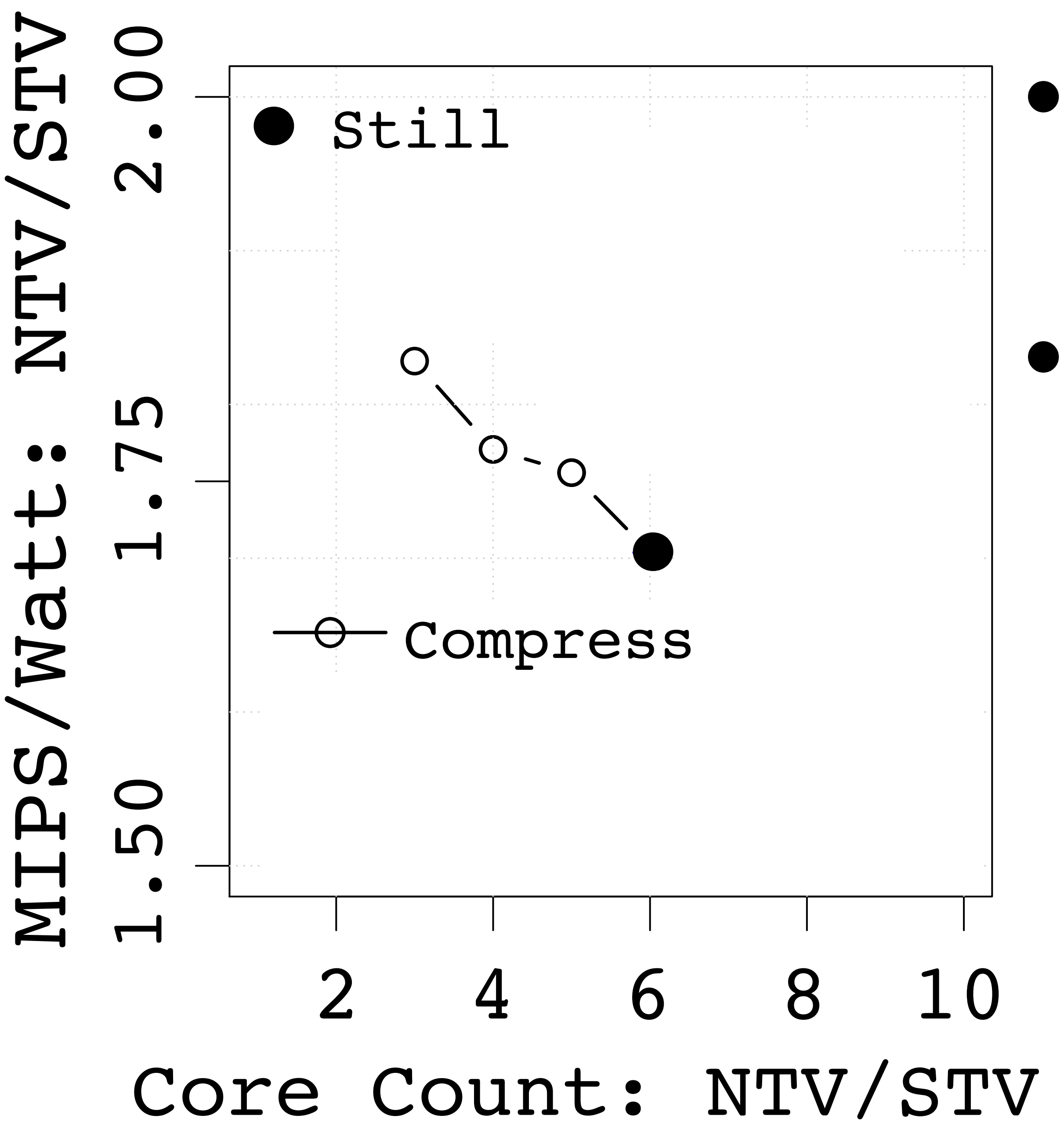
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Compress

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

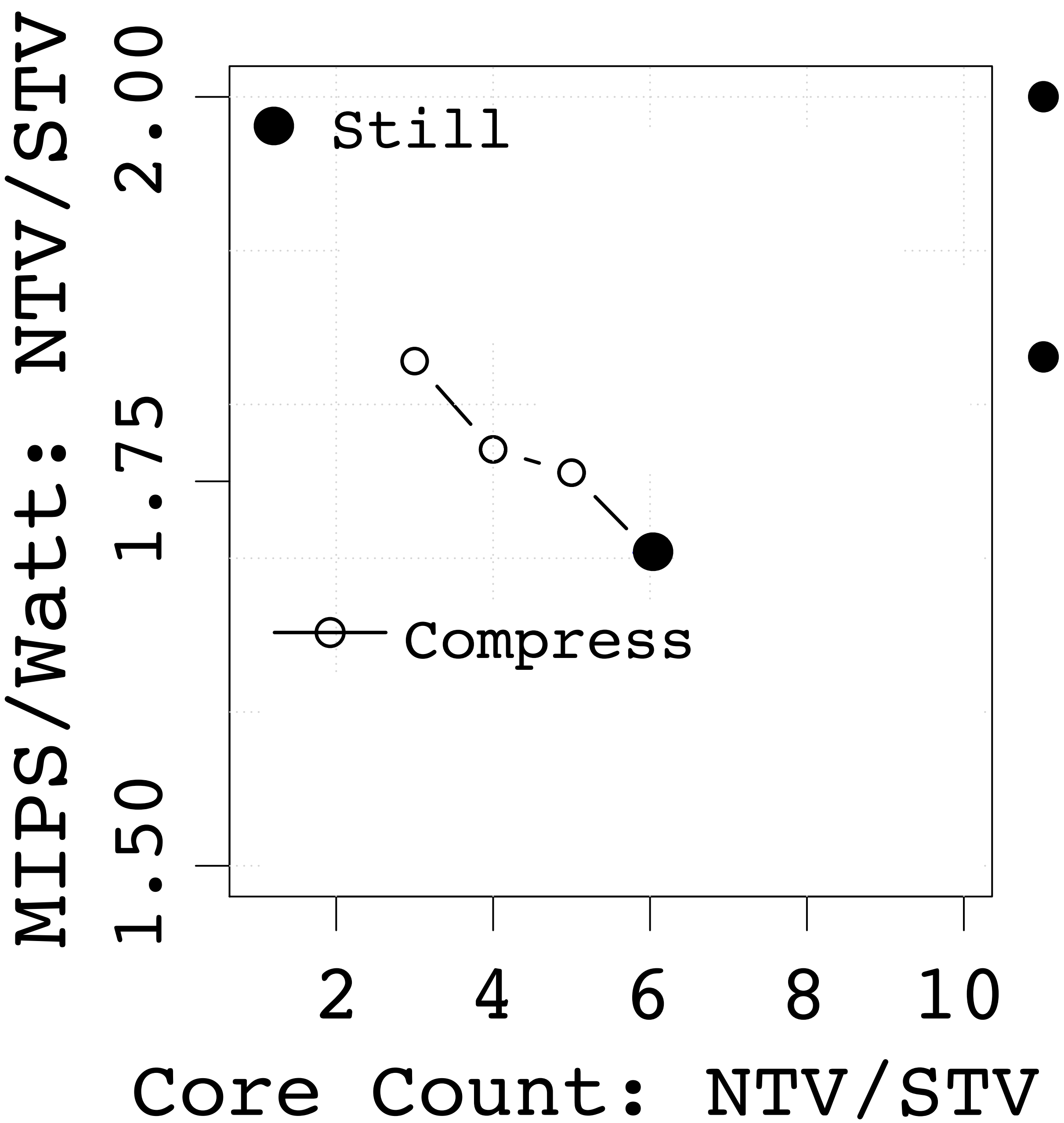
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Compress
  - Higher efficiency at lower core count

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

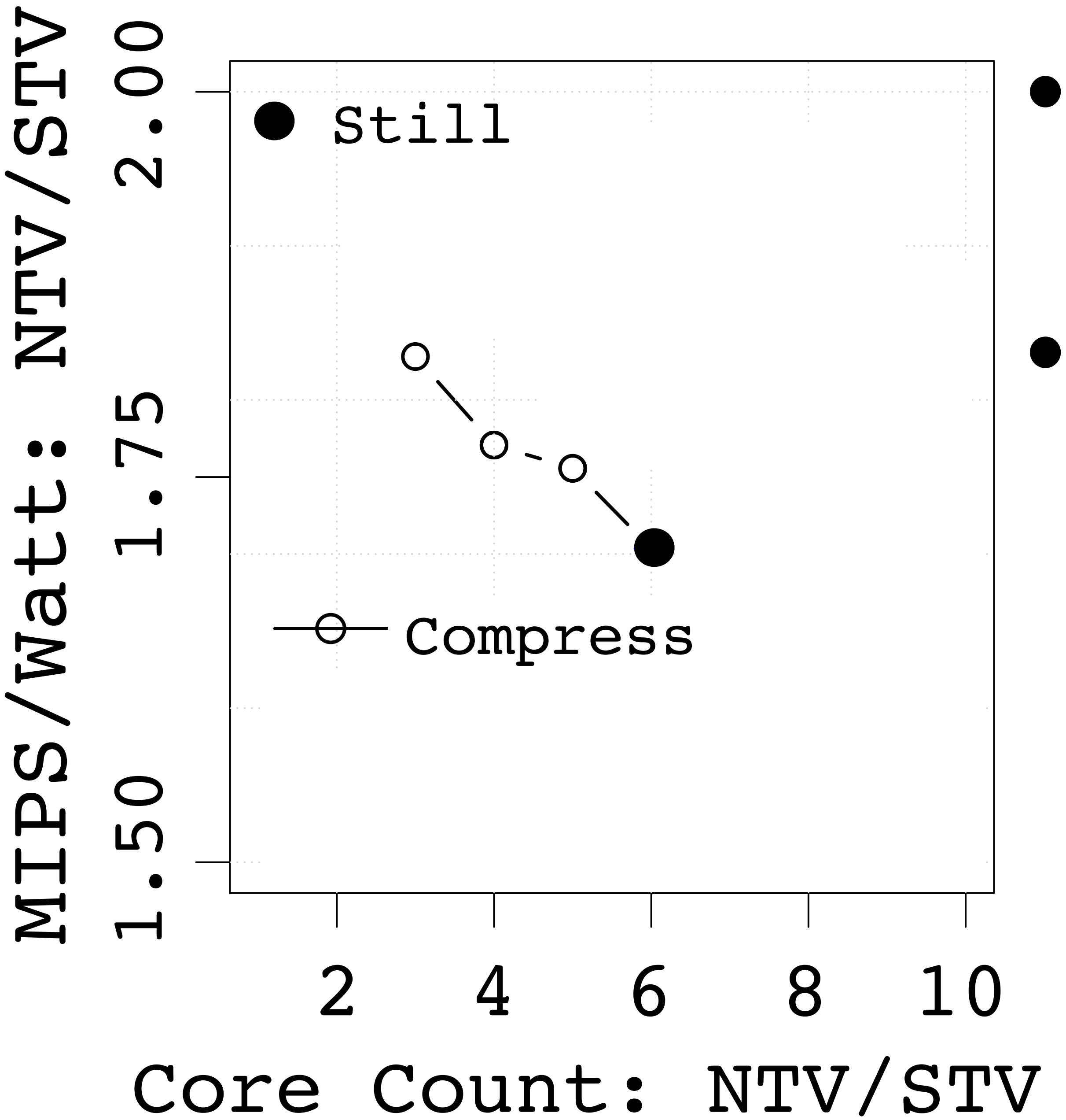
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Compress
  - Higher efficiency at lower core count
    - Higher f, lower power

