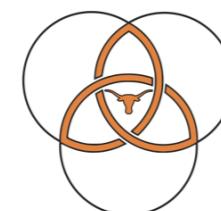


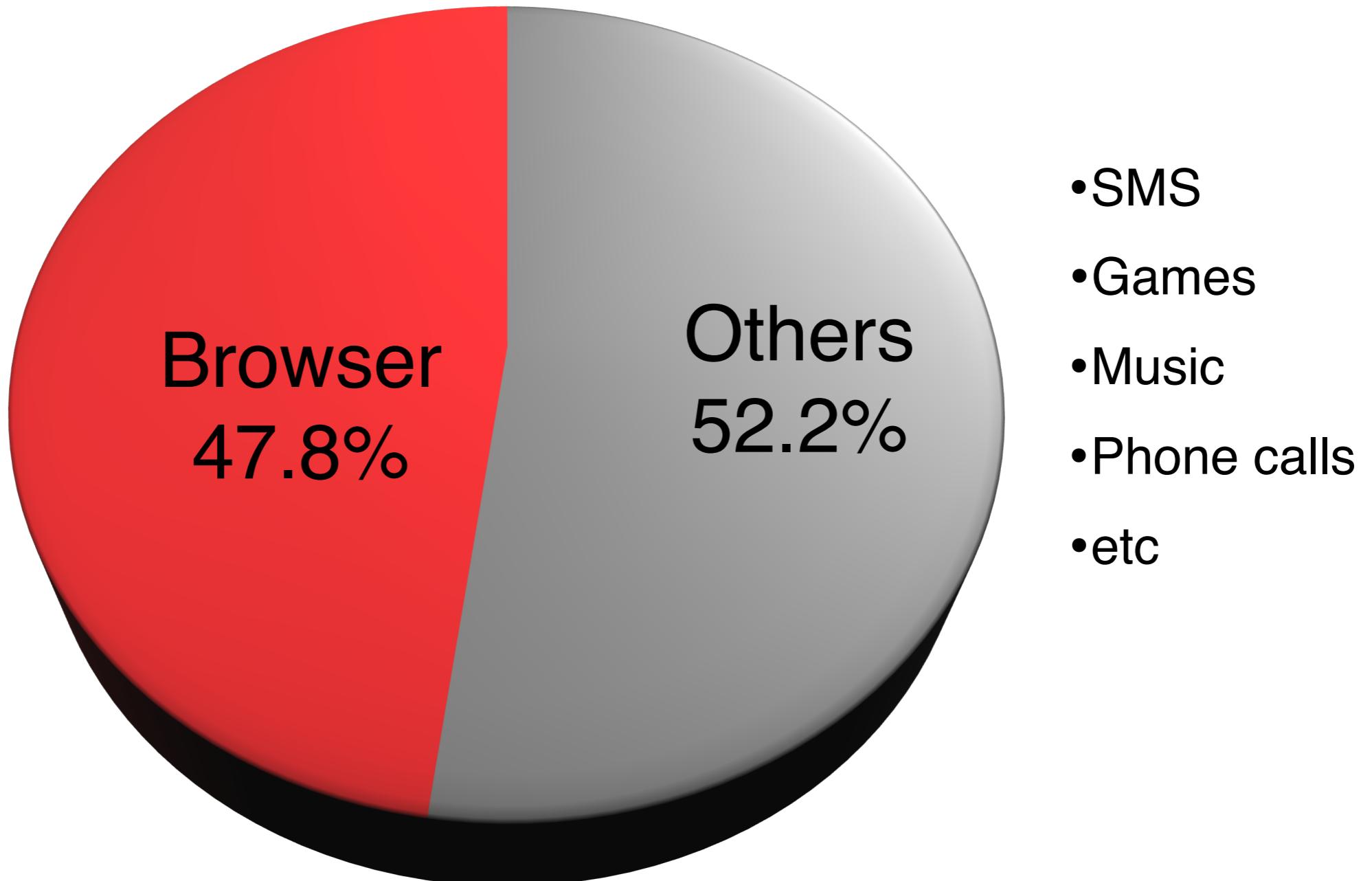
High-Performance and Energy-Efficient Mobile Web Browsing on Big/Little Systems

Yuhao Zhu
Vijay Janapa Reddi



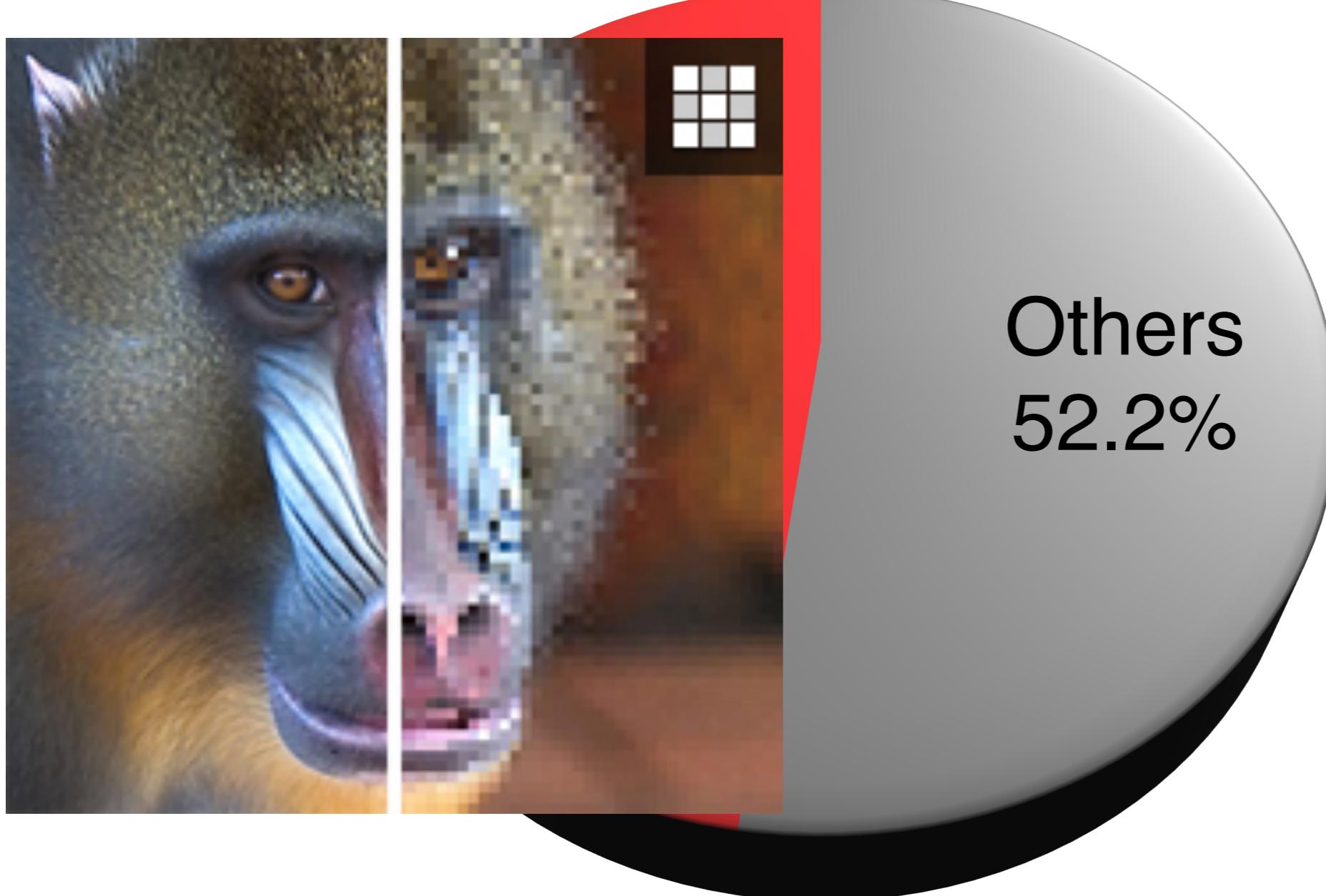
Trinity Research Group
The University of Texas at Austin

Web Browsing



Source: Flurry Analytics, 2011

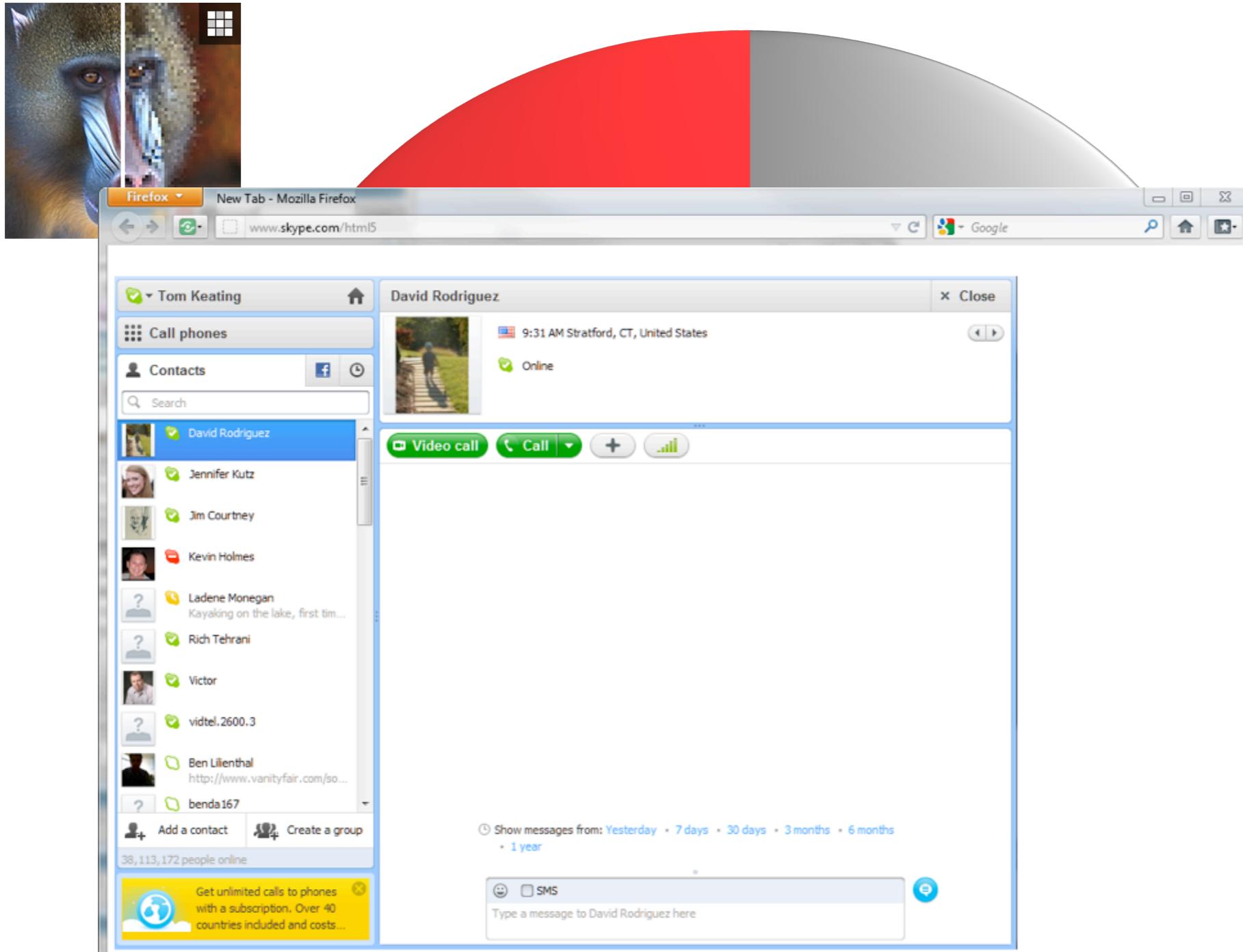
Web Browsing



- SMS
- Games
- Music
- Phone calls
- etc

Source: Flurry Analytics, 2011

Web Browsing



- SMS
- Games
- Music
- Phone calls
- etc

Source: Flurry Analytics, 2011

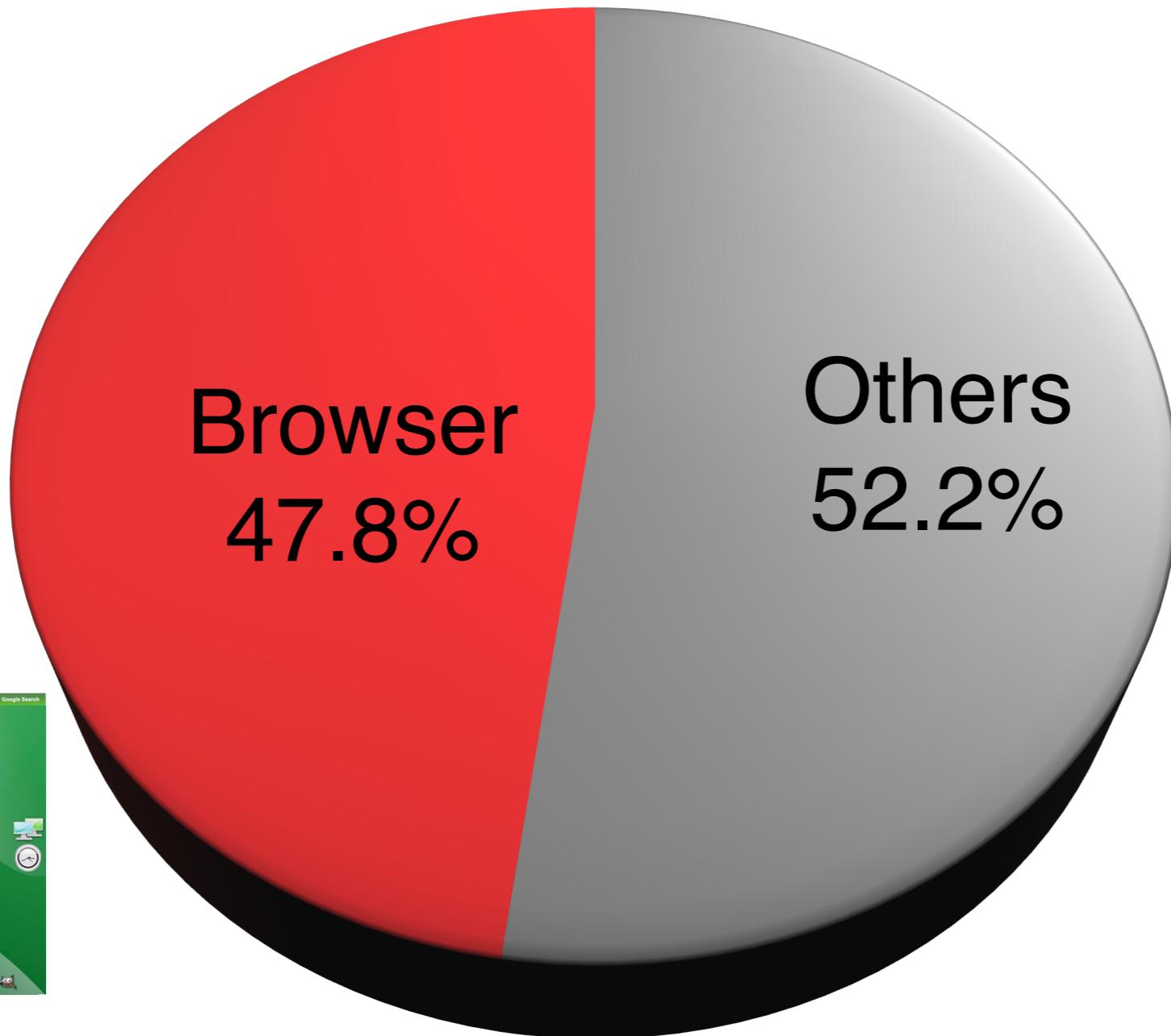
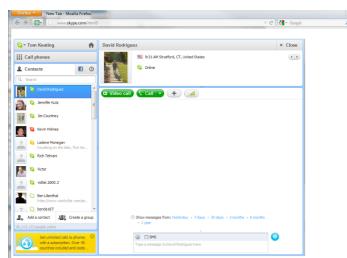
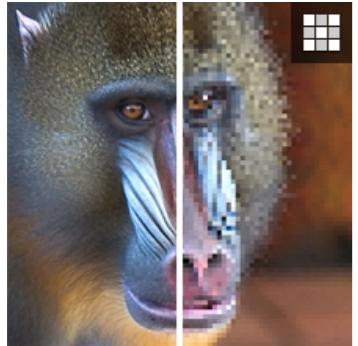
Web Browsing