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

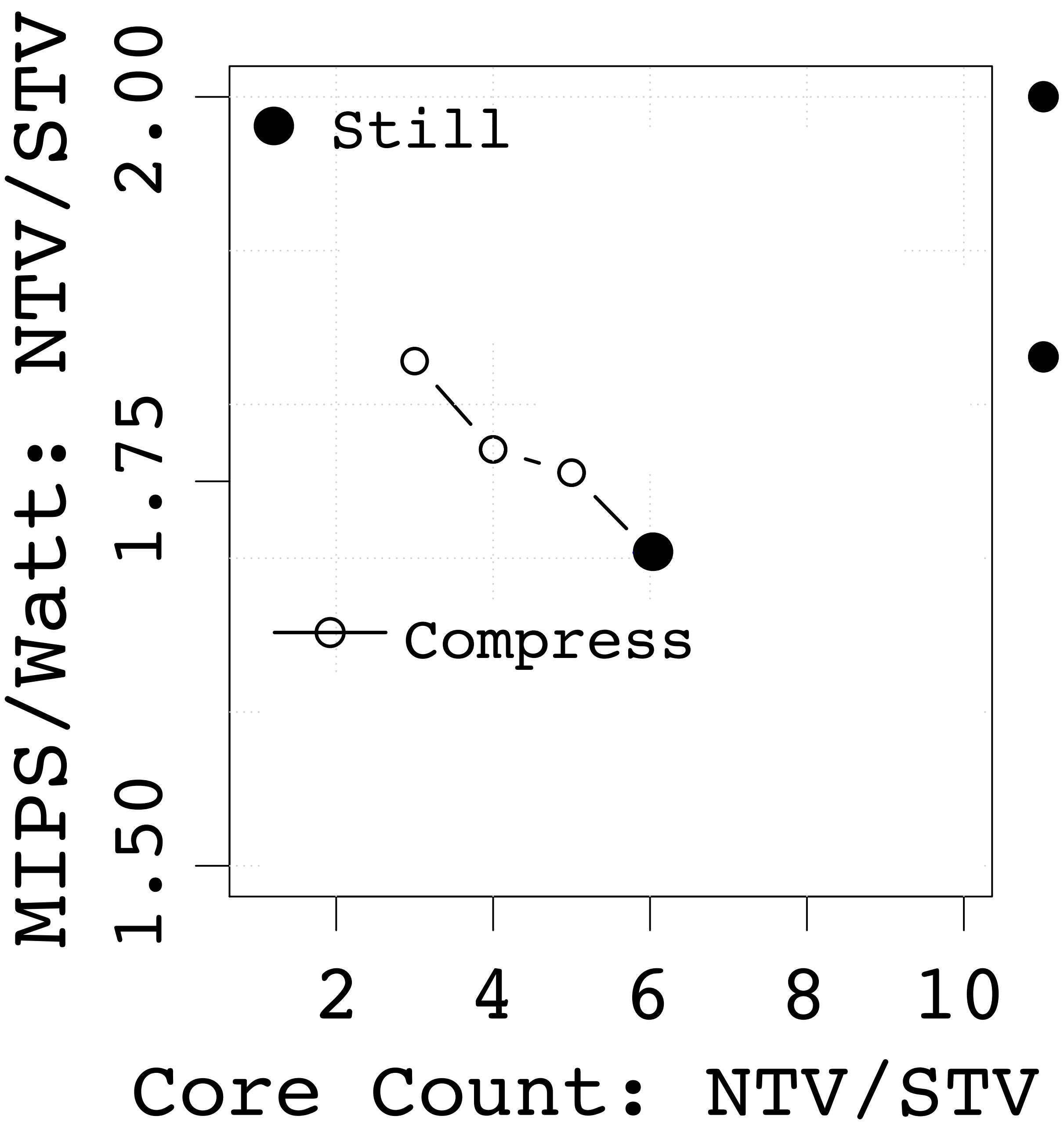
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Compress
  - Higher efficiency at lower core count
    - Higher f, lower power
  - Quality limited

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

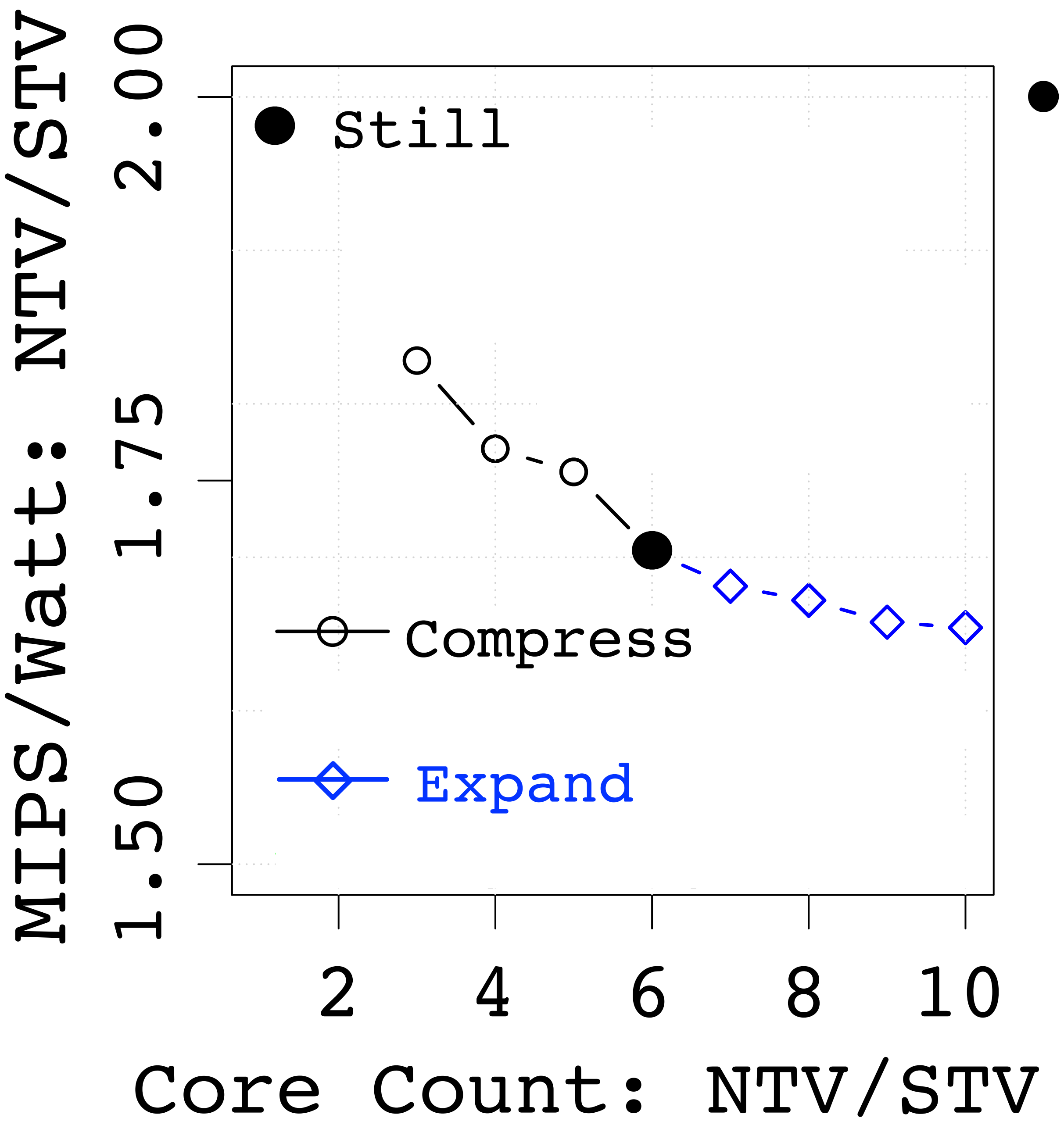
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Compress
  - Higher efficiency at lower core count
    - Higher f, lower power
  - Quality limited

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

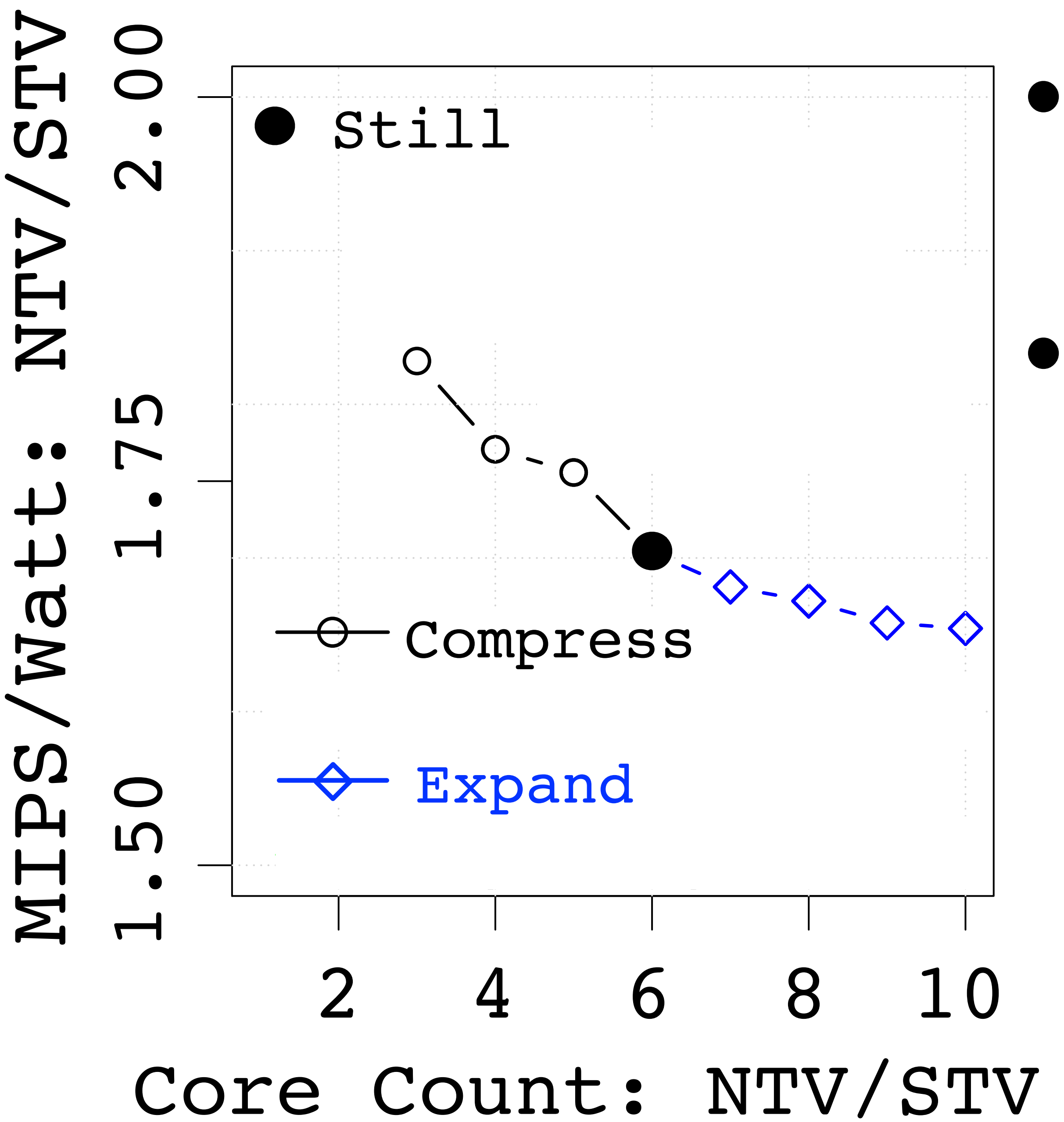
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{\text{f} \times \text{Core Count}}$$

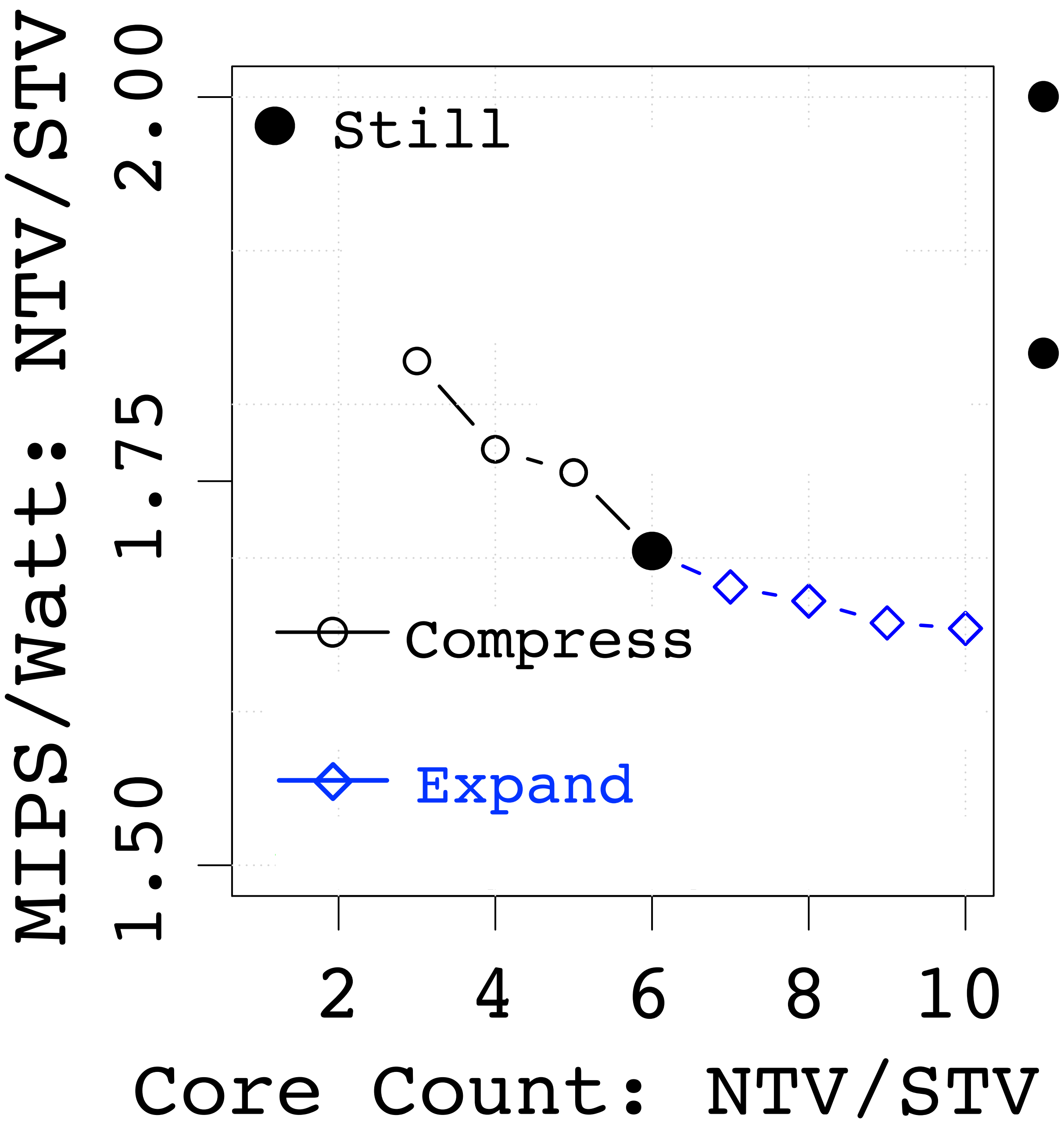
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Expand