- SMS
- Games
- Music
- Phone calls
- etc

Source: Flurry Analytics, 2011

Web Browsing

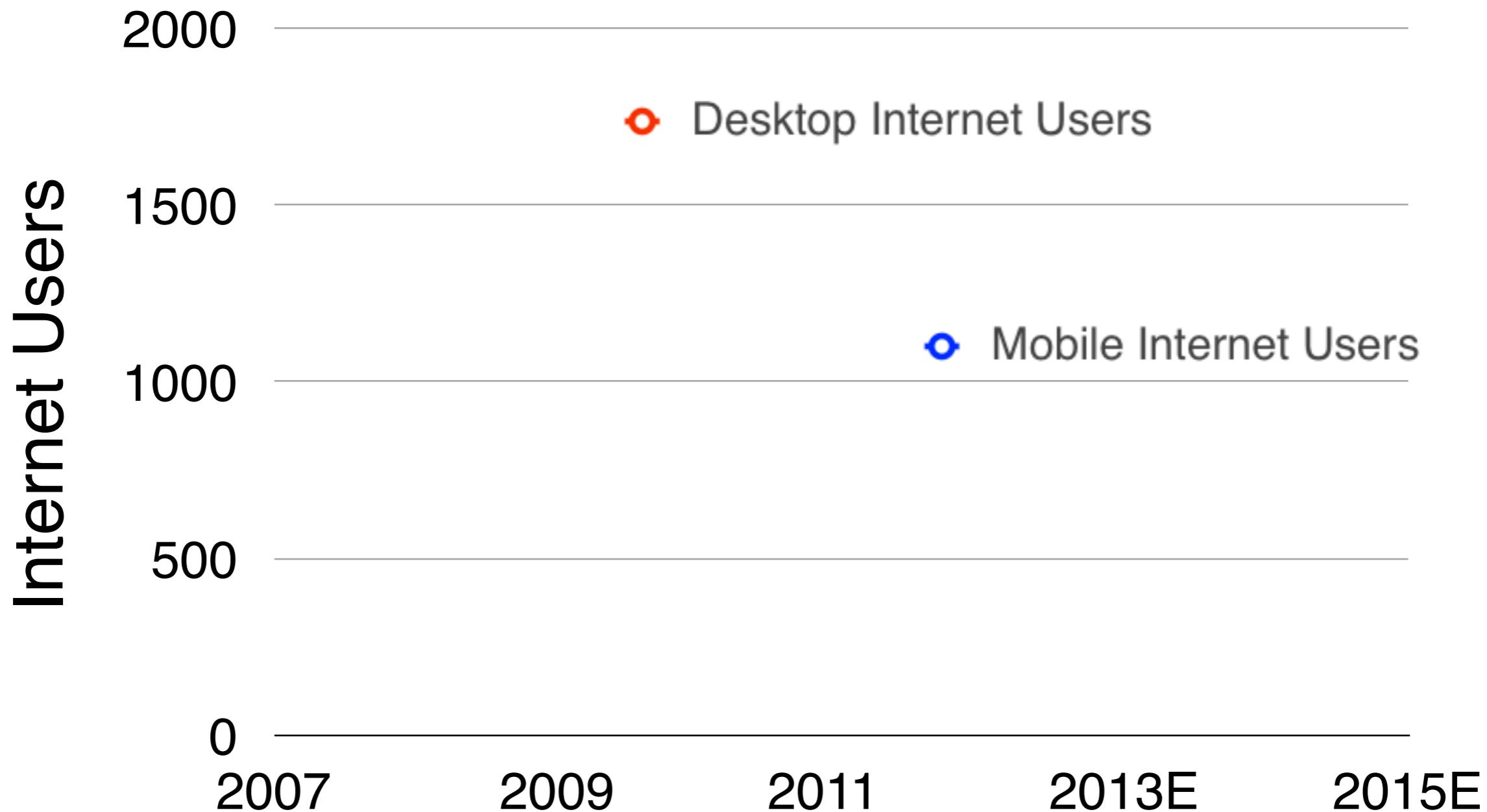


- SMS
- Games
- Music
- Phone calls
- etc

Source: Flurry Analytics, 2011

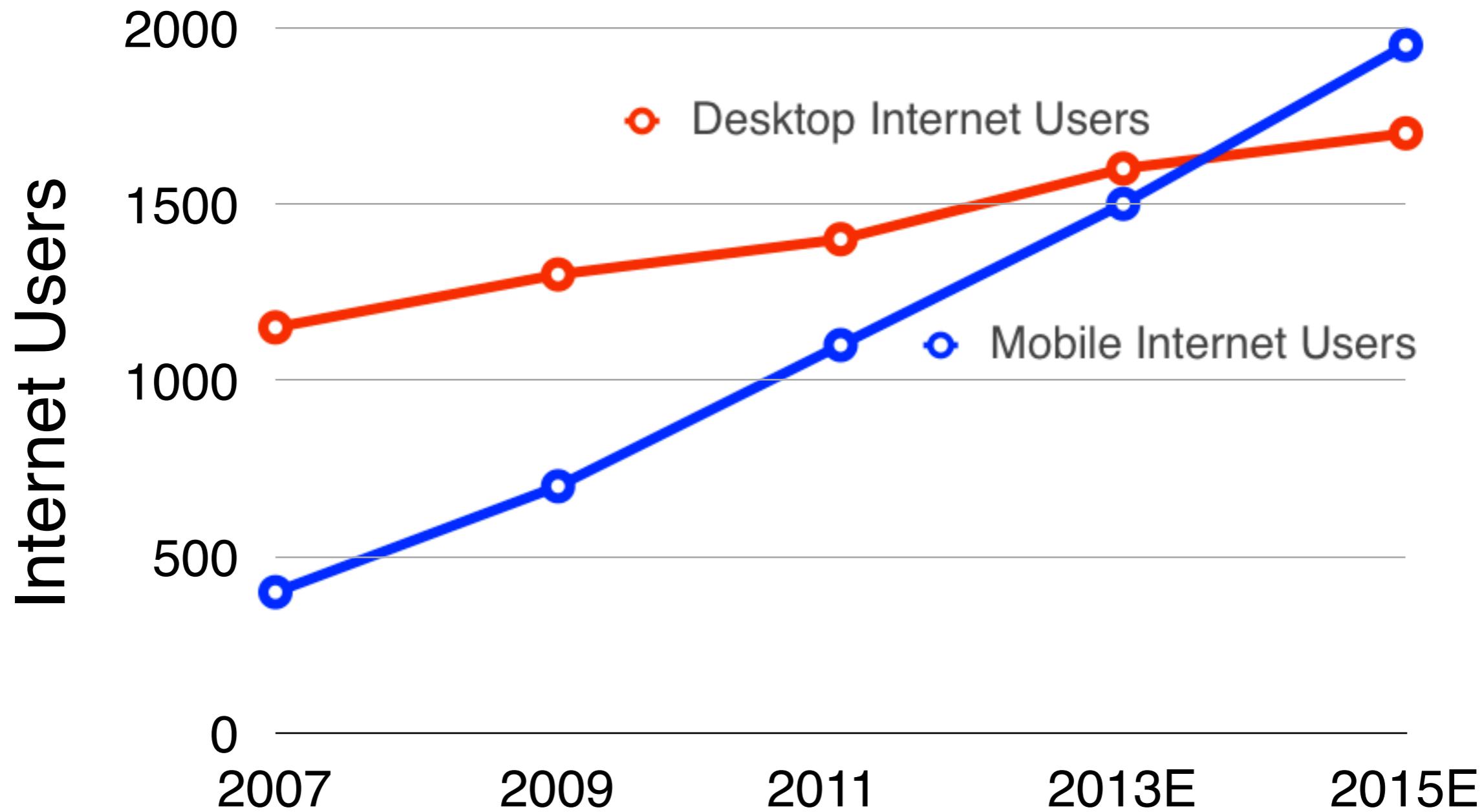
Mobile Web Browsing

Mobile Web Browsing



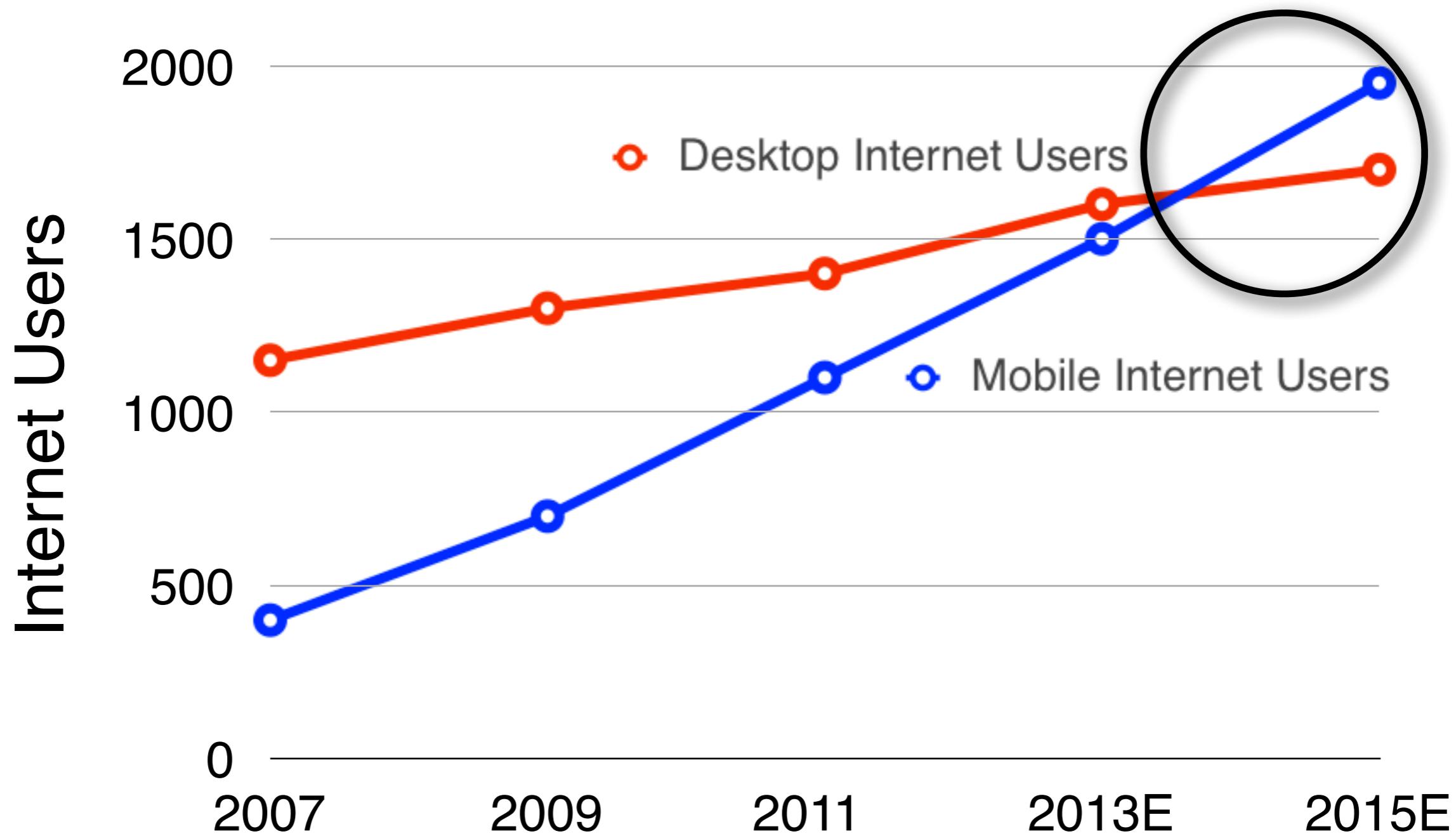
Source: Reproduced from Microsoft Tag, 2011