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

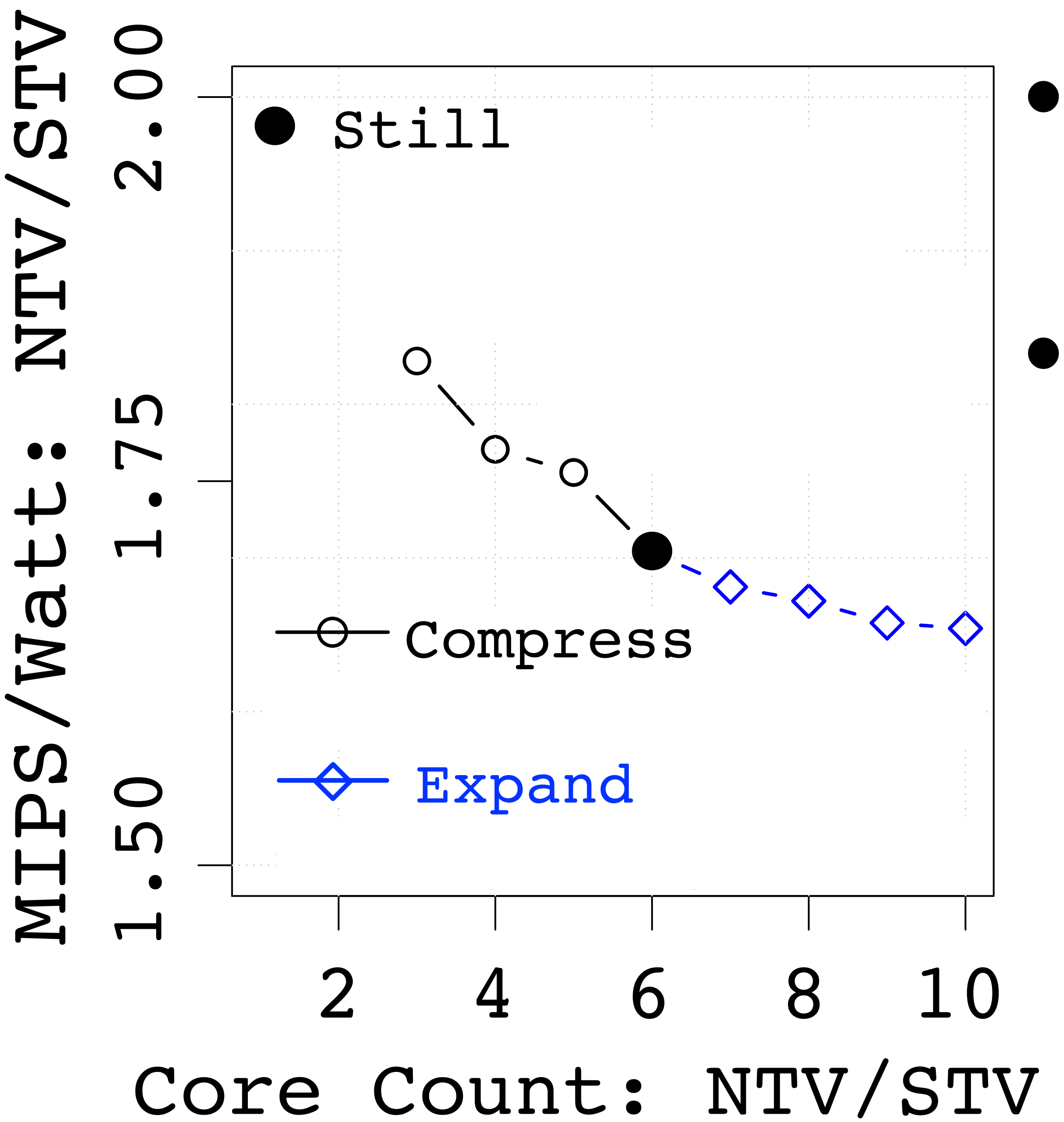
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Expand
  - Higher efficiency at lower core count

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{\text{f} \times \text{Core Count}}$$

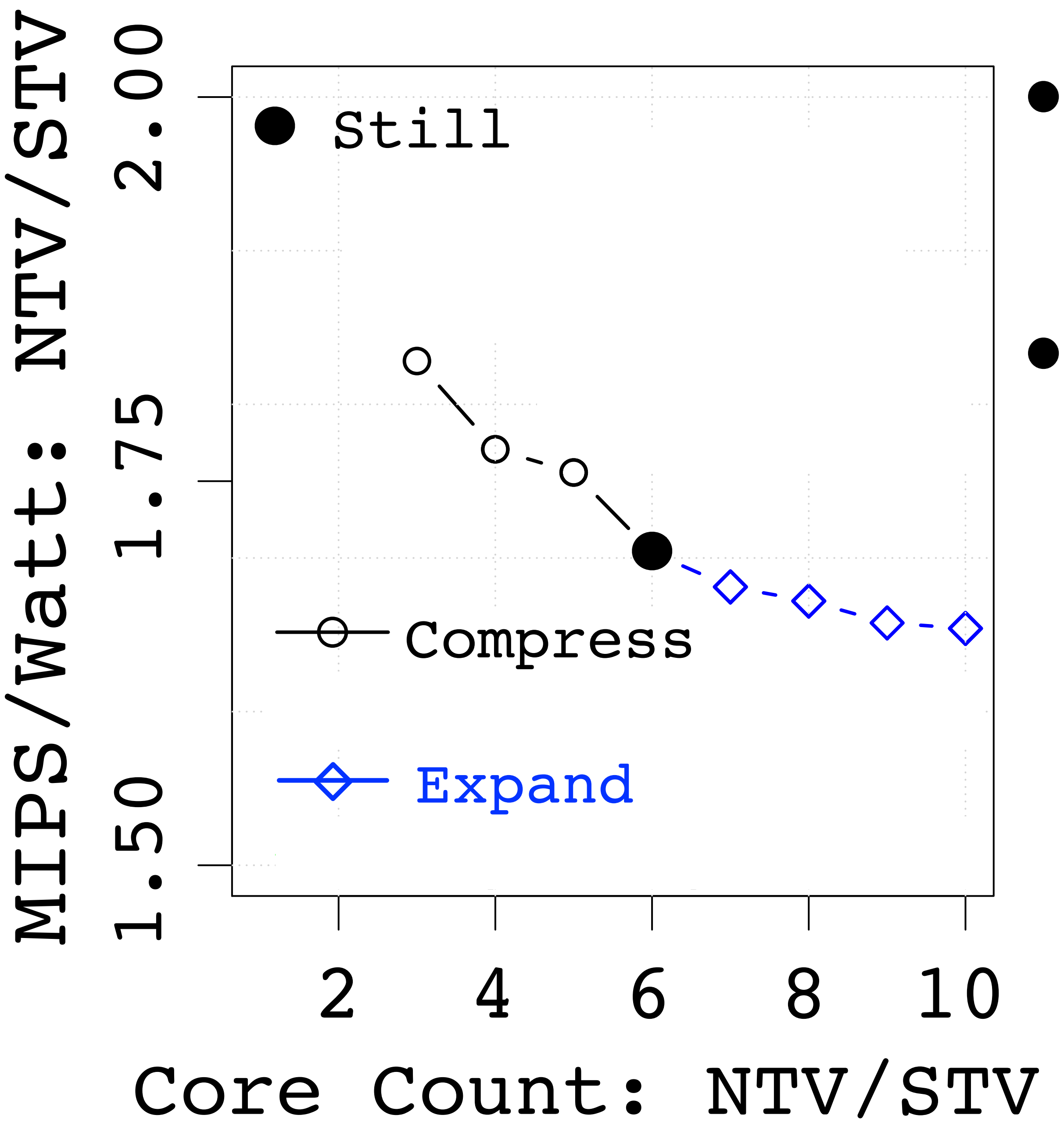
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Expand
  - Higher efficiency at lower core count
  - A feasible core count may not exist

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

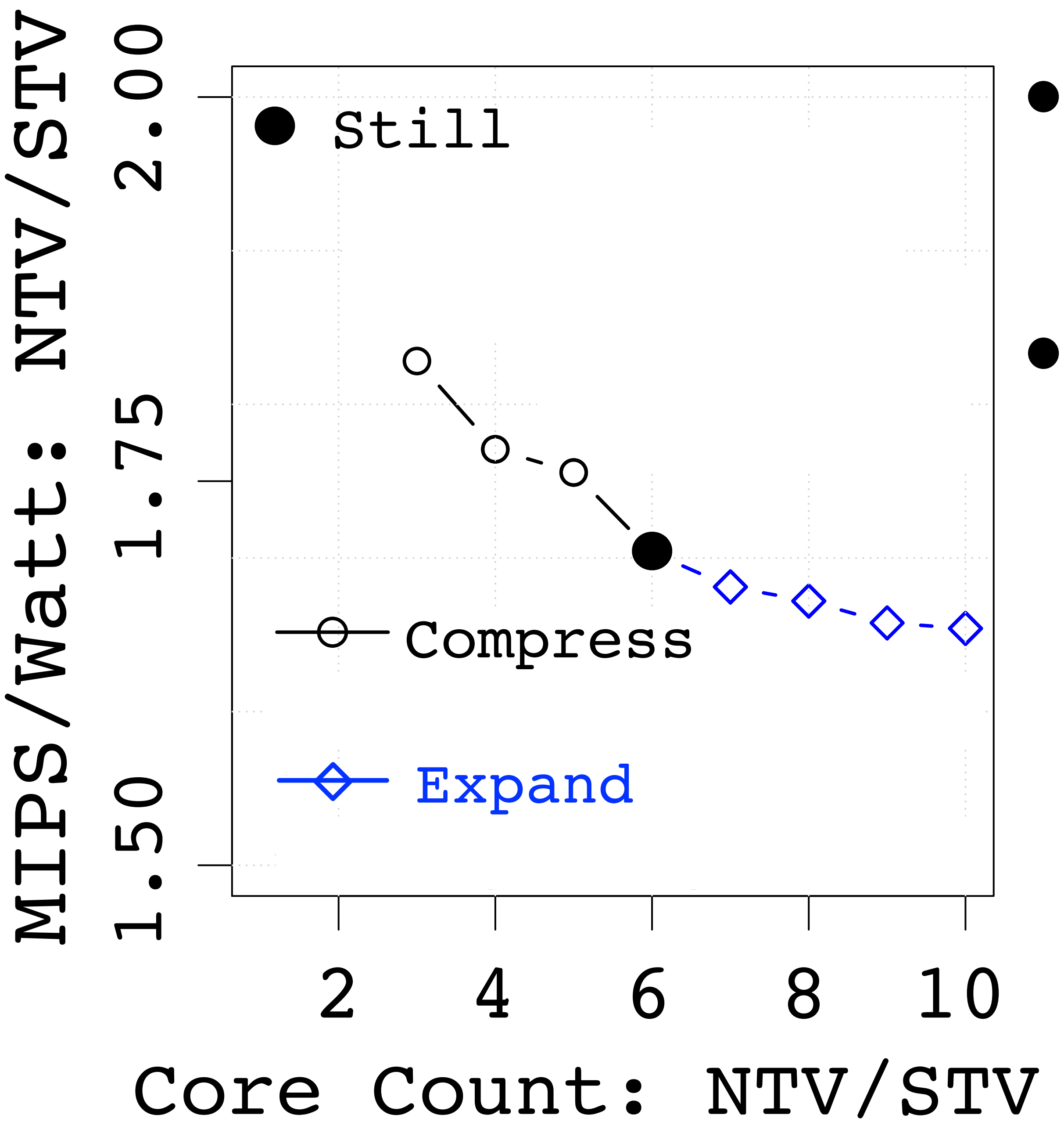
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Expand
  - Higher efficiency at lower core count
  - A feasible core count may not exist
  - Core count (area), or power limited