Mobile Web Browsing



Source: Reproduced from Microsoft Tag, 2011

Mobile Web Browsing

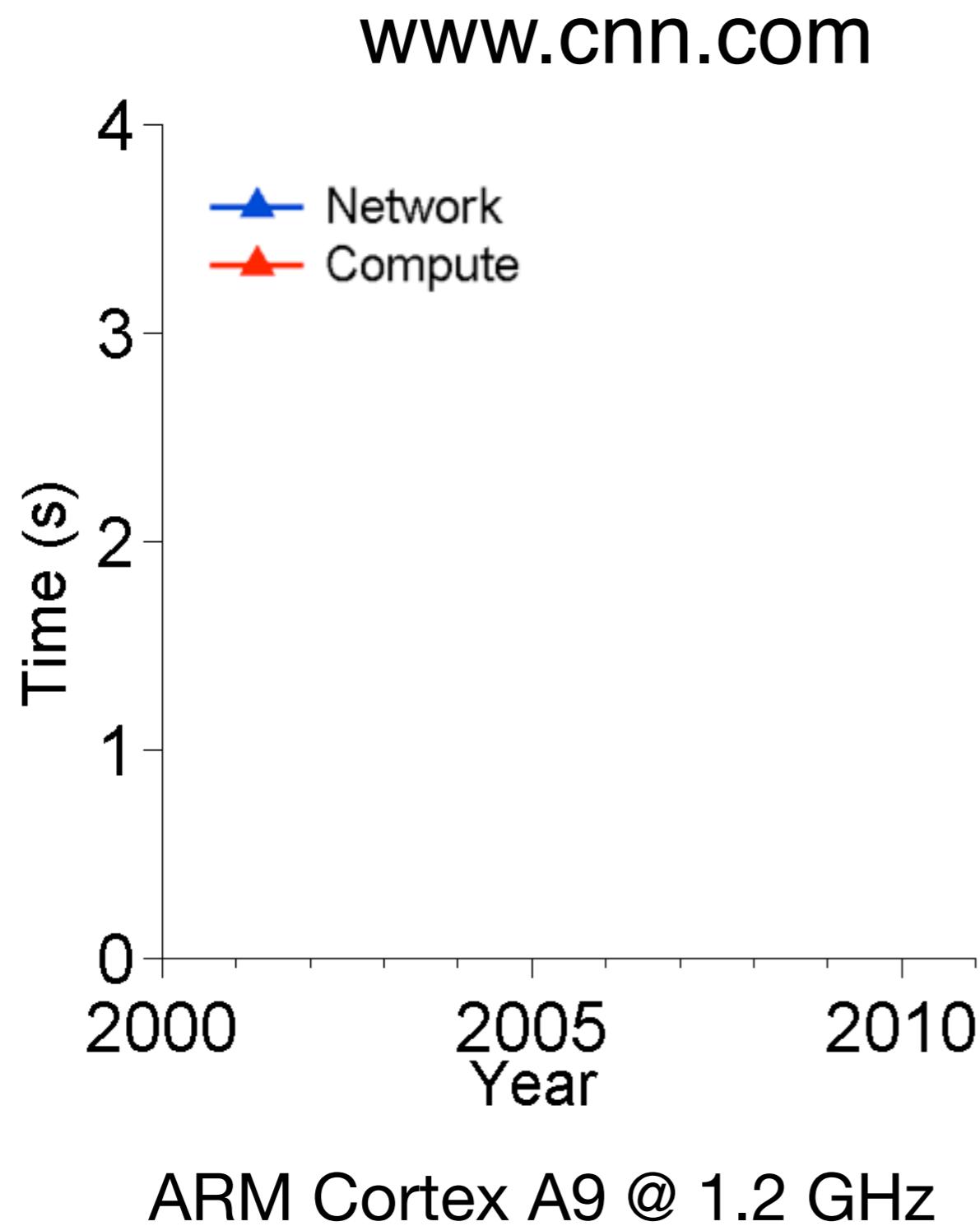


Source: Reproduced from Microsoft Tag, 2011

Increasing Computational Intensity

Webpages are more computationally intensive

Increasing Computational Intensity

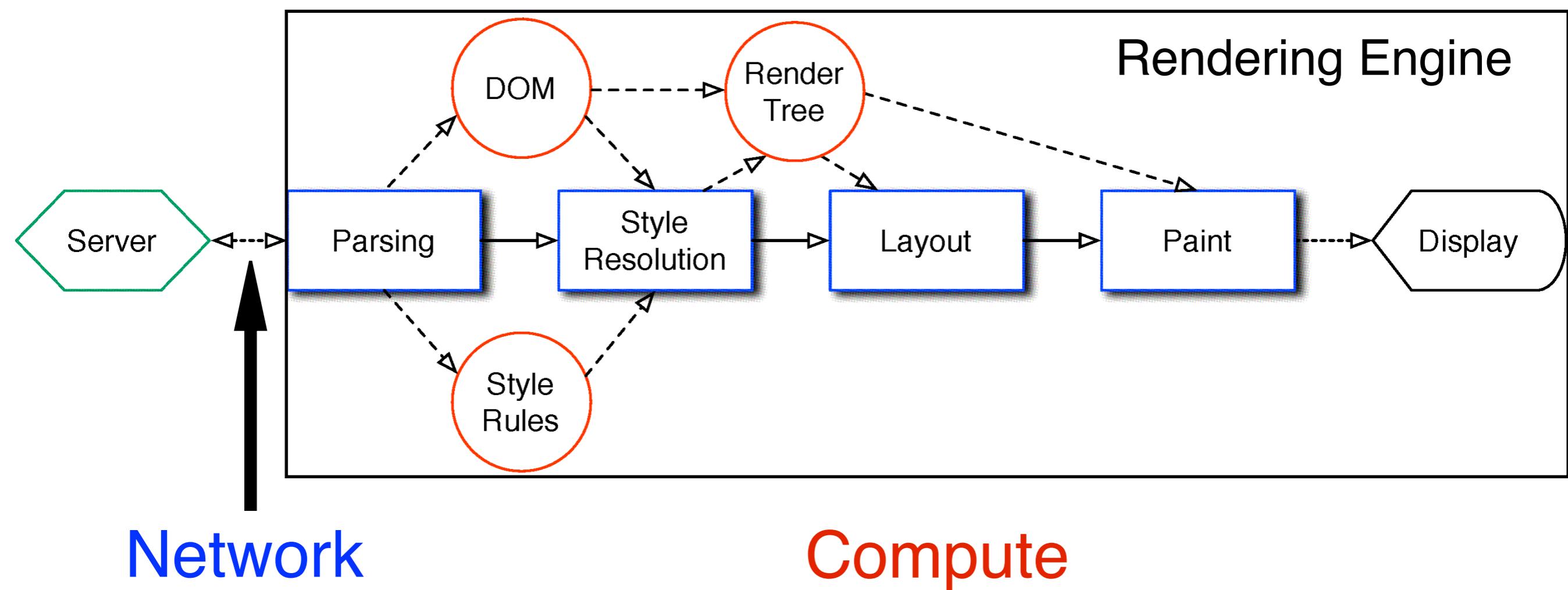


Webpages are more computationally intensive

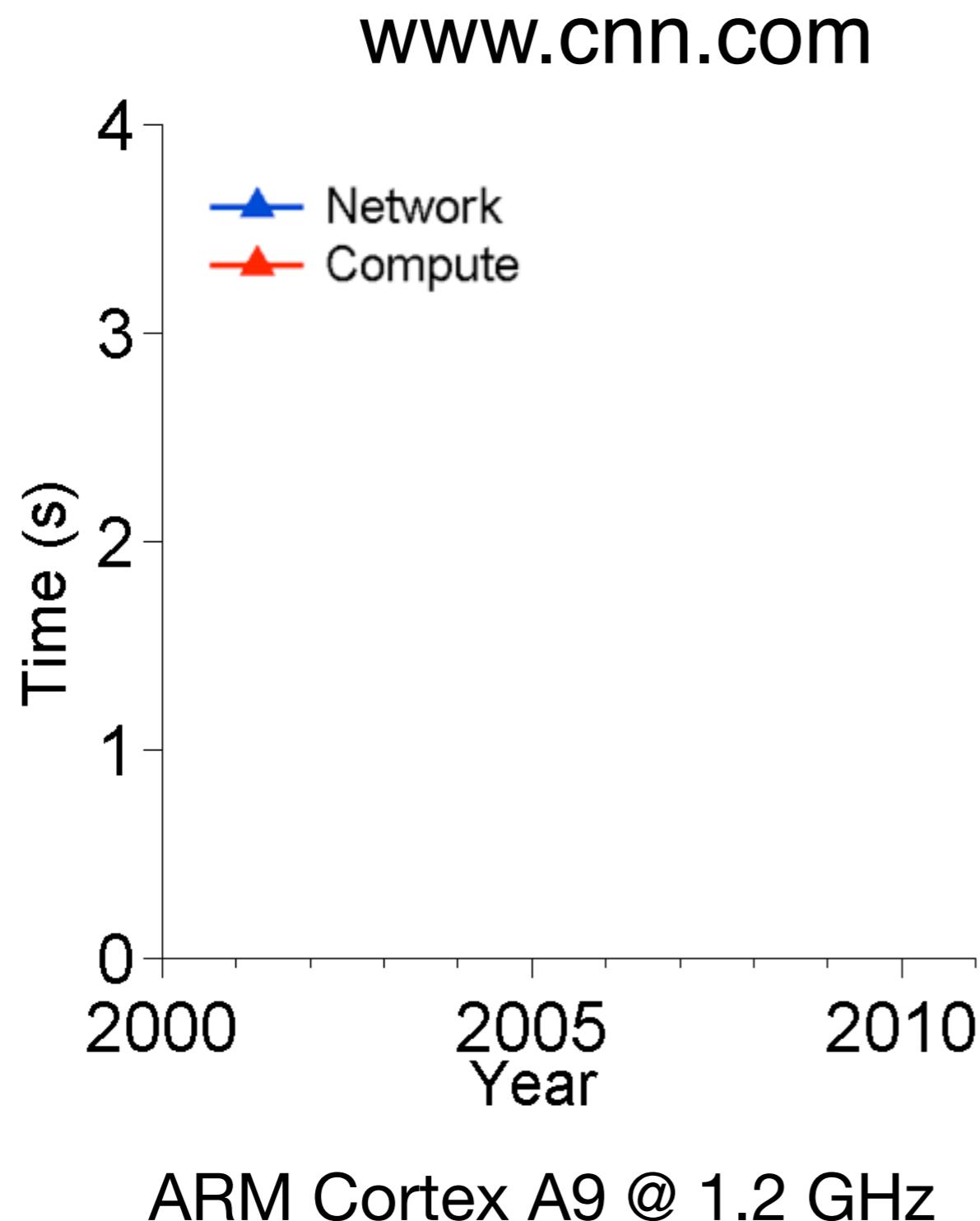
Increasing Computational Intensity



Increasing Computational Intensity

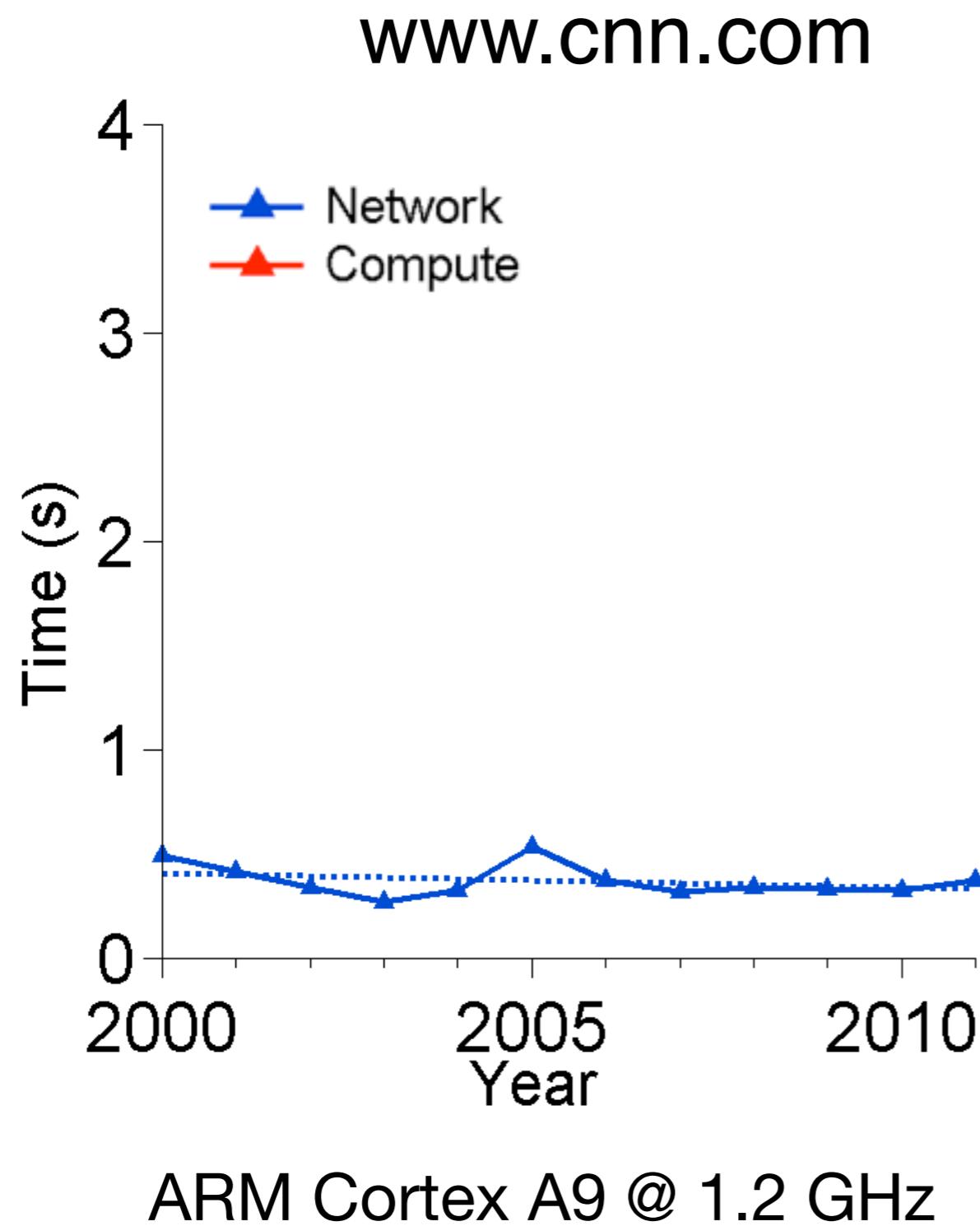


Increasing Computational Intensity



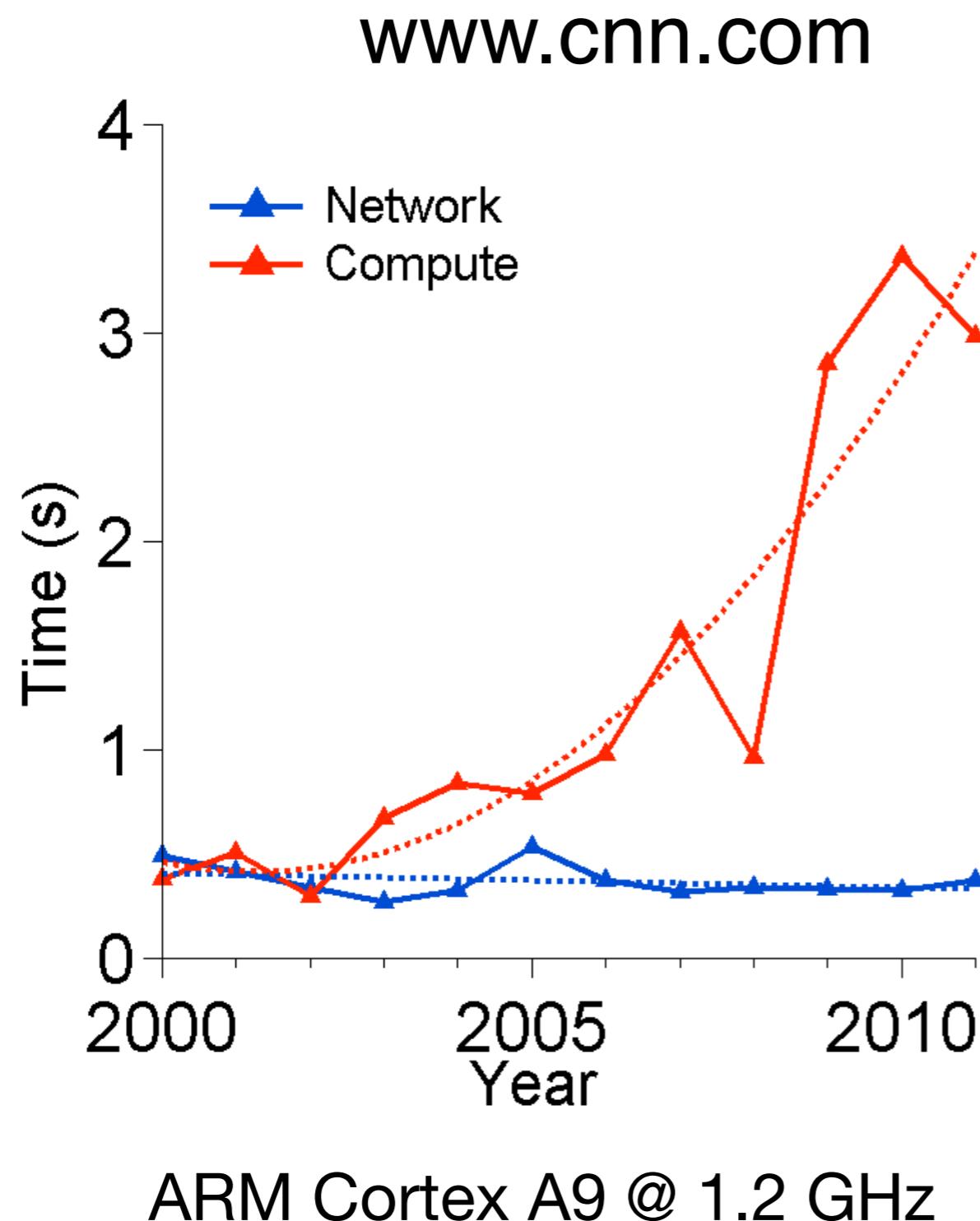
Webpages are becoming computationally intensive