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

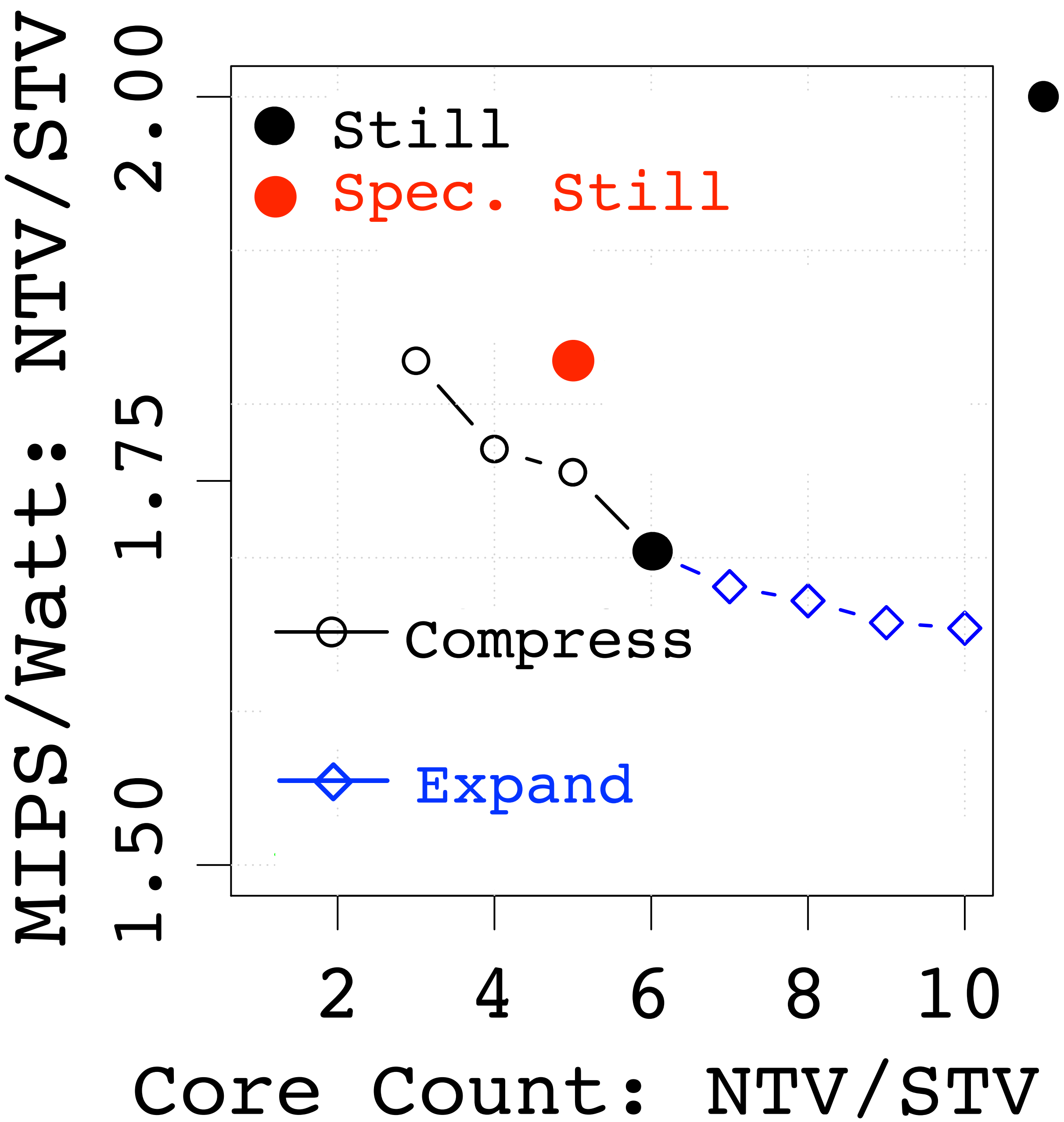
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Expand
  - Higher efficiency at lower core count
  - A feasible core count may not exist
  - Core count (area), or power limited

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

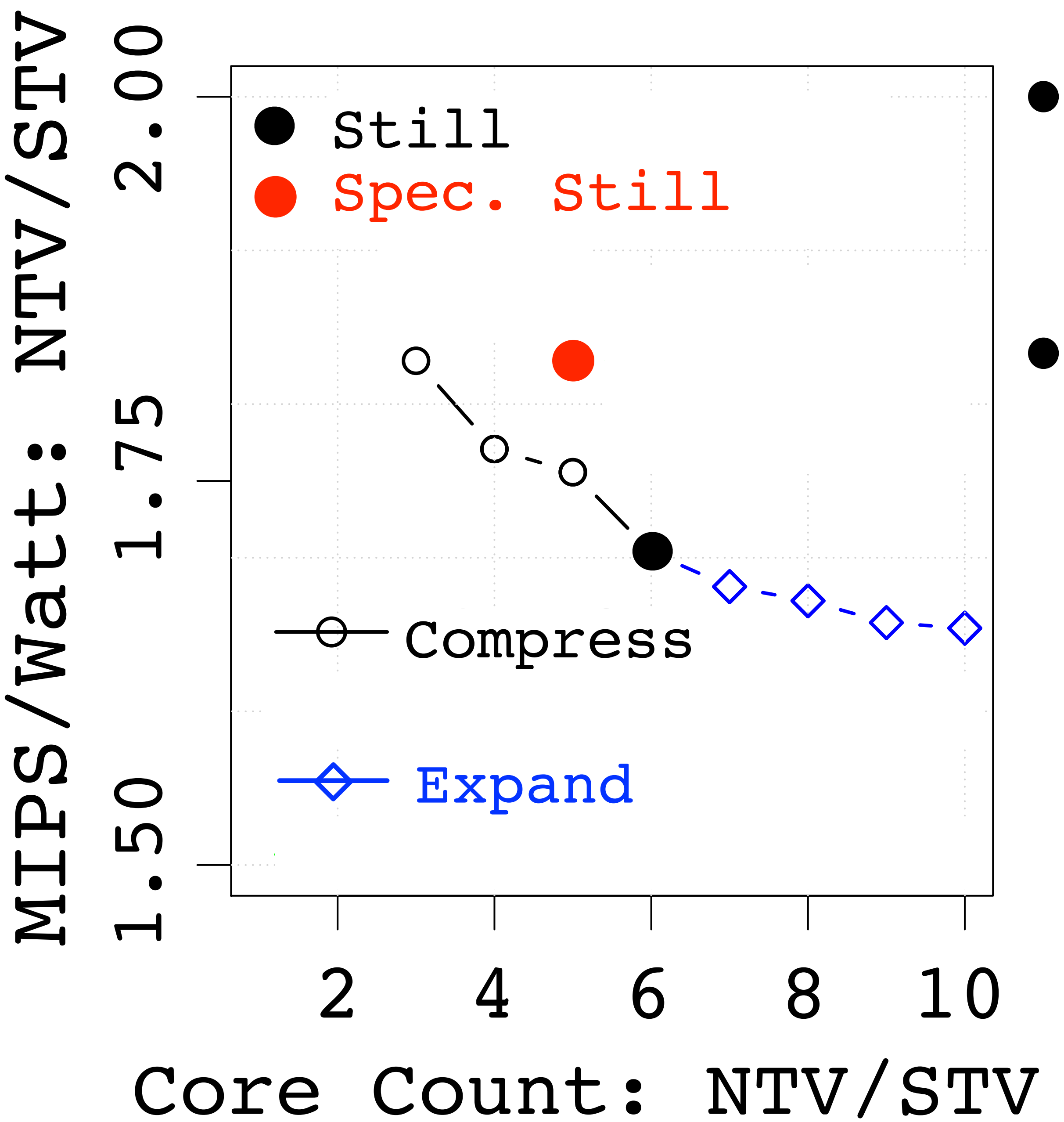
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

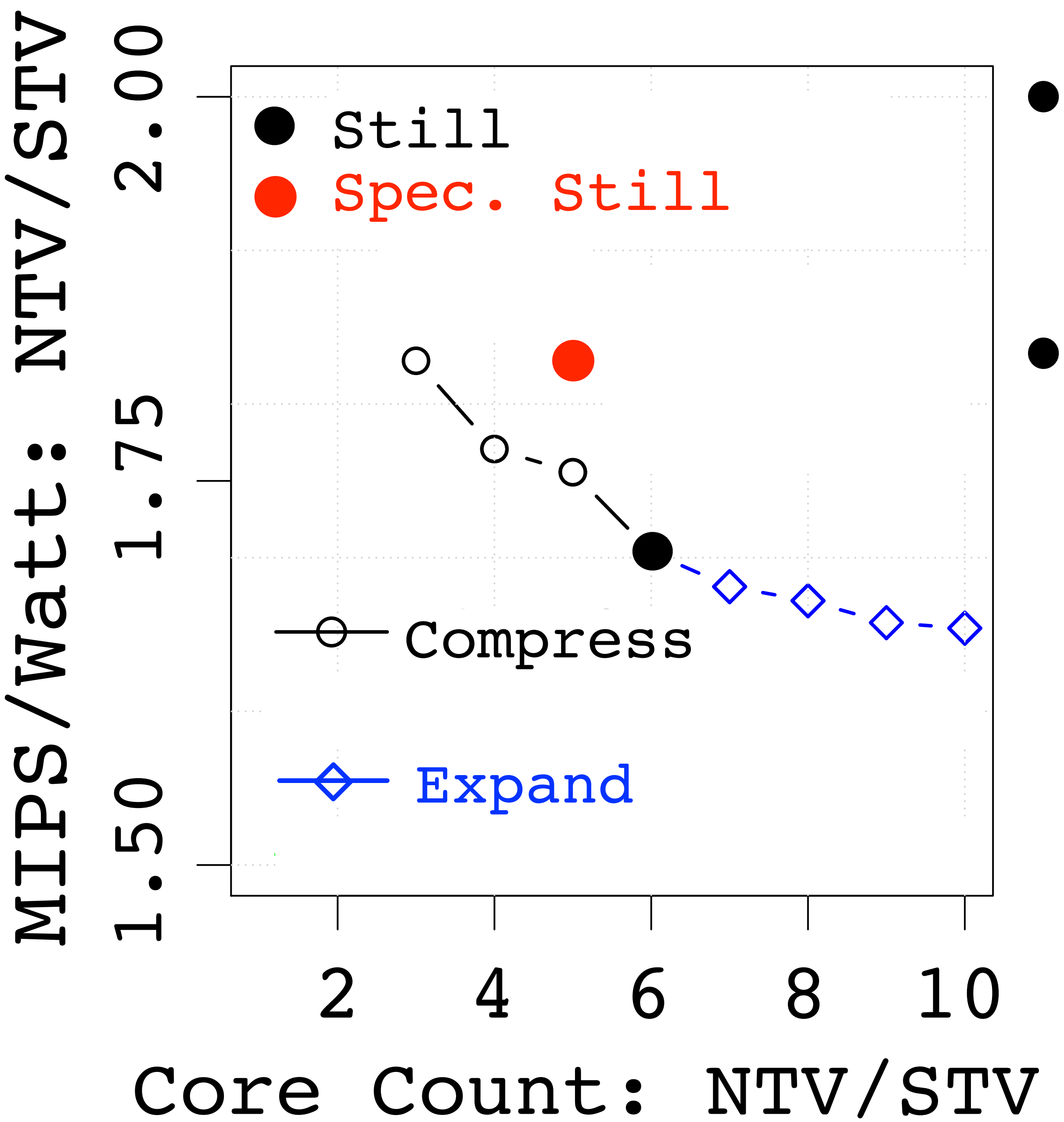
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Speculative

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

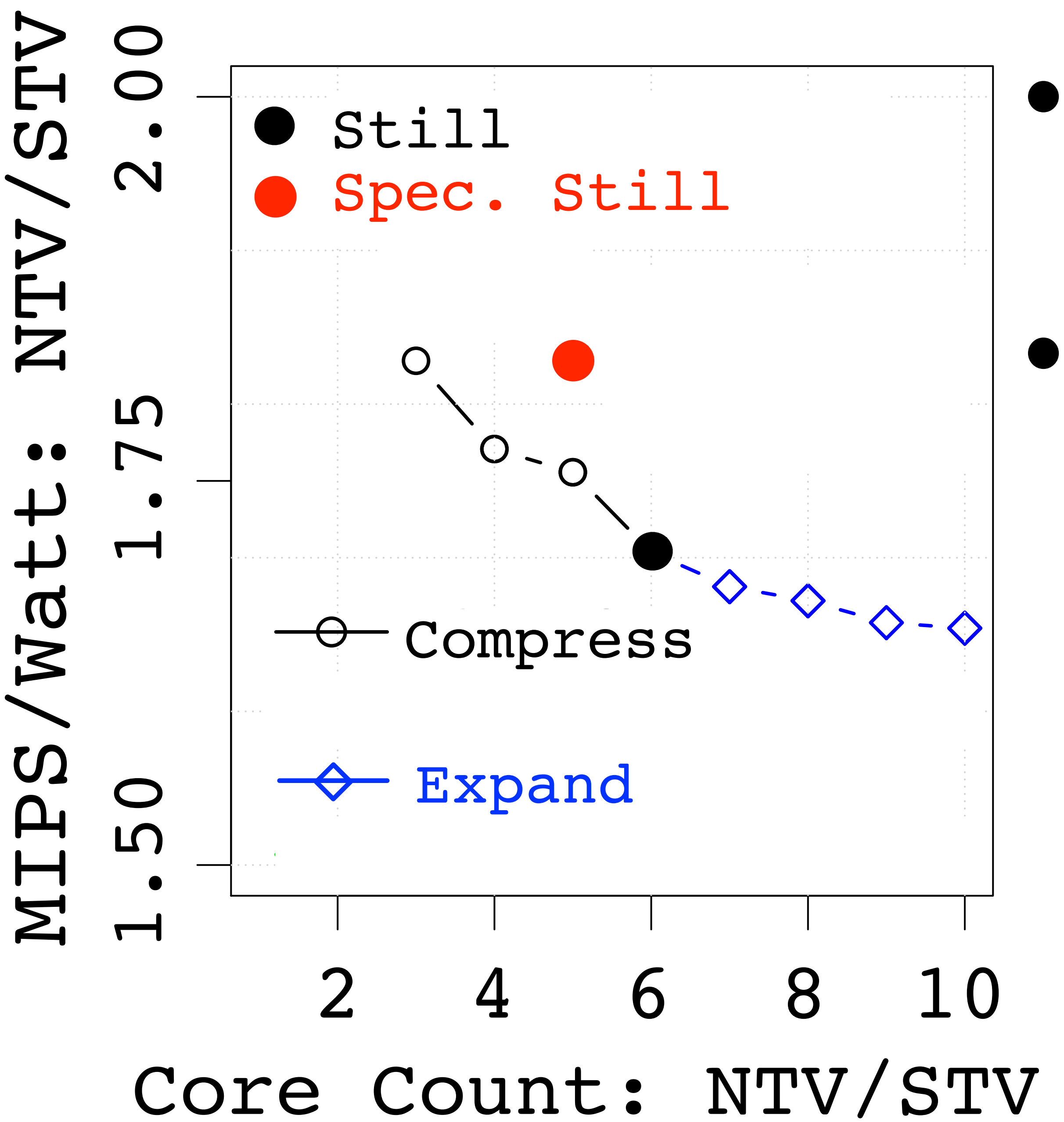
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Speculative
  - Higher f facilitates lower core count