Increasing Computational Intensity



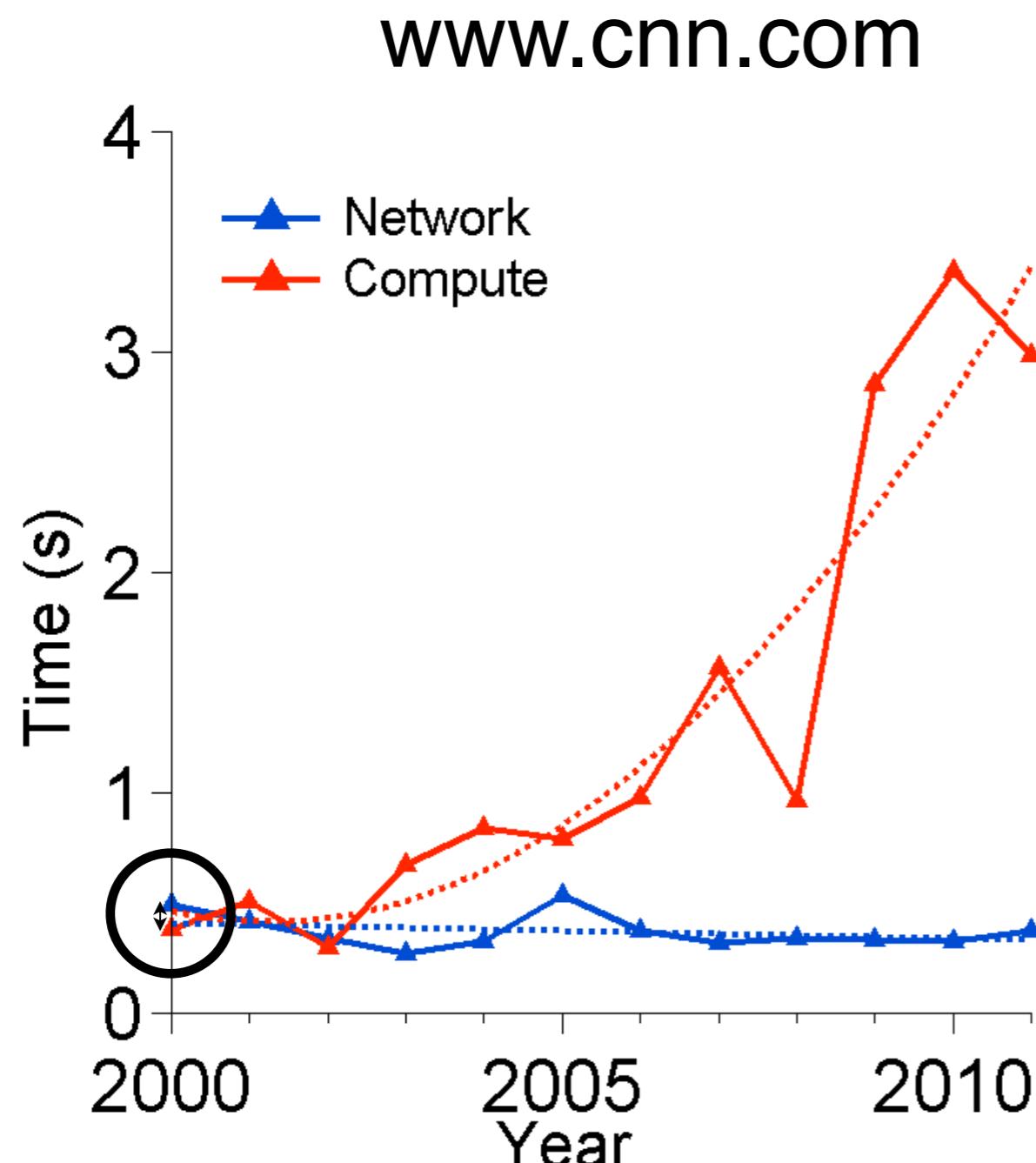
Webpages are becoming computationally intensive

Increasing Computational Intensity



Webpages are becoming computationally intensive

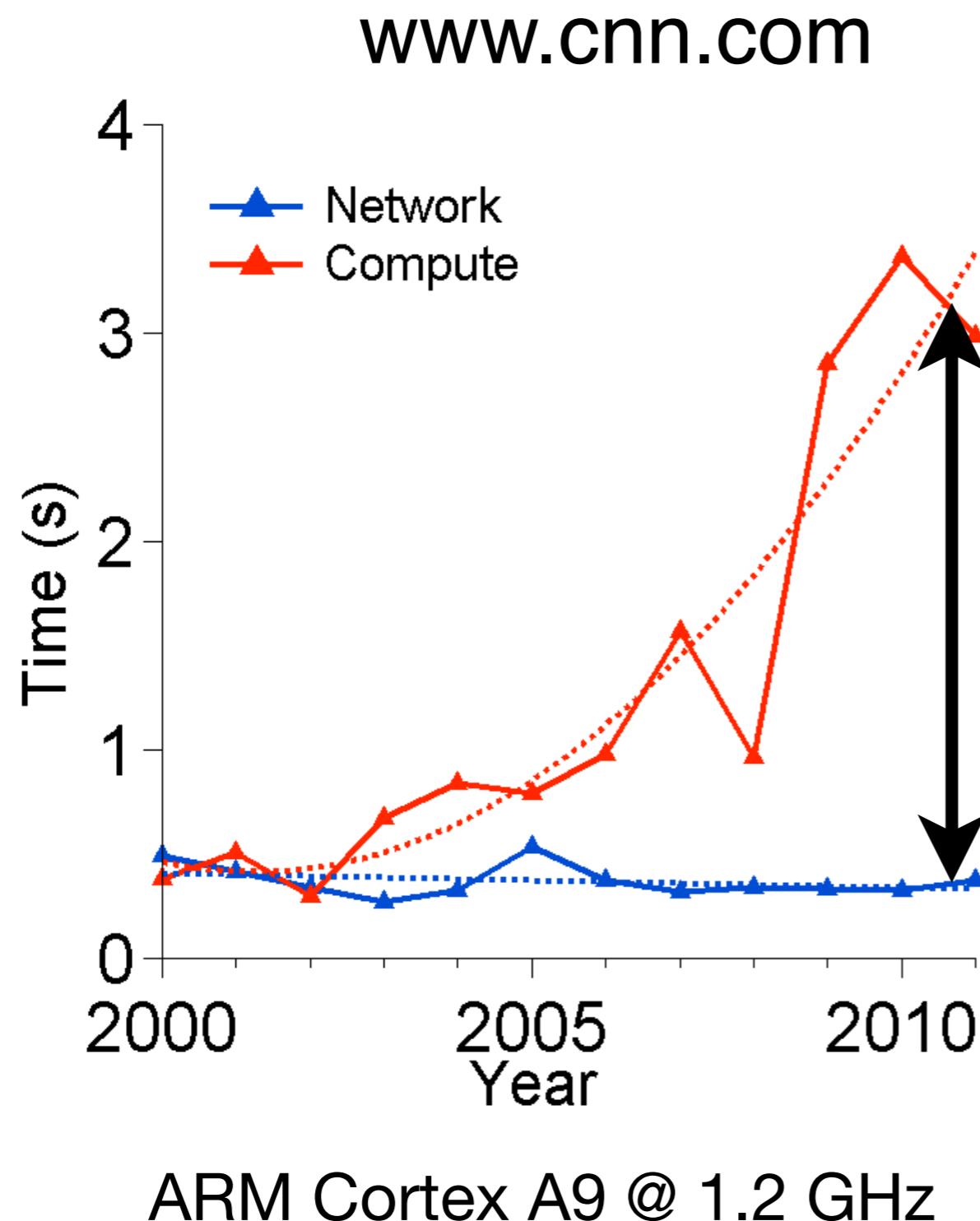
Increasing Computational Intensity



Webpages are becoming computationally intensive

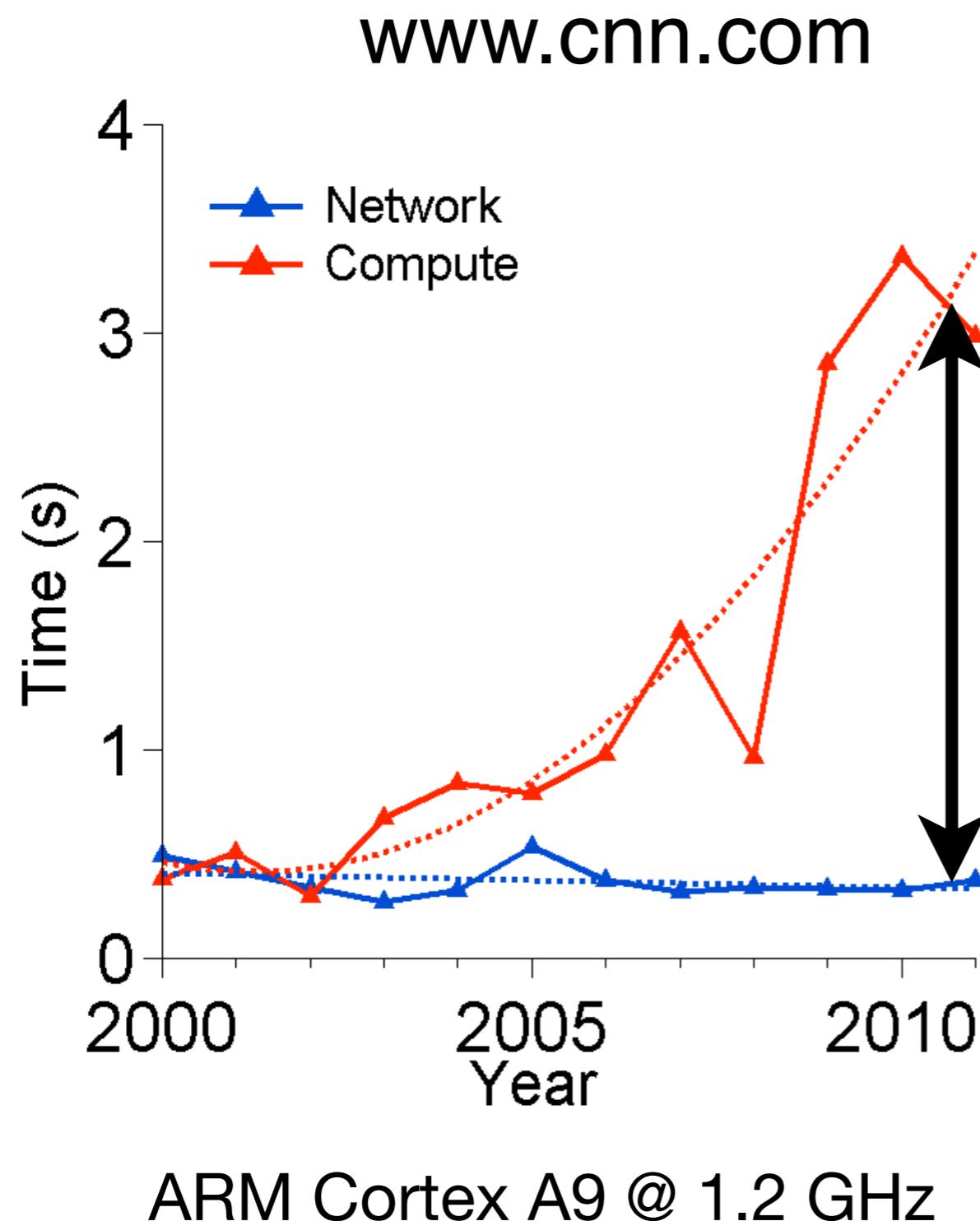
ARM Cortex A9 @ 1.2 GHz

Increasing Computational Intensity



Webpages are becoming computationally intensive

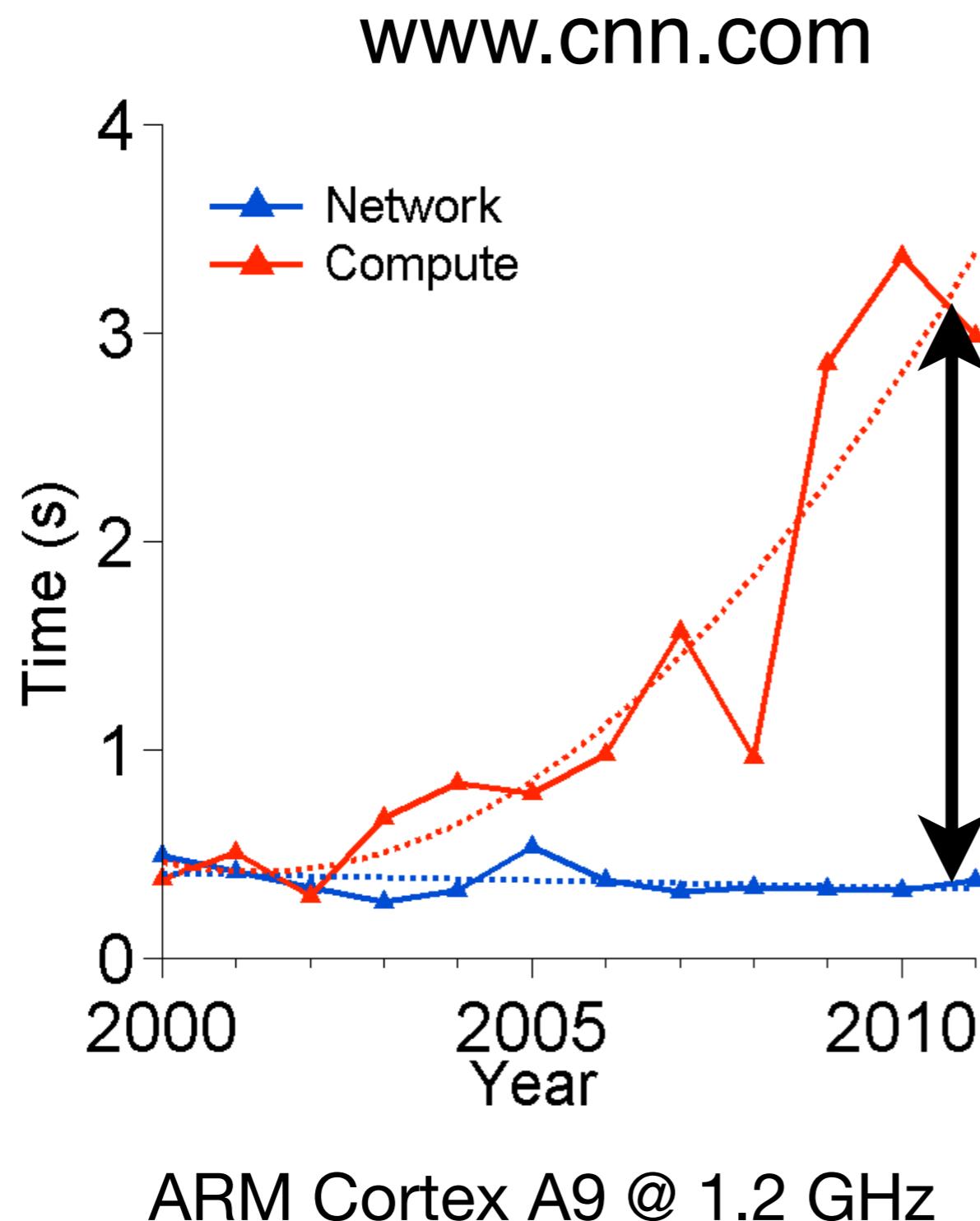
Increasing Computational Intensity



Webpages are becoming computationally intensive

Mobile systems demand high performance!

Increasing Computational Intensity

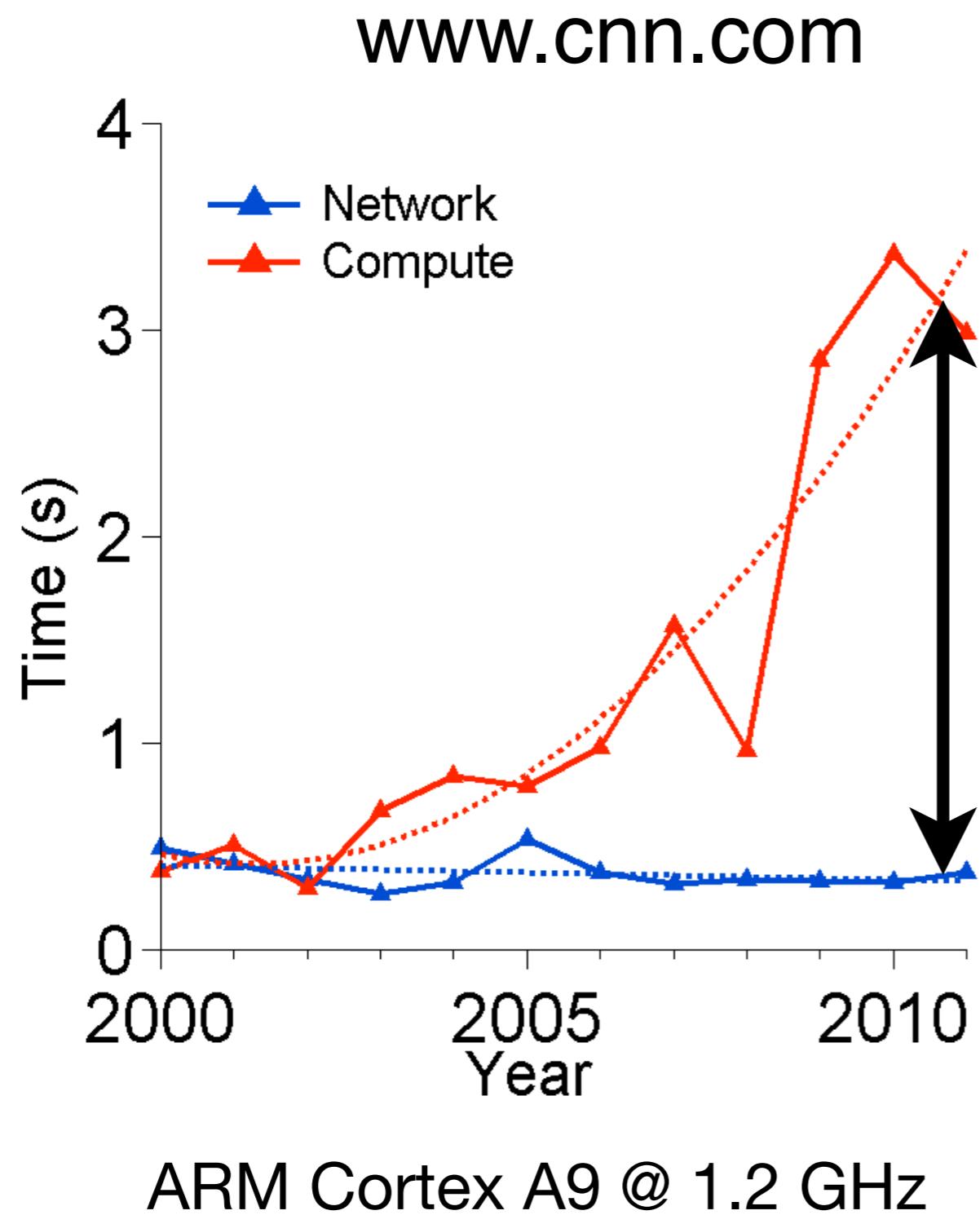


Webpages are becoming computationally intensive

Mobile systems demand high performance!

Can we deploy a mobile system purely driven by performance objectives?

Increasing Computational Intensity



Webpages are becoming computationally intensive

Mobile systems demand high performance!

Can we deploy a mobile system purely driven by performance objectives?

No. Mobile devices are battery-constrained!

Executive Summary

Challenge: How to design the system architecture that guarantees both high performance and energy efficiency?

Executive Summary

Challenge: How to design the system architecture that guarantees both high performance and energy efficiency?

Alternatives: Single big/little core; symmetric designs; asymmetric designs;

Executive Summary

Challenge: How to design the system architecture that guarantees both high performance and energy efficiency?

Alternatives: Single big/little core; symmetric designs; asymmetric designs; **Big/Little systems**

- Different microarchitectures (Big, OoO + little, in-order)
- Different operating points (DVFS)

Executive Summary

Challenge: How to design the system architecture that guarantees both high performance and energy efficiency?

Alternatives: Single big/little core; symmetric designs; asymmetric designs; **Big/Little systems**

- Different microarchitectures (Big, OoO + little, in-order)
- Different operating points (DVFS)

Key insight: Webpages have different characteristics that lead to load time and energy consumption variance

Executive Summary

Challenge: How to design the system architecture that guarantees both high performance and energy efficiency?

Alternatives: Single big/little core; symmetric designs; asymmetric designs; **Big/Little systems**

- Different microarchitectures (Big, OoO + little, in-order)
- Different operating points (DVFS)

Key insight: Webpages have different characteristics that lead to load time and energy consumption variance

Solution: Predict \langle core, frequency \rangle configuration and schedule webpages accordingly

Software Setup



Independent of the particular browser

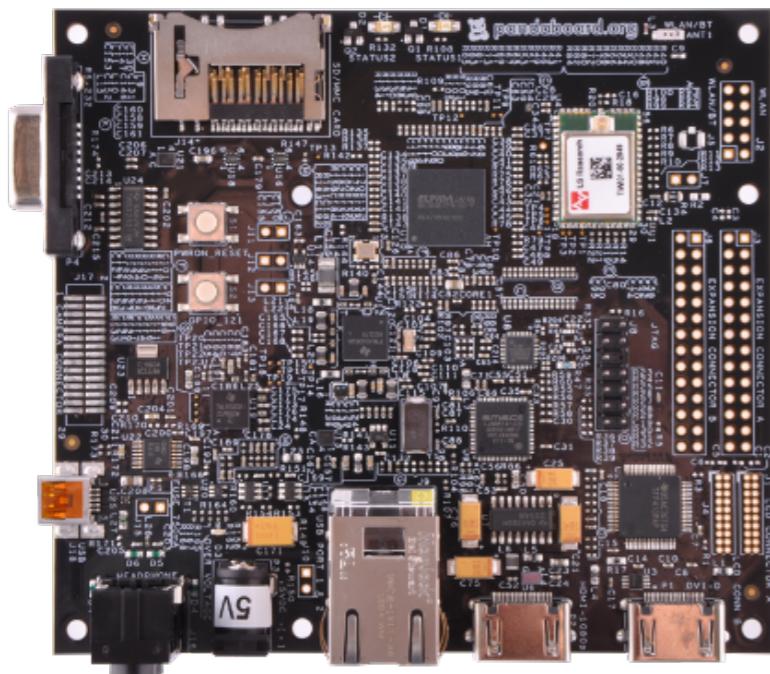
- We studied the Gecko rendering engine in Firefox
- Excluded boot-strap and shut-down effects
- Disabled browser cache

A screenshot of the Alexa website showing the 'Top Sites' section. The page lists the top 5 websites: Google, Facebook, YouTube, Yahoo!, and Blogger.com. Each entry includes a brief description, a star rating, and links to 'Search Analytics' and 'Audience'. Red arrows point from the text descriptions to red boxes labeled with the content analysis type: 'Content Based On Keyword Terms' for Google, 'Content Based On Social Information' for Facebook, 'Content Based On Video' for YouTube, 'Content Based On Keyword Terms' for Yahoo!, and 'Content Based On The Written Word' for Blogger.com.

Rank	Website	Description	Content Type
1	Google	Enables users to search the Web, Usenet, and images. Features incl More	Content Based On Keyword Terms
2	Facebook	A social utility that connects people, to keep up with friends, upload p	Content Based On Social Information
3	YouTube	YouTube is a way to get your videos to the people who matter to you.	Content Based On Video
4	Yahoo!	A major internet portal and service provider offering search results, cu	Content Based On Keyword Terms
5	Blogger.com	Free, automated weblog publishing tool that sends updates to a site v	Content Based On The Written Word

- Hottest 5,000 webpages from www.alexa.com
- Downloaded and mapped to the memory

Hardware Setup



Big core: ARM Cortex A9: OoO with 4 issue (e.g. in Tegra 3-based tablets)

PandaBoard ES Rev B1, 45nm

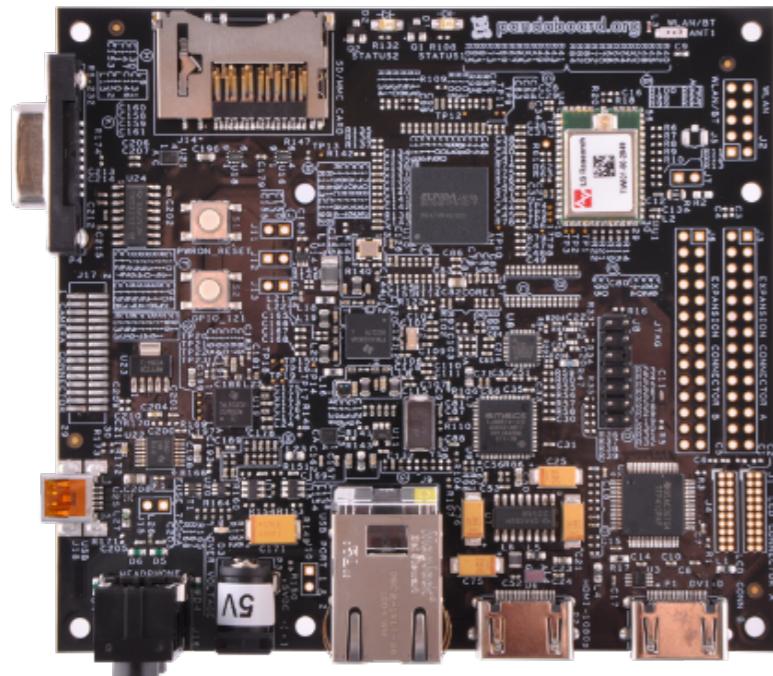
DVFS: 350 MHz, 0.83 V

700 MHz, 1.01 V

920 MHz, 1.11 V

1.2 GHz, 1.27 V

Hardware Setup



Big core: ARM Cortex A9: OoO with 4 issue (e.g. in Tegra 3-based tablets)

PandaBoard ES Rev B1, 45nm

DVFS: 350 MHz, 0.83 V

700 MHz, 1.01 V

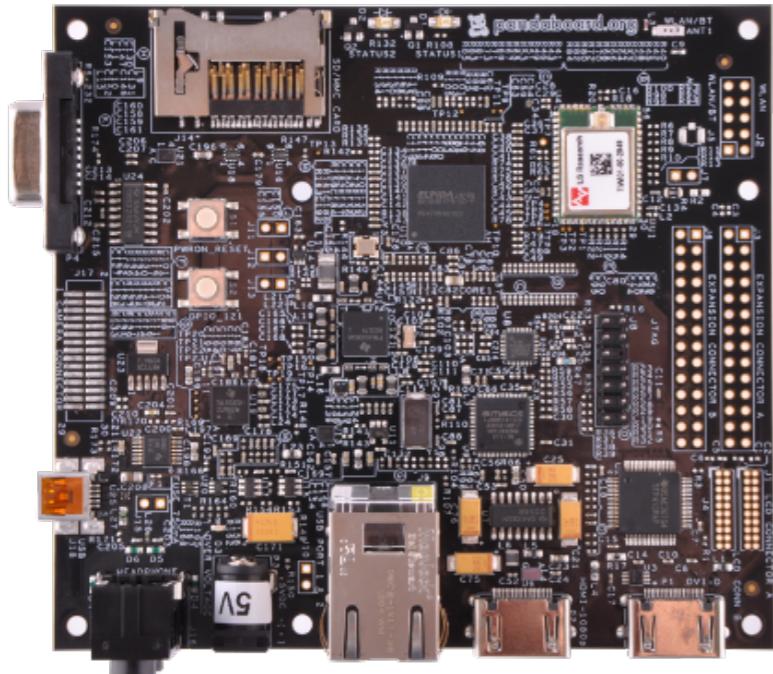
920 MHz, 1.11 V

1.2 GHz, 1.27 V

Built a current sensing circuitry to measure the voltage and energy of the SoC (isolate from other board peripherals)