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

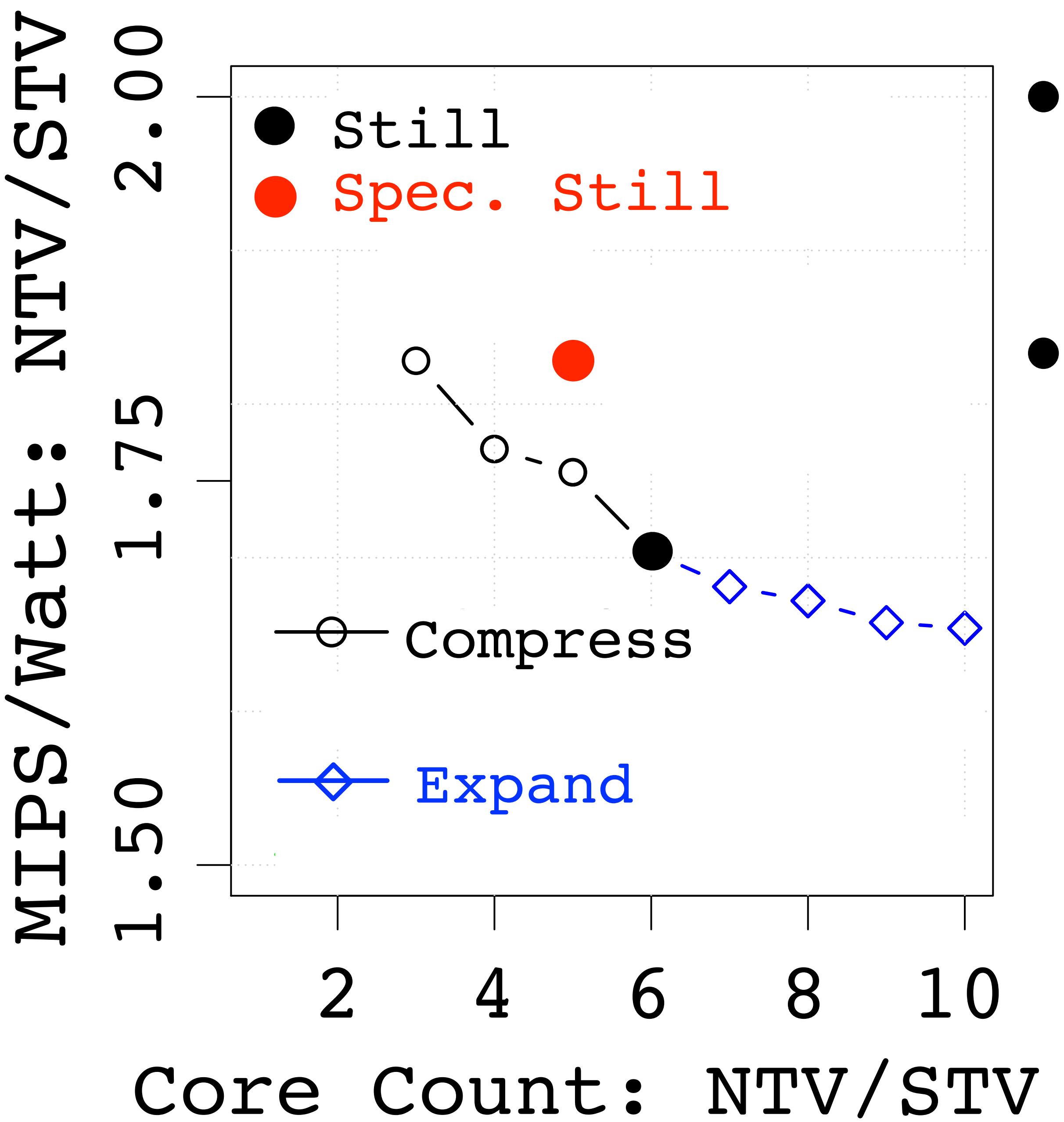
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Speculative
  - Higher f facilitates lower core count
  - Lower core count → lower power

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

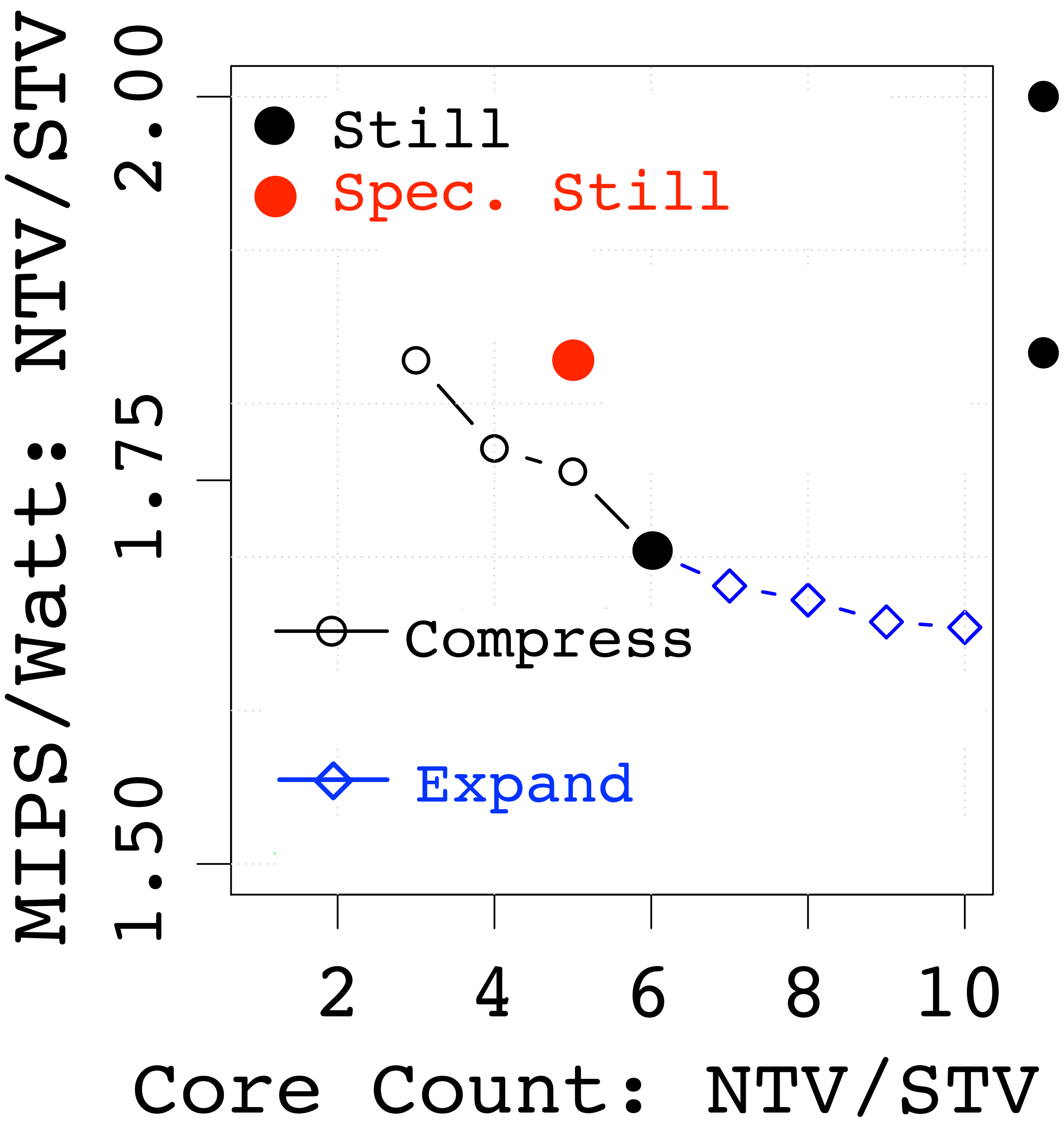
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Speculative
  - Higher f facilitates lower core count
  - Lower core count → lower power
    - Energy efficiency increases

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

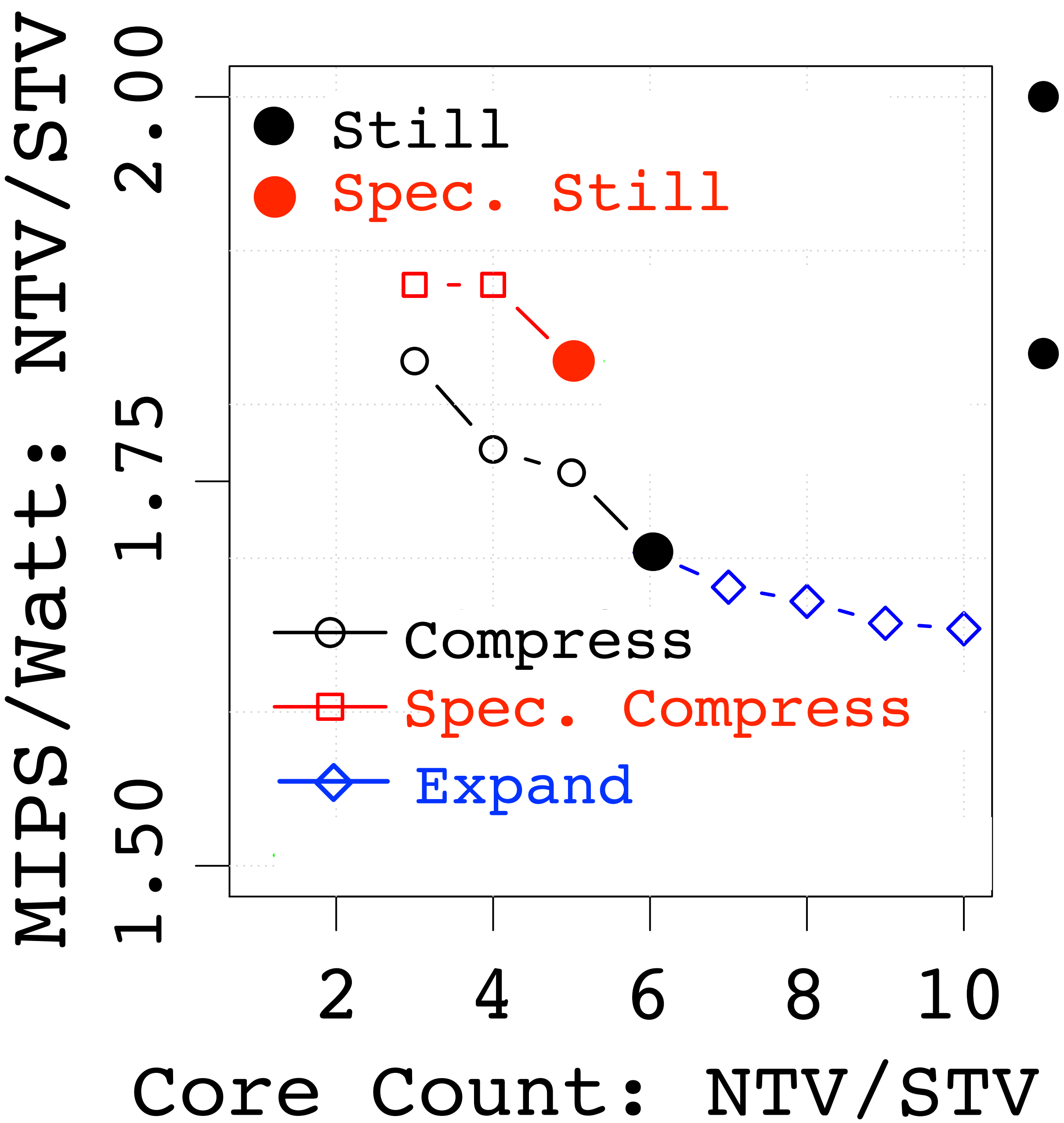
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Speculative
  - Higher f facilitates lower core count
  - Lower core count → lower power
    - Energy efficiency increases

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

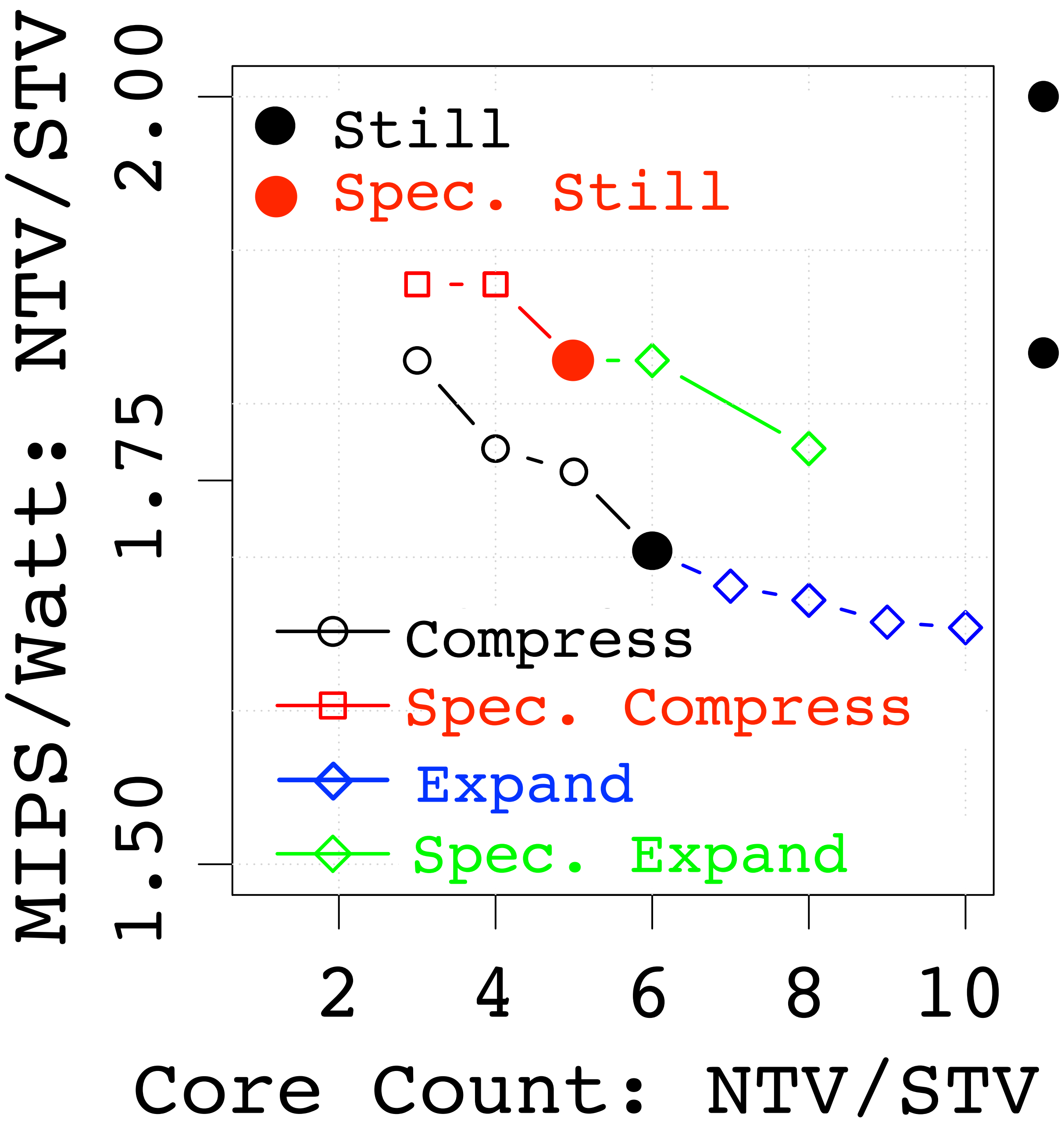
Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Speculative
  - Higher f facilitates lower core count
  - Lower core count → lower power
    - Energy efficiency increases

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

Iso-execution time front (canneal)



- As core count increases
  - More likely to engage slower cores
  - f decreases
- Speculative
  - Higher f facilitates lower core count
  - Lower core count → lower power
    - Energy efficiency increases

$$\text{Execution Time} \propto \frac{\text{Problem Size}}{f \times \text{Core Count}}$$

Conclusion

Conclusion

- Devises problem size as the main knob to overcome NTC barriers

Conclusion

- Devises problem size as the main knob to overcome NTC barriers
  - Problem size dictates

Conclusion

- Devises problem size as the main knob to overcome NTC barriers
  - Problem size dictates
    - the number of cores engaged in computation

Conclusion

- Devises problem size as the main knob to overcome NTC barriers
  - Problem size dictates
    - the number of cores engaged in computation
    - variation induced output quality degradation

Conclusion

- Devises problem size as the main knob to overcome NTC barriers
  - Problem size dictates
    - the number of cores engaged in computation
    - variation induced output quality degradation
- Decouples data & control to confine errors where they can be tolerated

Conclusion

- Devises problem size as the main knob to overcome NTC barriers
  - Problem size dictates
    - the number of cores engaged in computation
    - variation induced output quality degradation
- Decouples data & control to confine errors where they can be tolerated
- Can achieve STV execution time

Conclusion

- Devises problem size as the main knob to overcome NTC barriers
  - Problem size dictates
    - the number of cores engaged in computation
    - variation induced output quality degradation
- Decouples data & control to confine errors where they can be tolerated
- Can achieve STV execution time
  - while operating 1.61-1.87x more energy-efficiently

Accordion

Toward Soft Near-threshold Voltage Computing

Ulya R. Karpuzcu, Ismail Akturk

Nam Sung Kim

UNIVERSITY OF MINNESOTA

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

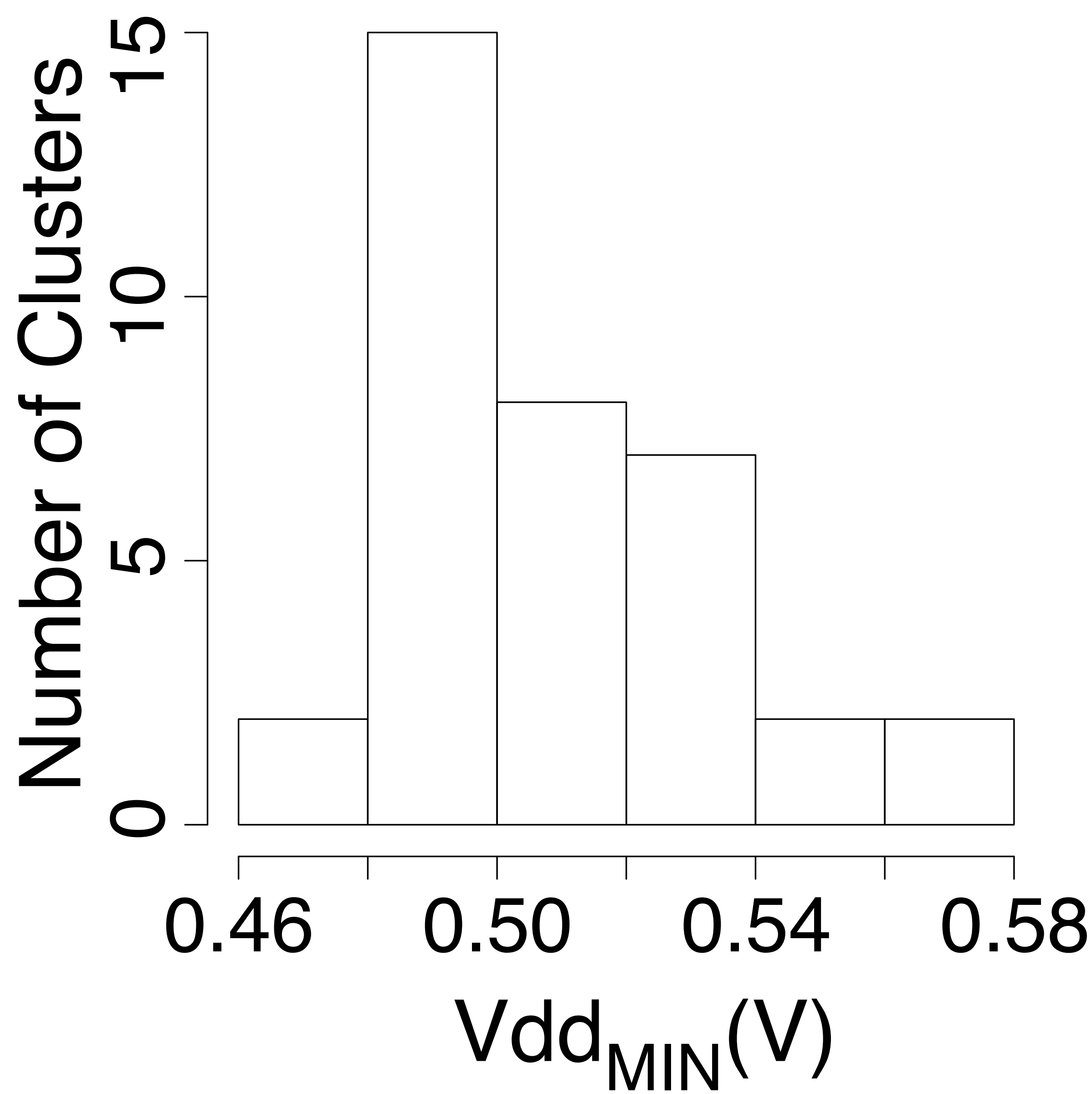
Evaluation Set-up

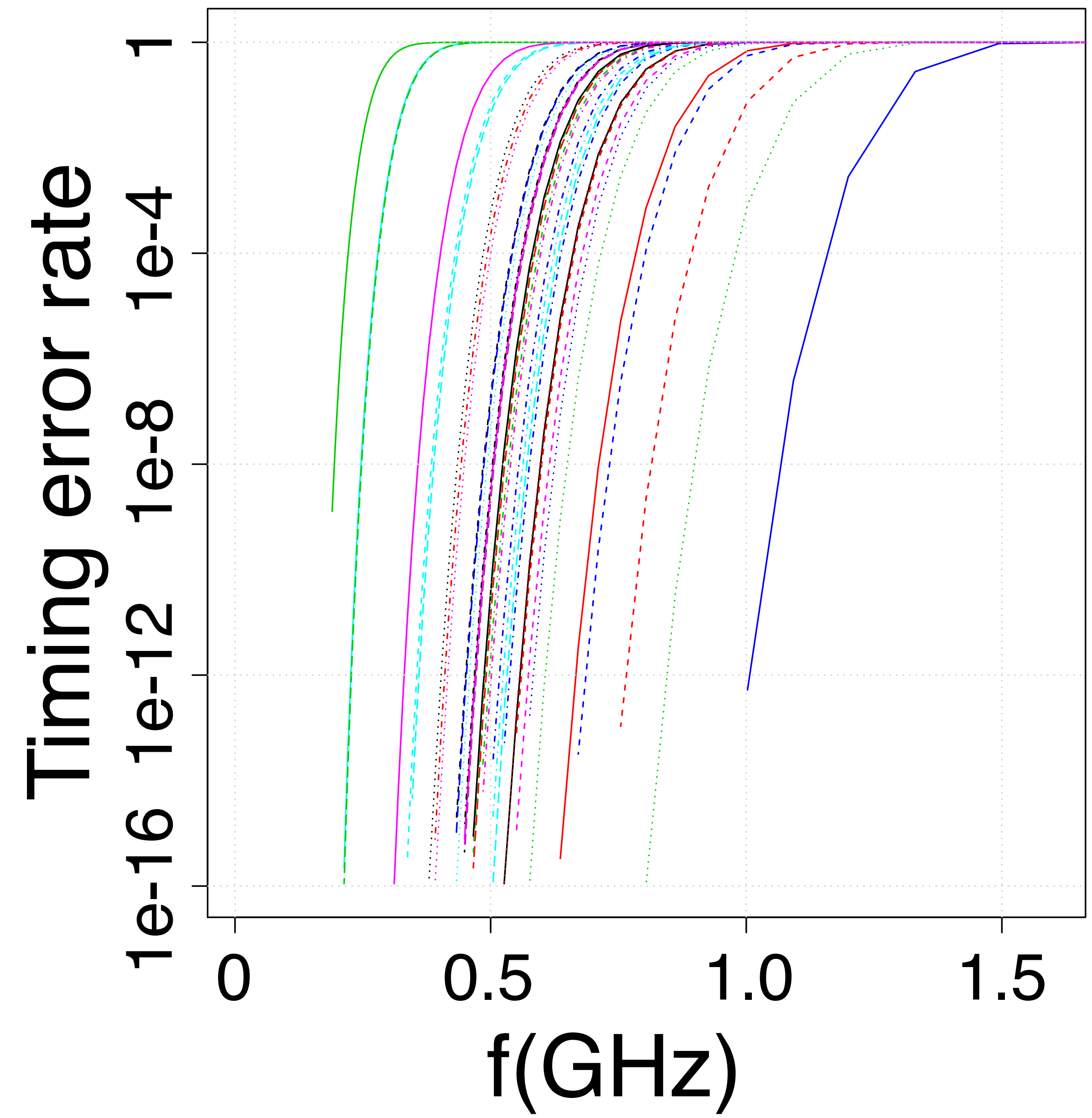
Benchmark	Application domain	Quality metric	Accordion input	Dependence on Accordion input: Problem Size	Dependence on Accordion input: Quality
canneal (PARSEC)	Optimization	Relative routing cost	Swaps per temperature step Number of temperature steps	linear linear	linear linear
ferret (PARSEC)	Similarity search	Based on number of common images	Size factor	complex	complex
bodytrack (PARSEC)	Computer vision	SSD based	Number of annealing layers	complex	complex
x264 (PARSEC)	Multimedia	SSIM based	Quantizer	complex	linear
hotspot (Rodinia)	Physics simulation	SSD based	Number of iterations	linear	linear
srad (Rodinia)	Image processing	PSNR based	Number of iterations	linear	linear