< 3% run to run variation across 10 runs; use the median

Hardware Setup



Big core: ARM Cortex A9: OoO with 4 issue (e.g. in Tegra 3-based tablets)

PandaBoard ES Rev B1, 45nm

DVFS: 350 MHz, 0.83 V

700 MHz, 1.01 V

920 MHz, 1.11 V

1.2 GHz, 1.27 V

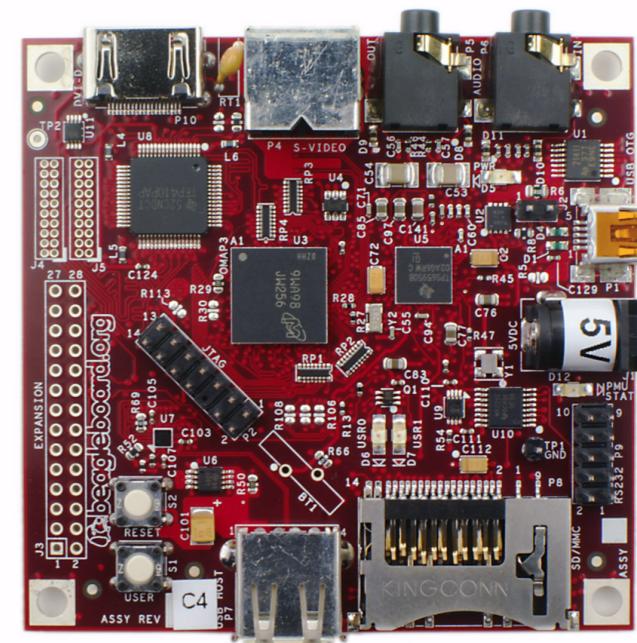
Little core: ARM Cortex A8: In-order with 2 issue (e.g. in Apple A4 -- iPhone 4)

BeagleBoard xM, 45 nm

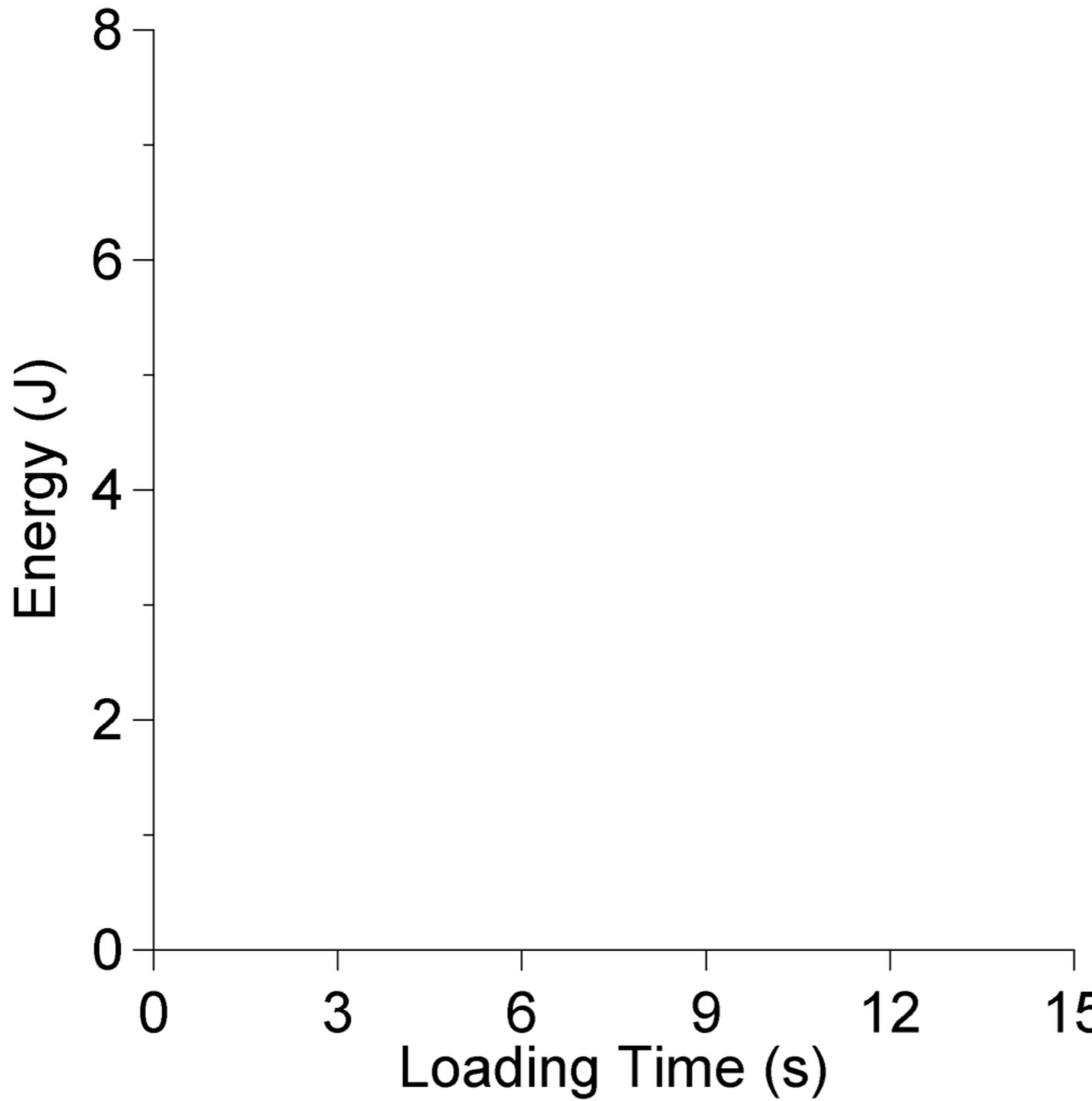
DVFS: 300 MHz, 0.94 V

600 MHz, 1.10 V

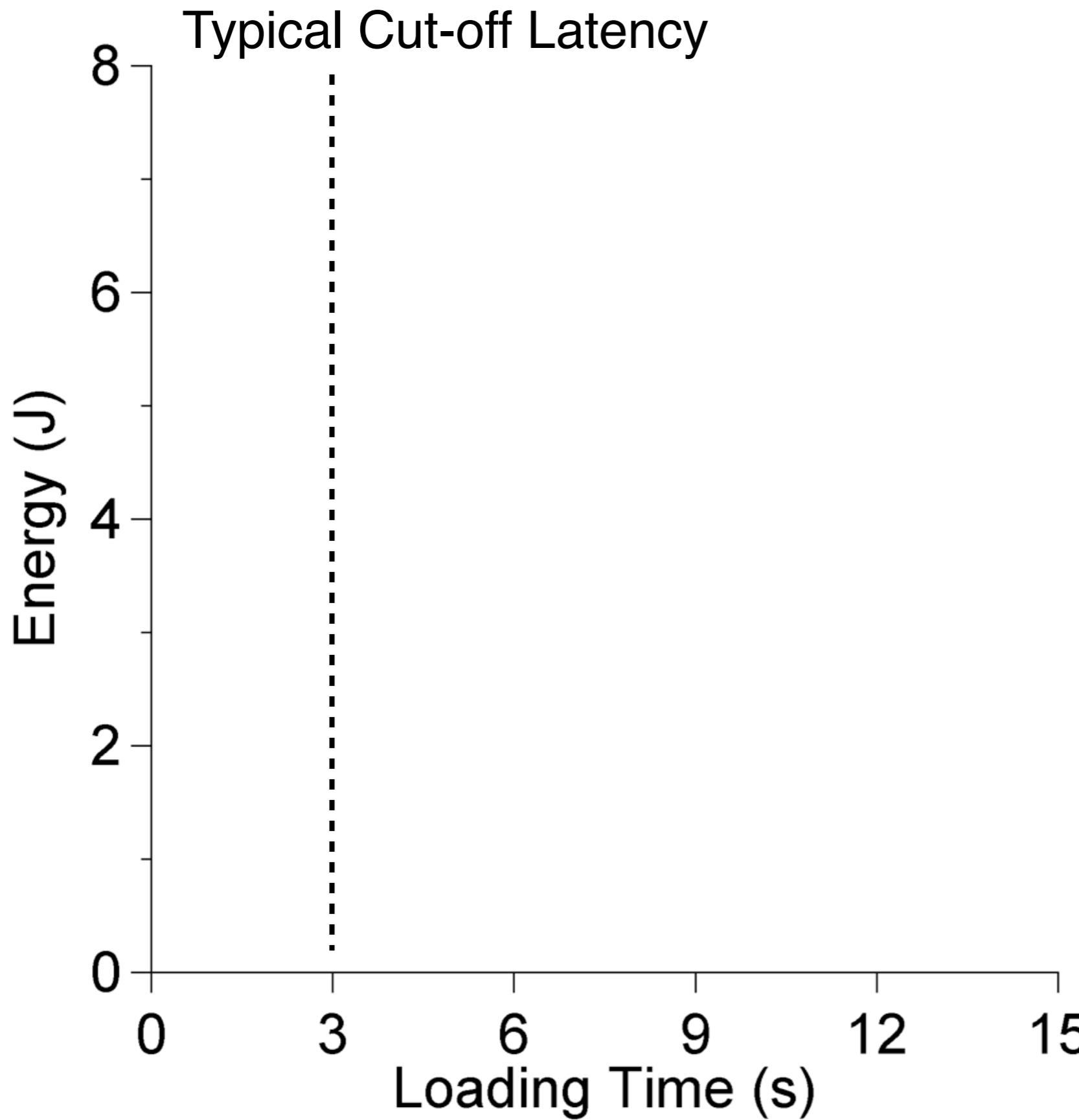
800 MHz, 1.26 V



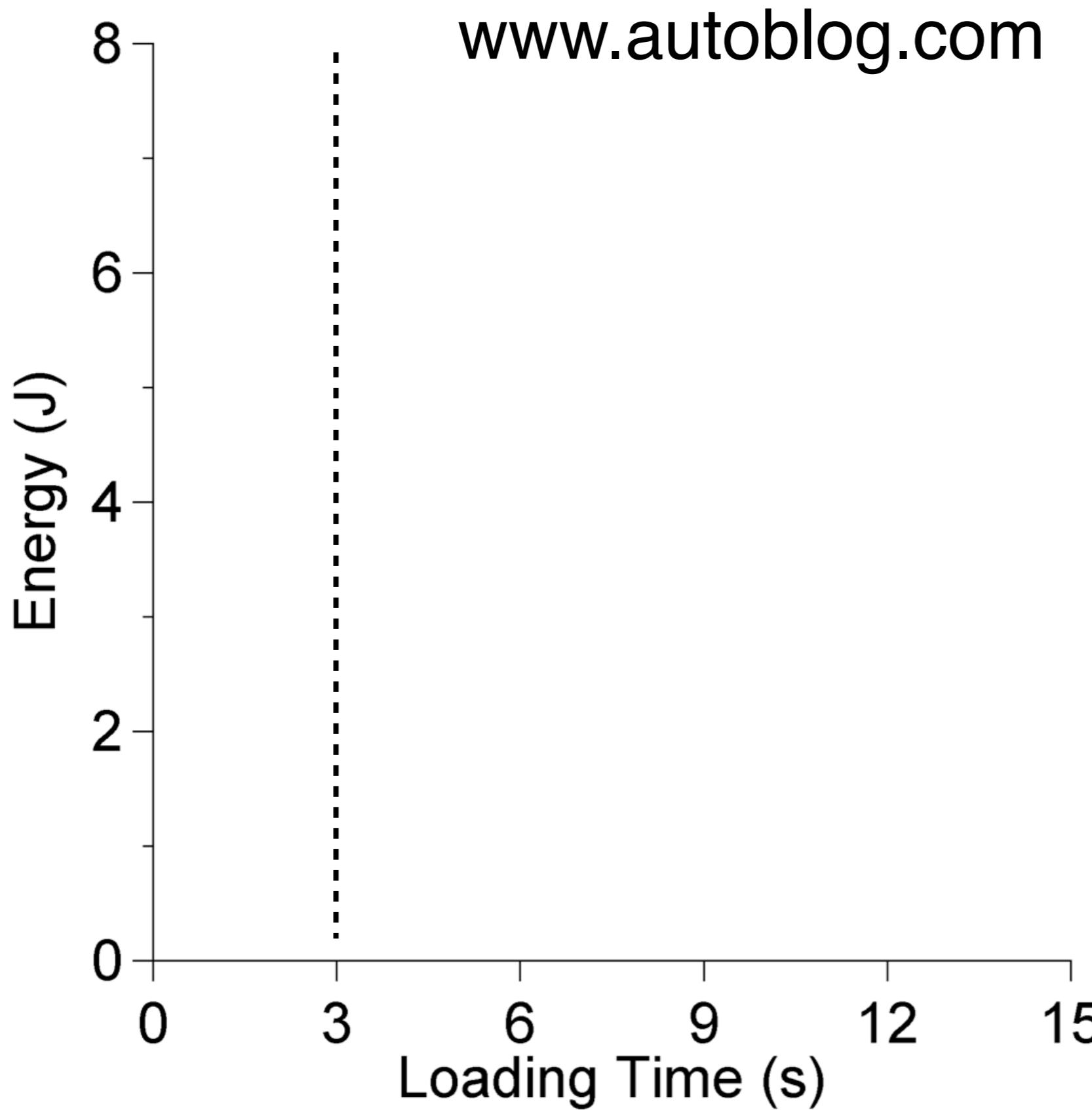
Why Big/Little Systems?