Evaluation Set-up

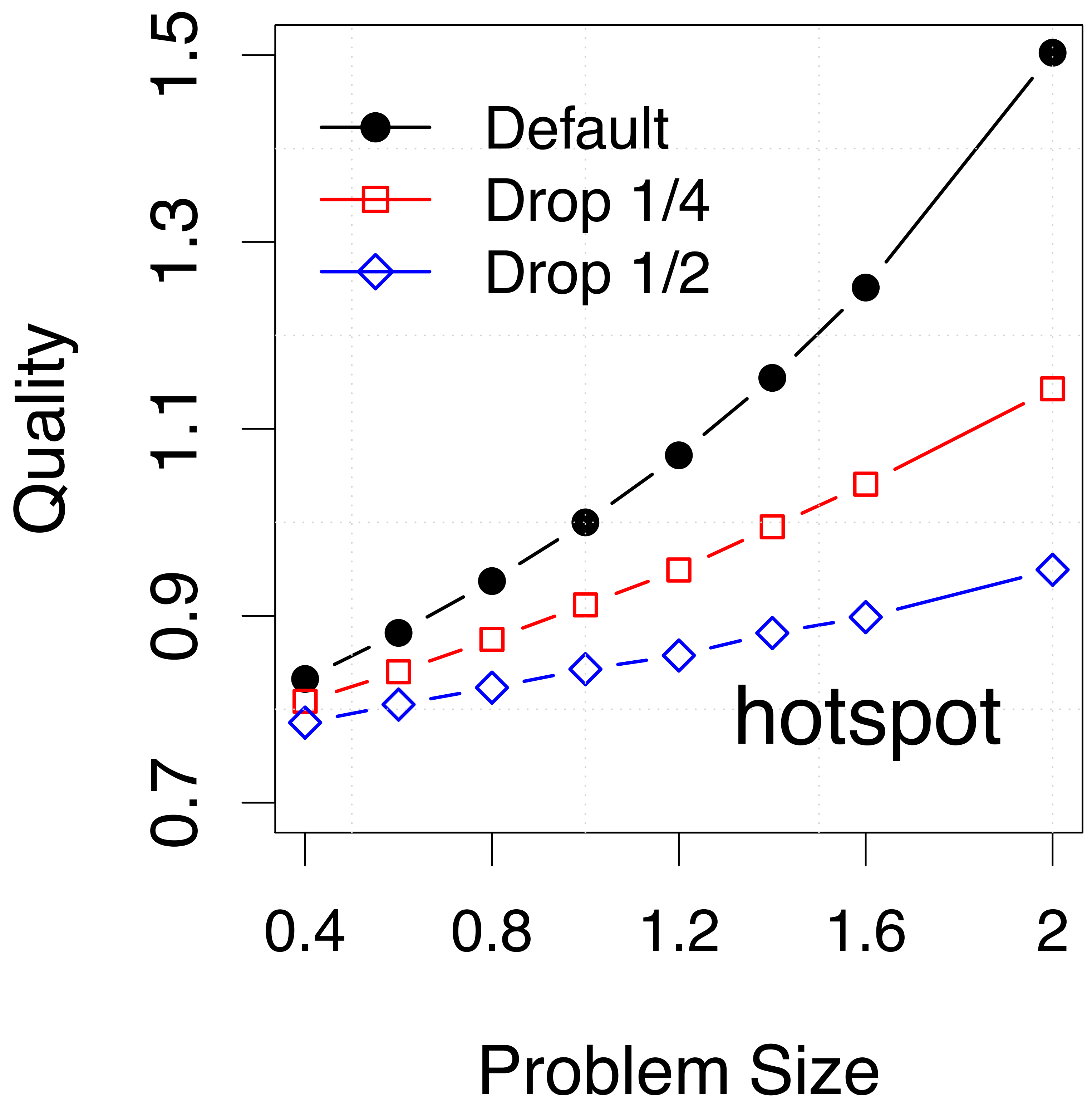
System Parameters	
Technology node: 11nm # cores: 288 # clusters: 36 (8 cores/cluster)	P_{MAX}= 100W T_{MIN}= 80°C Chip area ≈ 20mm x 20mm
Variation Parameters	
Correlation range: $\phi = 0.1$ Total $(\sigma/\mu)_{Vth} = 15\%$	Sample size: 100 chips Total $(\sigma/\mu)_{Leff} = 7.5\%$
Technology Parameters	
Vdd_{NOM}= 0.55V Vth_{NOM}= 0.33V	f_{NOM}= 1.0GHz $f_{network}$= 0.8GHz
Architectural Parameters	
Core-private mem: 64KB WT, 4-way, 2ns access, 64B line Network: bus inside cluster and 2D-torus across clusters	Cluster mem: 2MB WB, 16-way, 10ns access, 64B line Avg. mem round-trip access time (without contention): ≈80ns

Impact of Parametric Variation

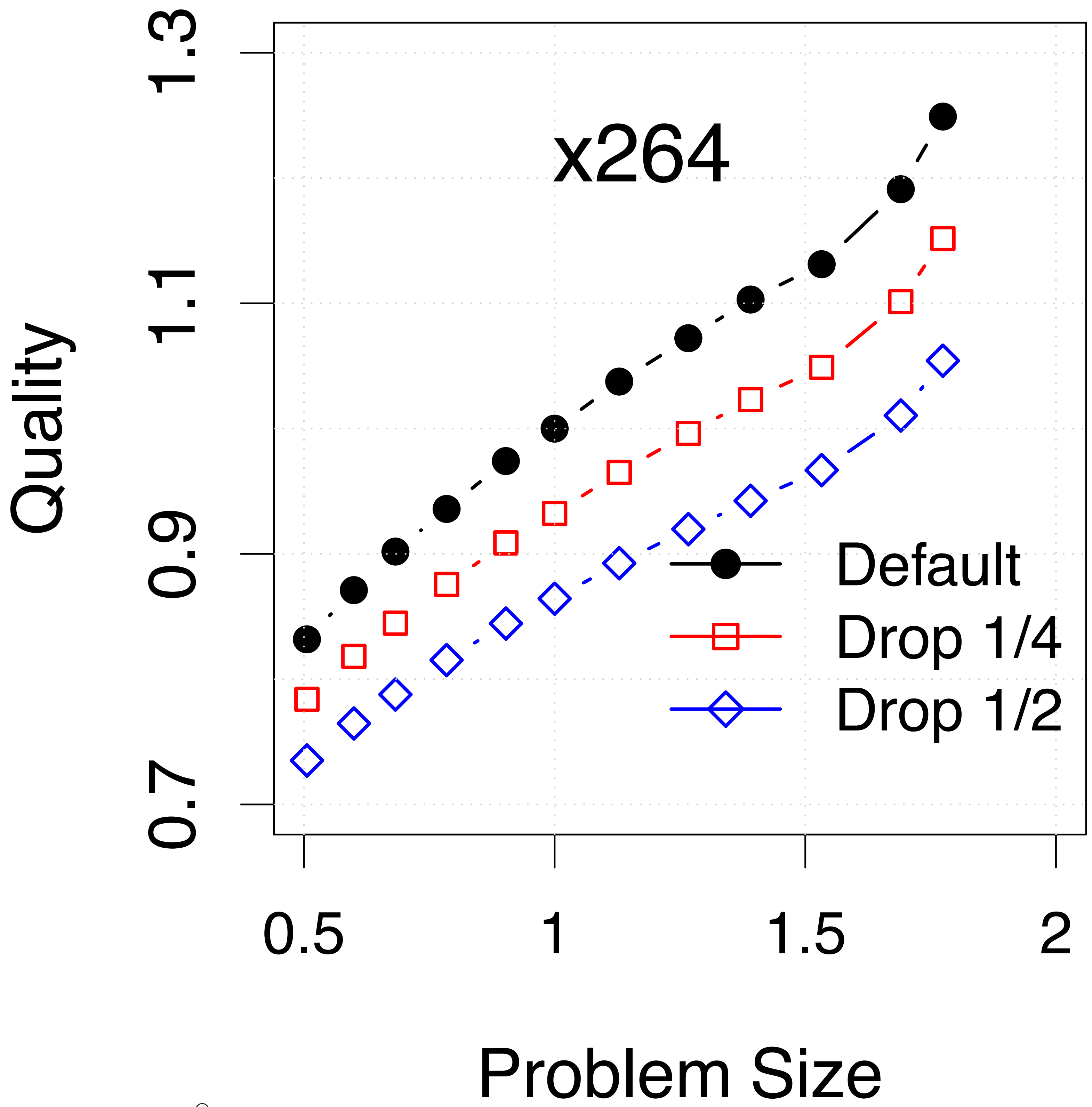




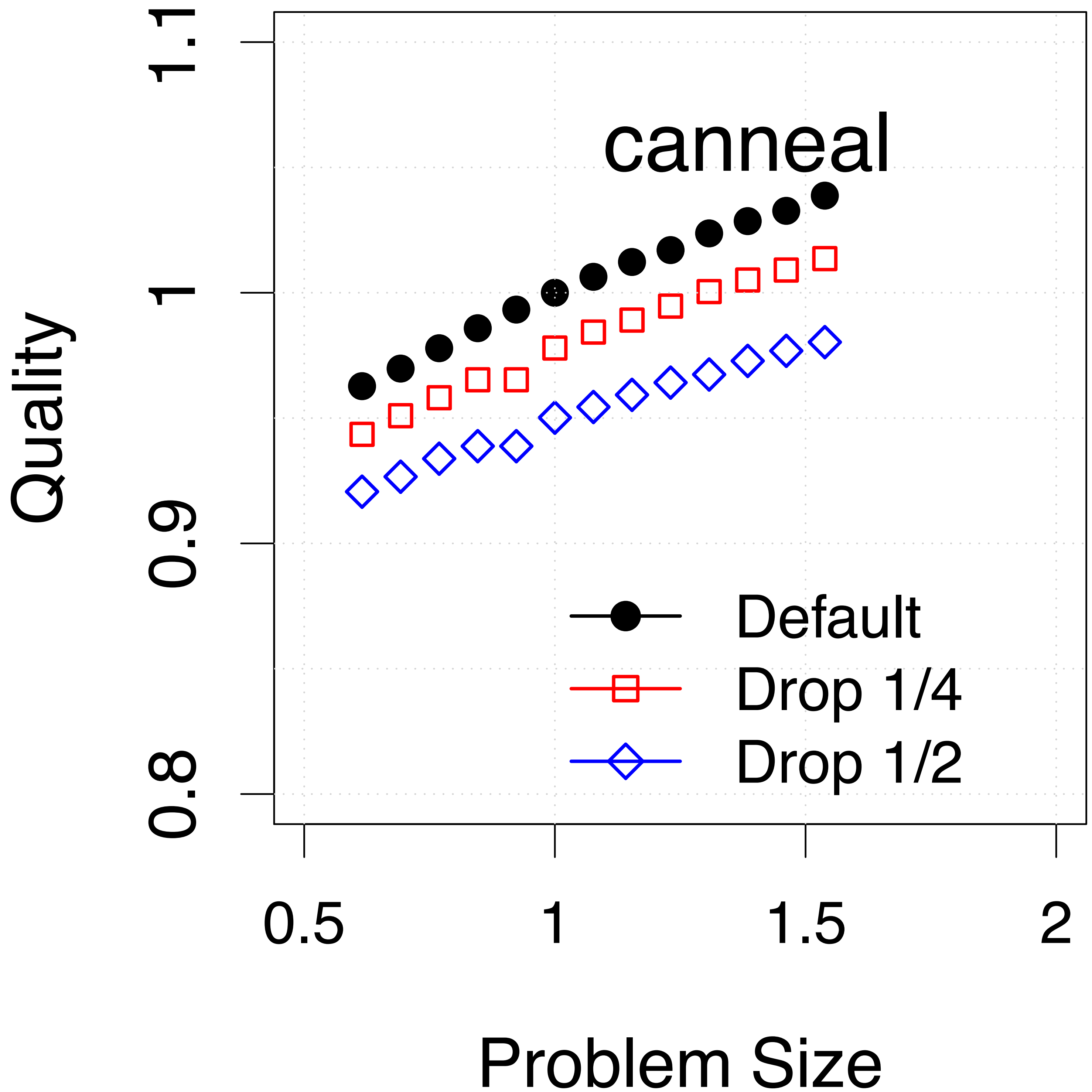
Problem Size vs. Quality



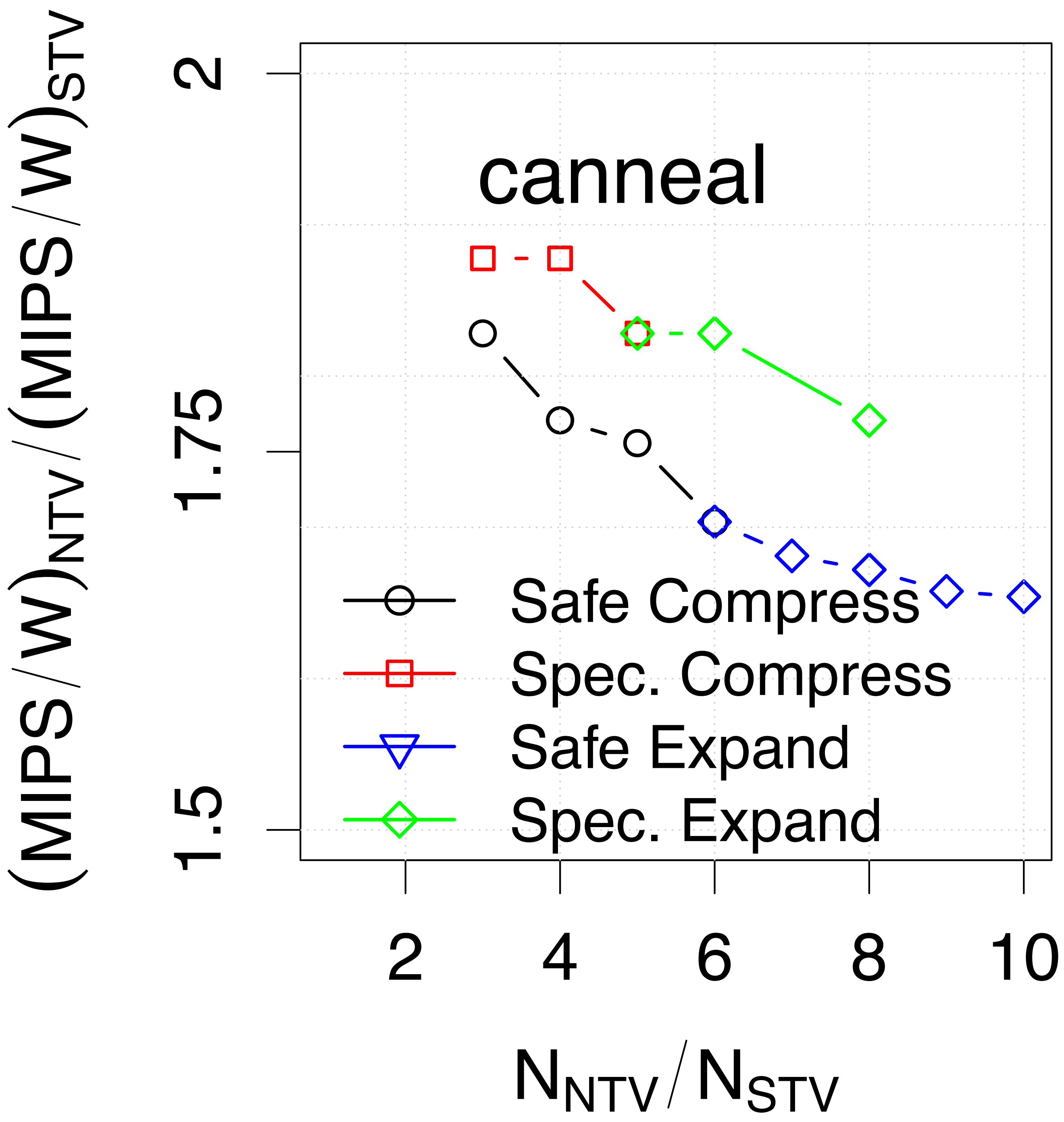
Problem Size vs. Quality



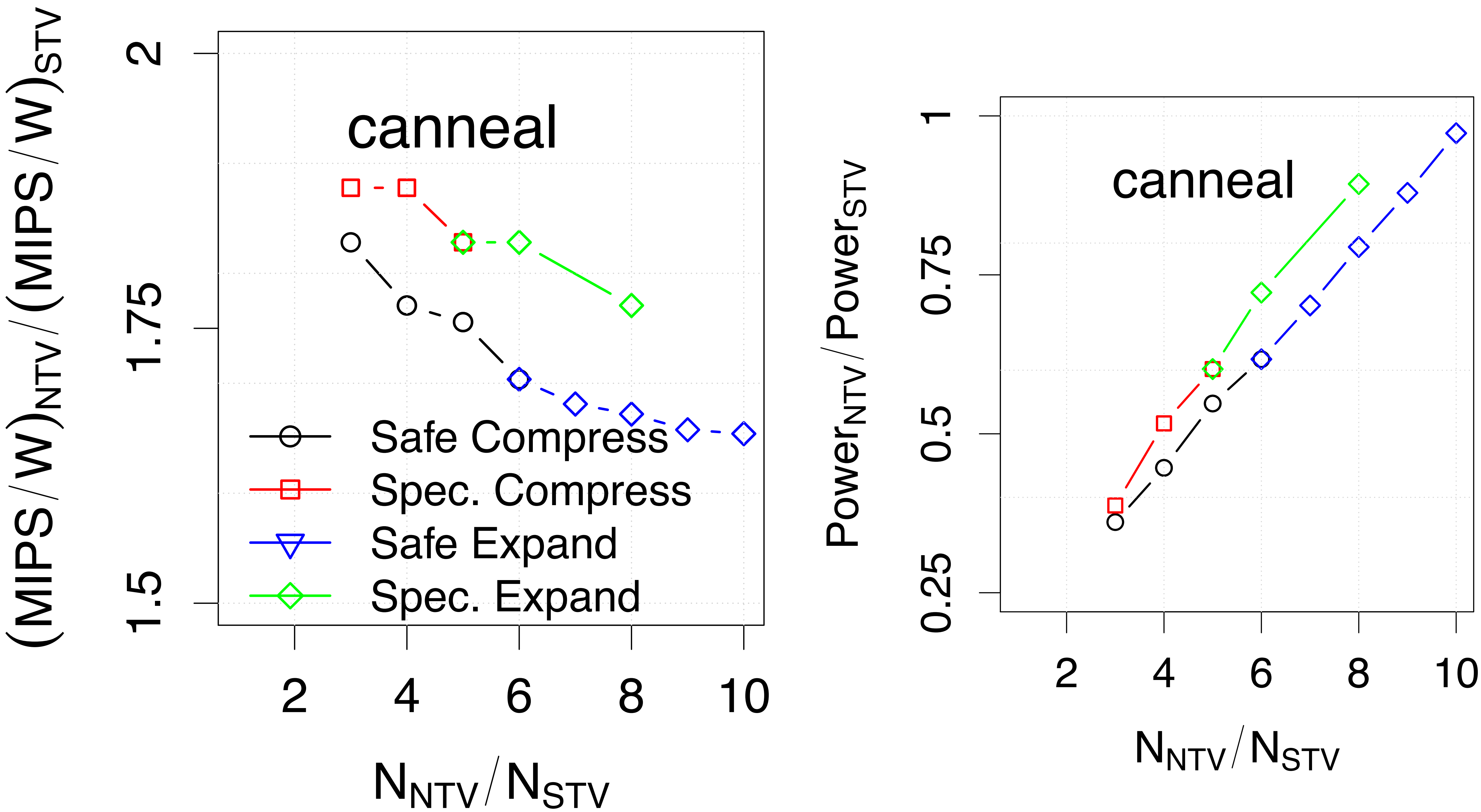
Problem Size vs. Quality of Computing



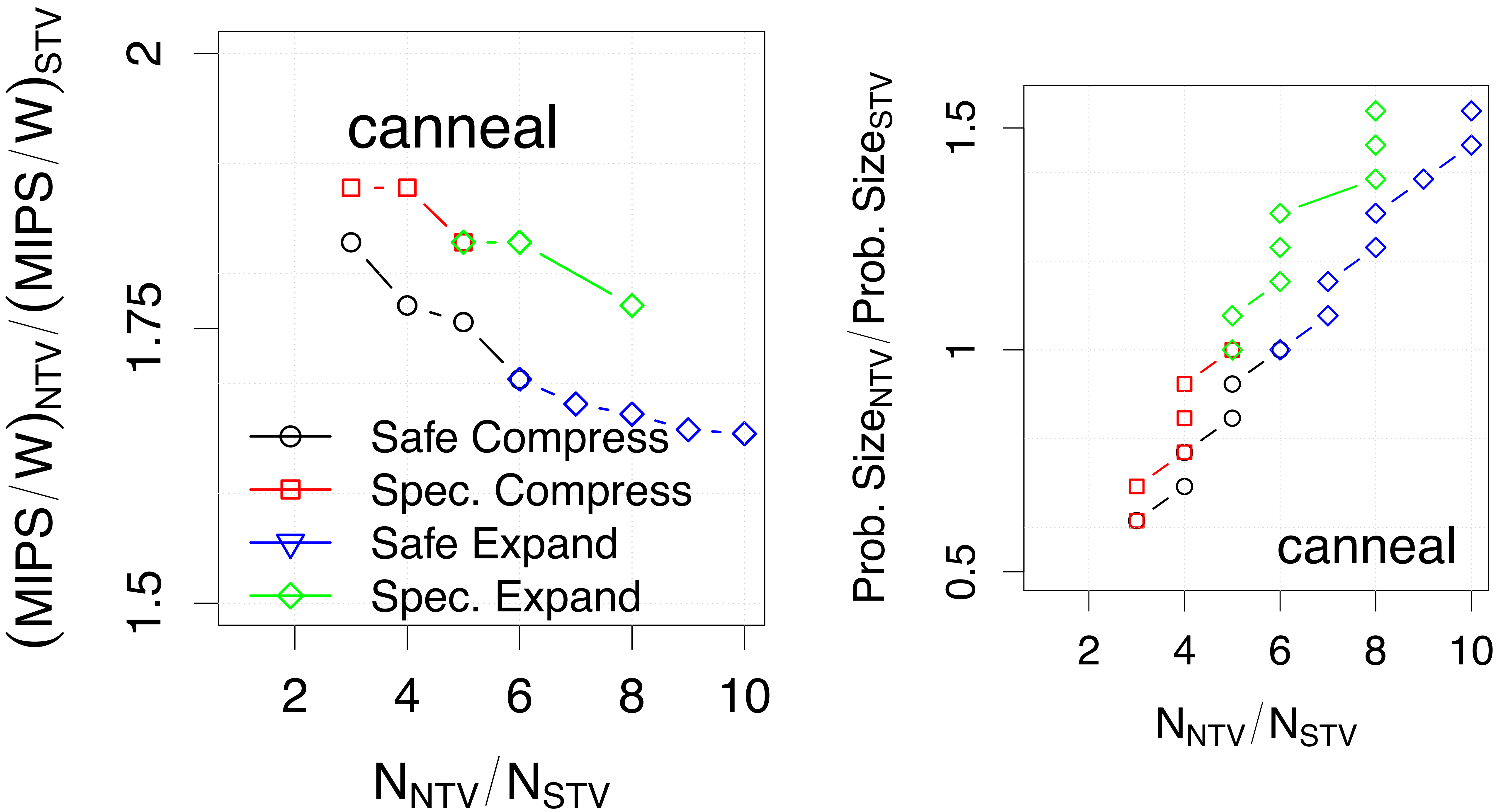
Iso-execution time fronts



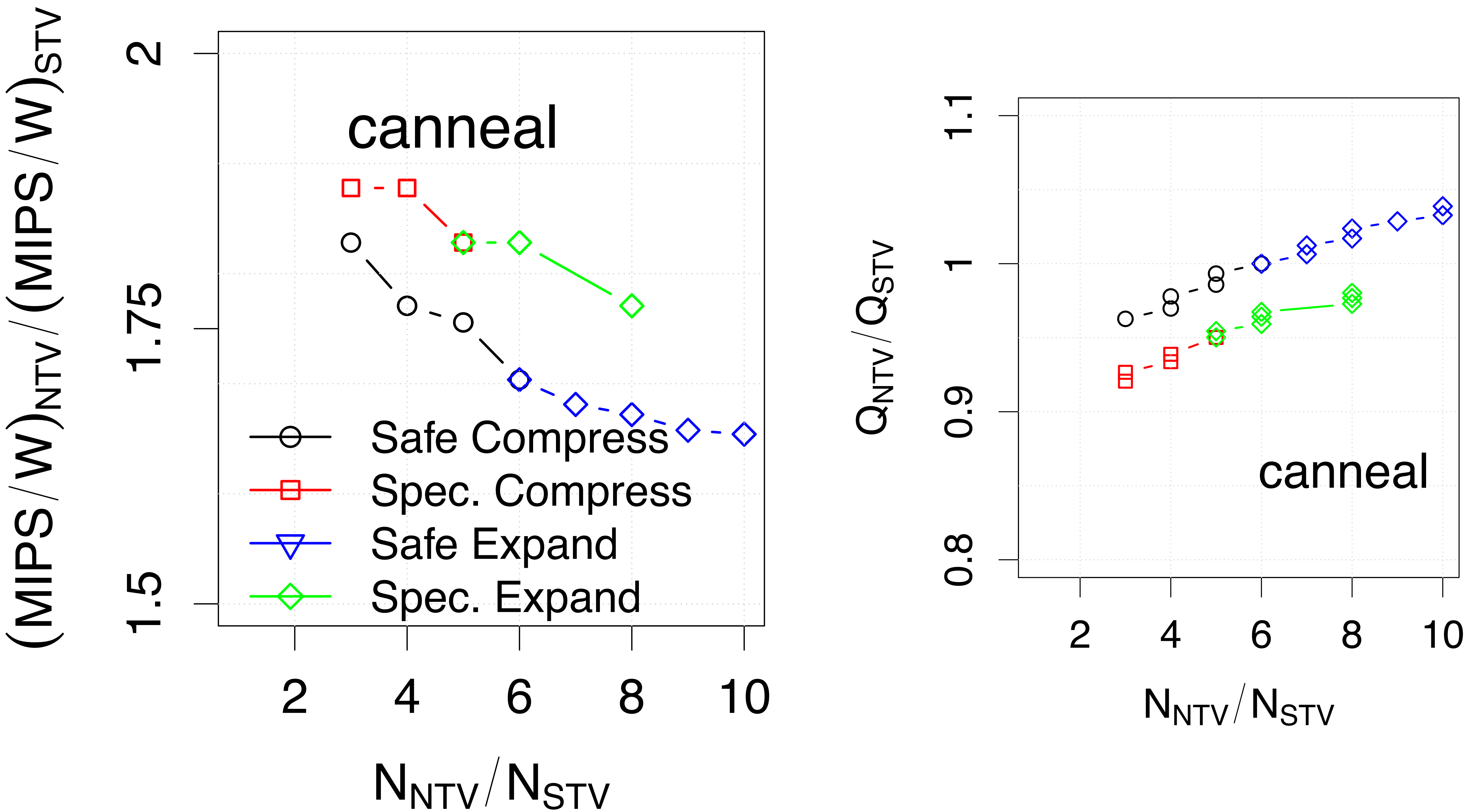
Iso-execution time fronts



Iso-execution time fronts



Evaluation


canneal
Safe Compress
Spec. Compress
Safe Expand
Spec. Expand
(MIPS/W)NTV/(MIPS/W)STV
2
1.75
1.5
2
4
6
8
10
NNTV/NSTV
canneal
QNTV/QSTV
1.1
1
0.9
0.8
2
4
6
8
10
NNTV/NSTV