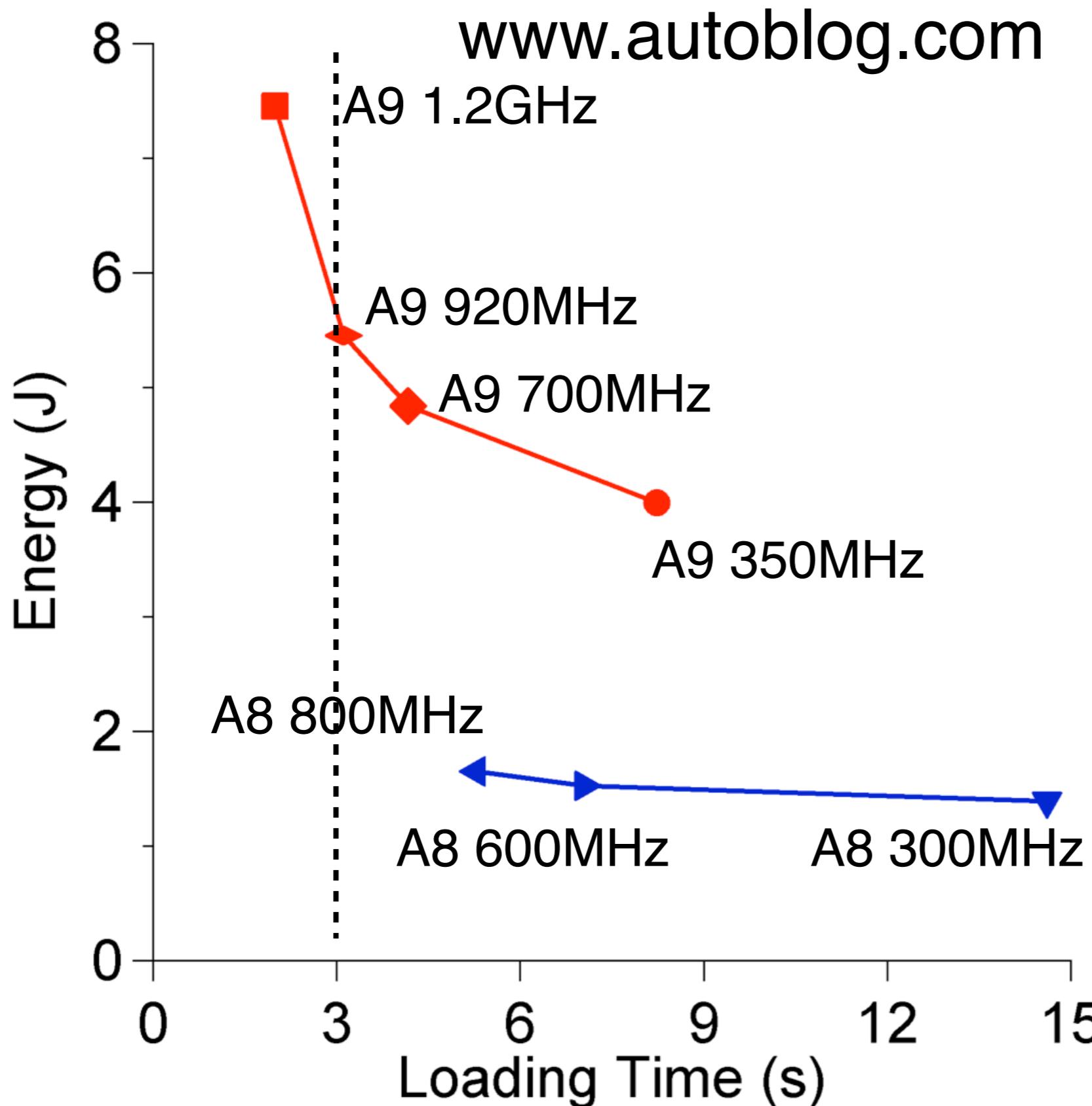
Why Big/Little Systems?



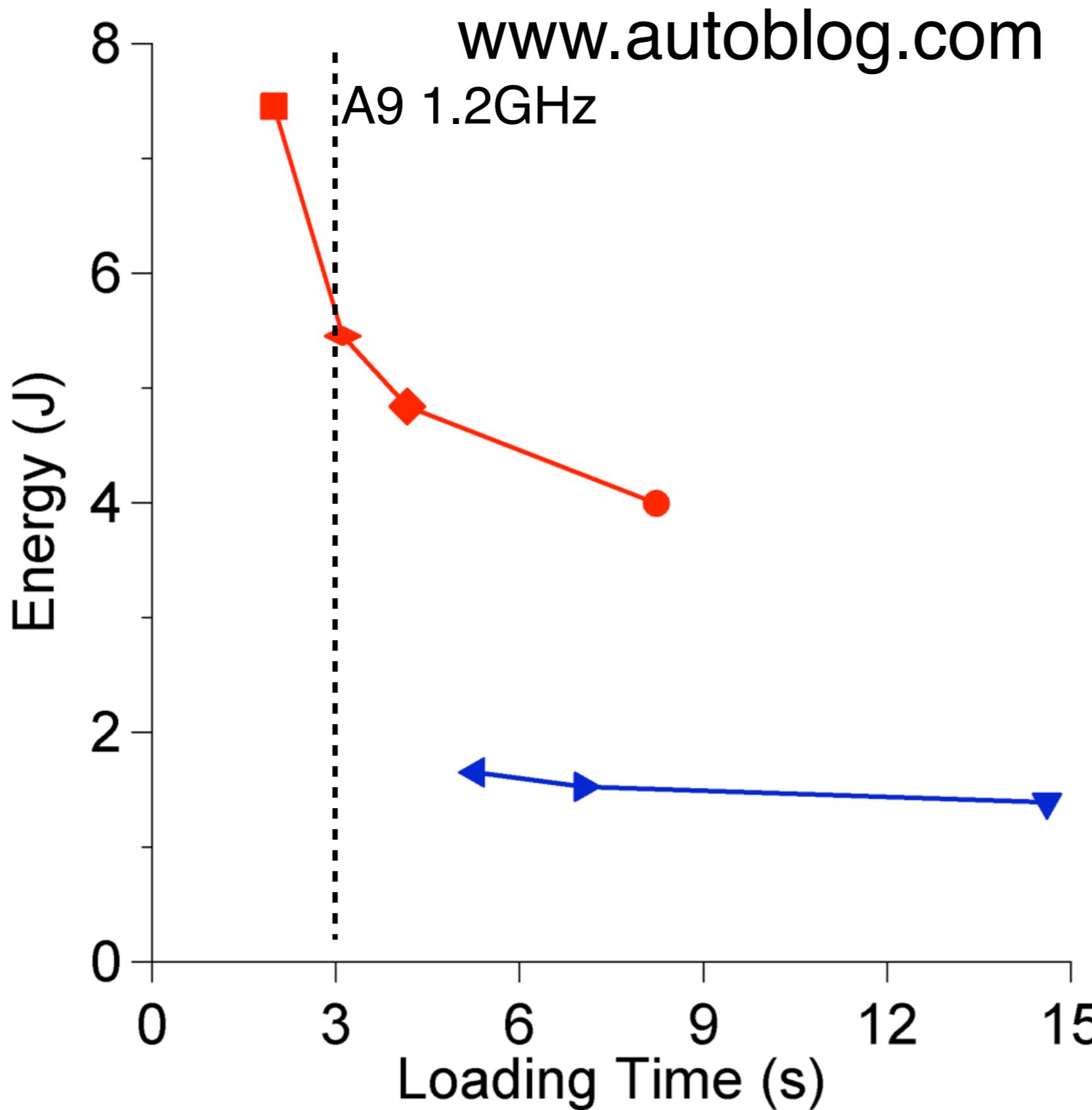
Why Big/Little Systems?



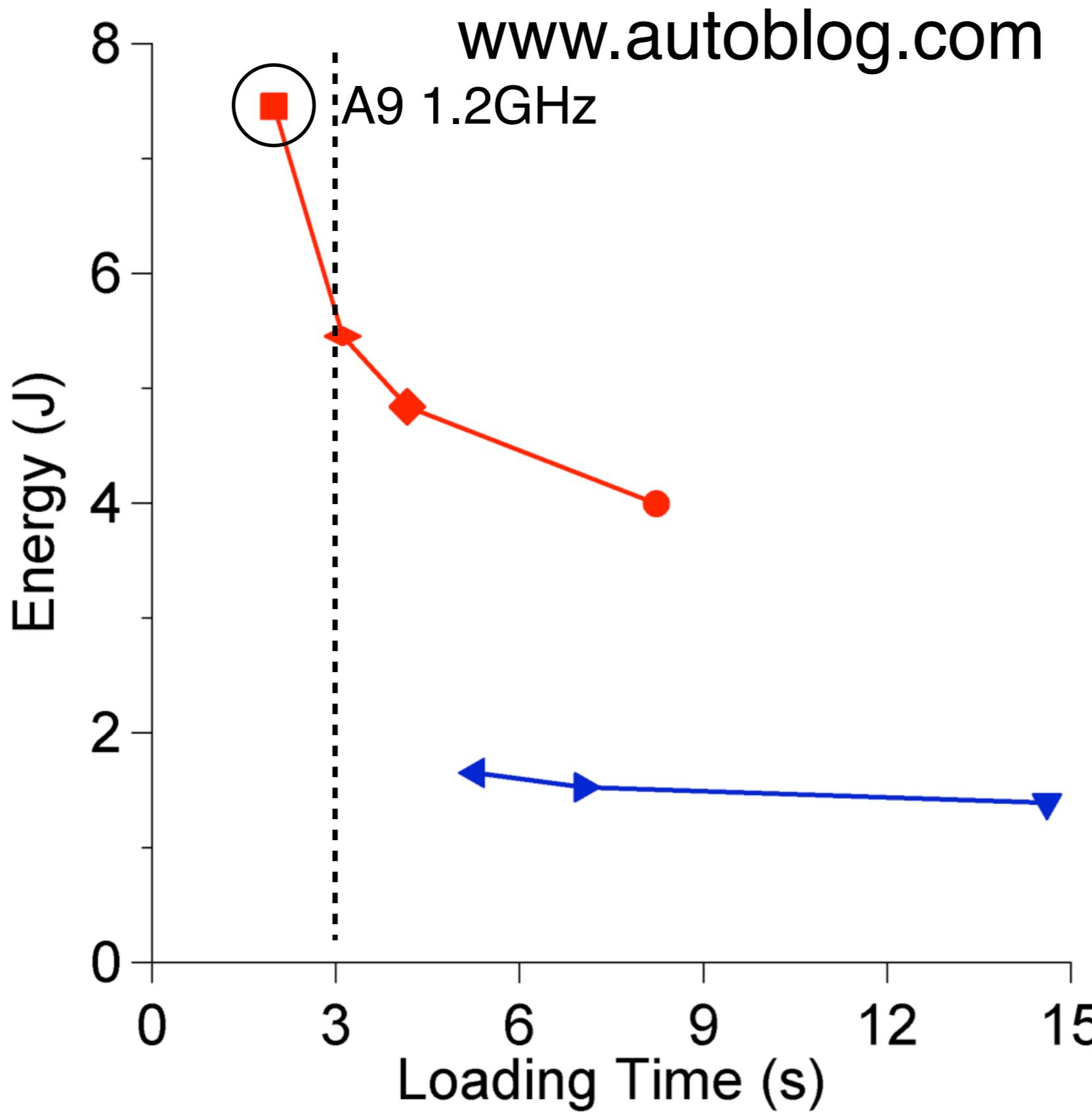
Why Big/Little Systems?



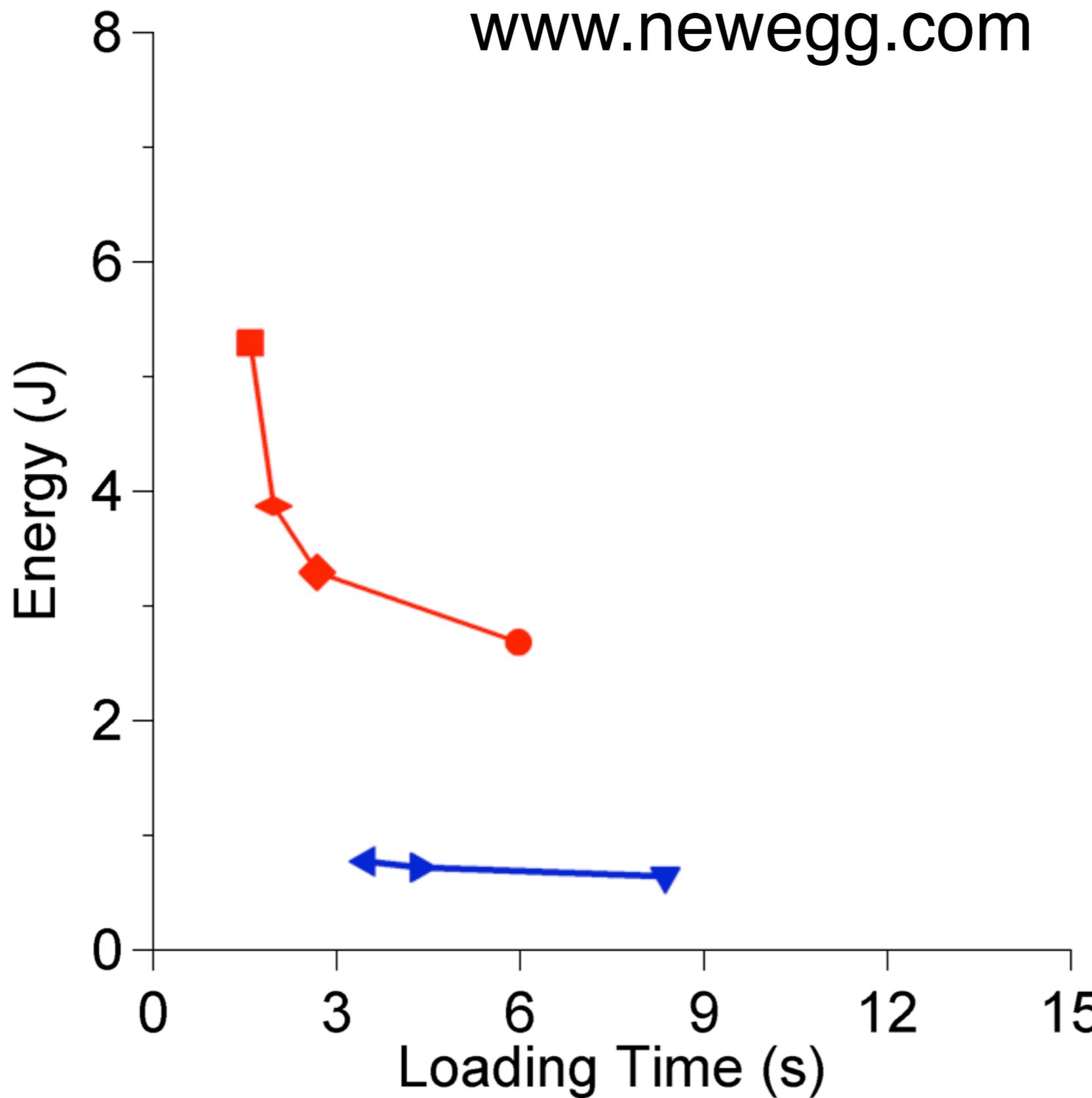
Why Big/Little Systems?



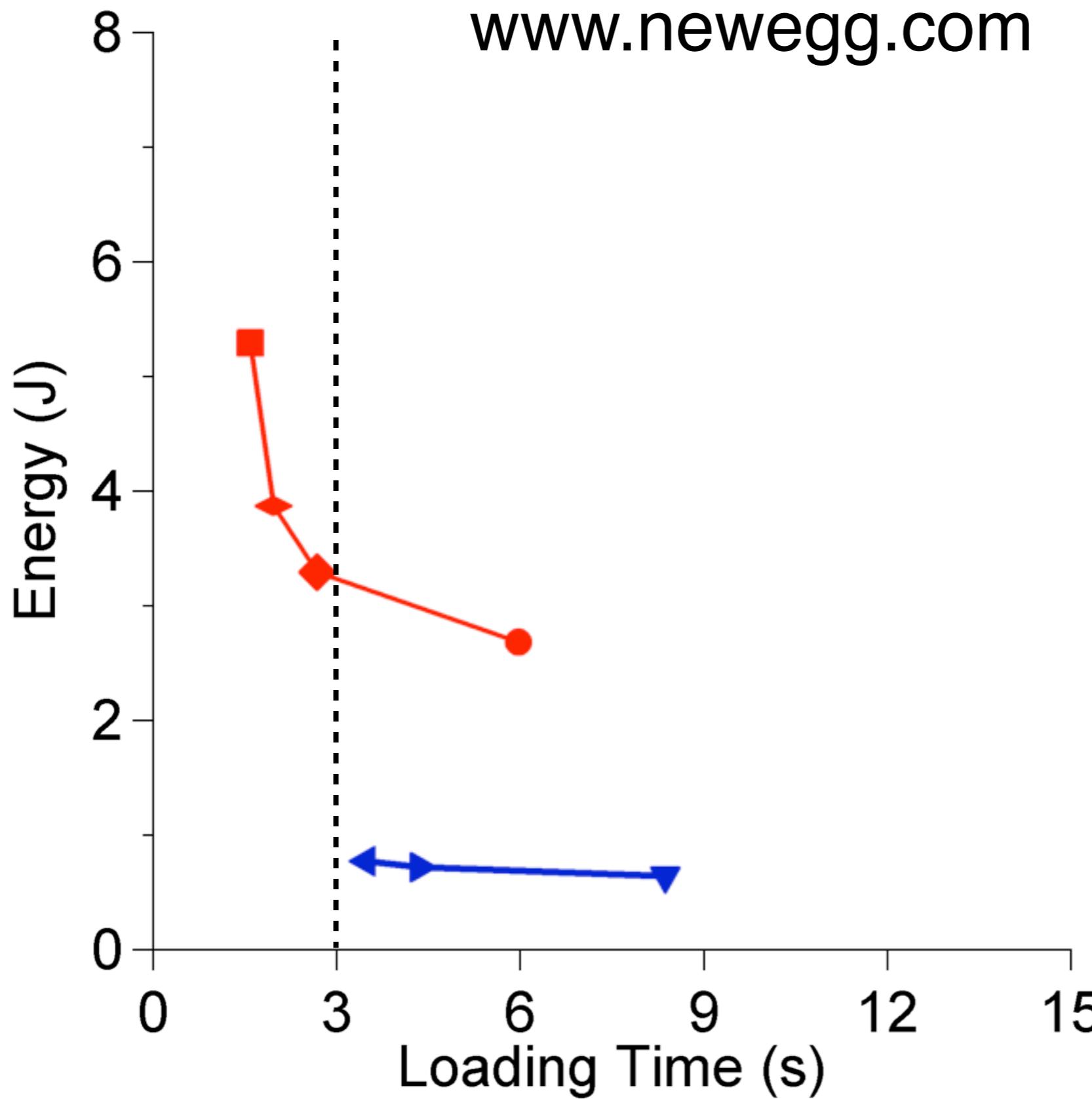
Why Big/Little Systems?



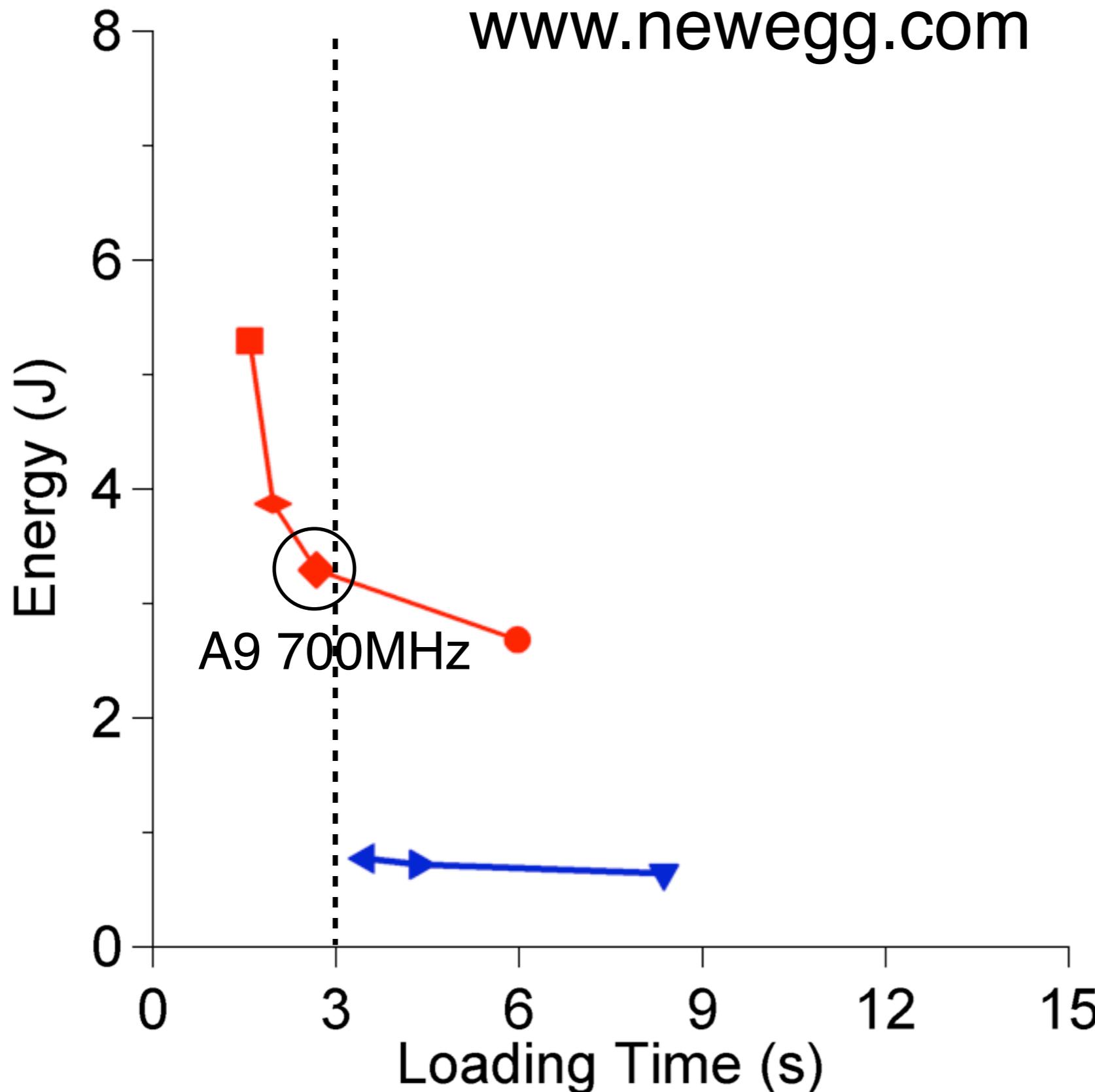
Why Big/Little Systems?



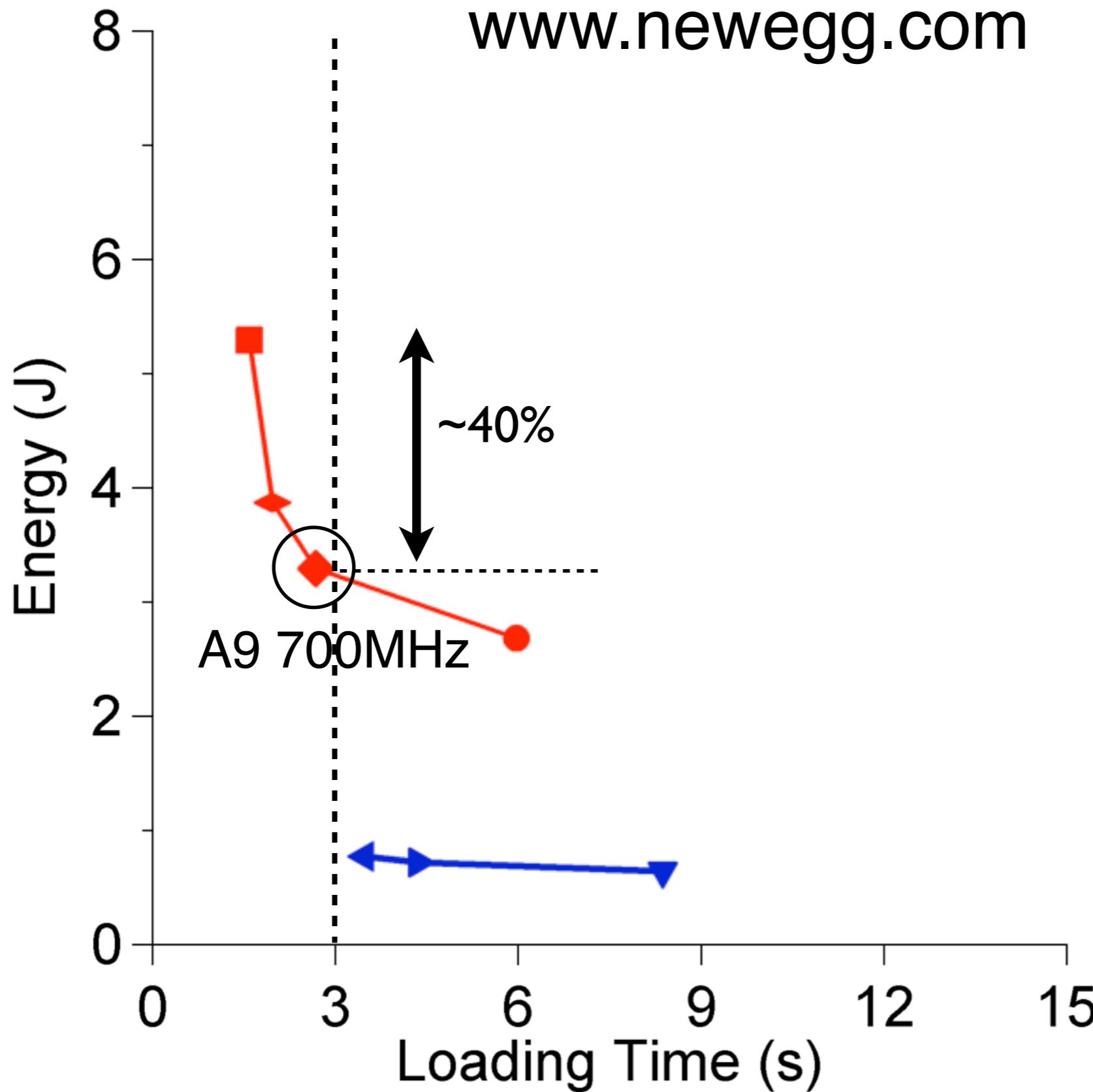
Why Big/Little Systems?



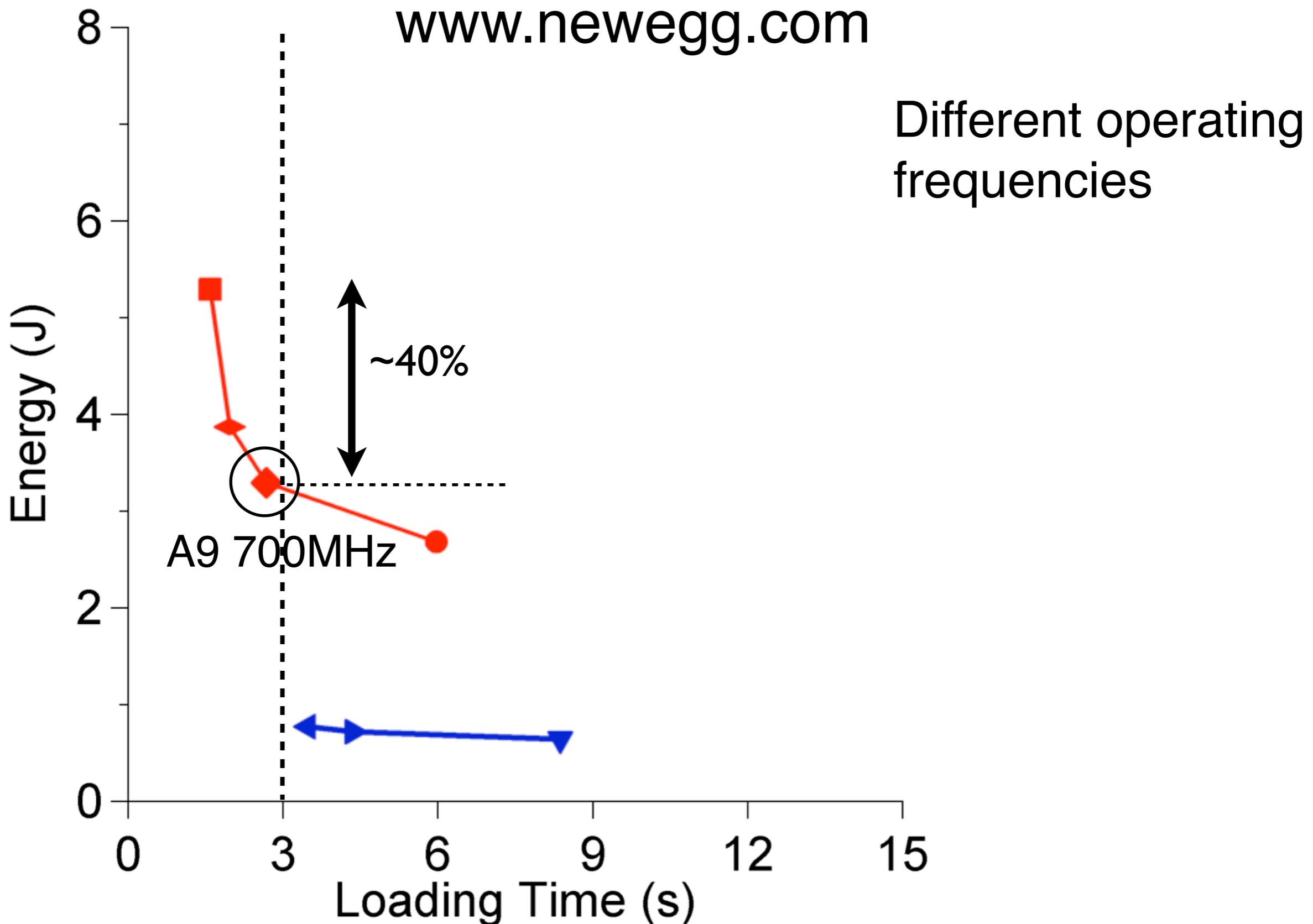
Why Big/Little Systems?



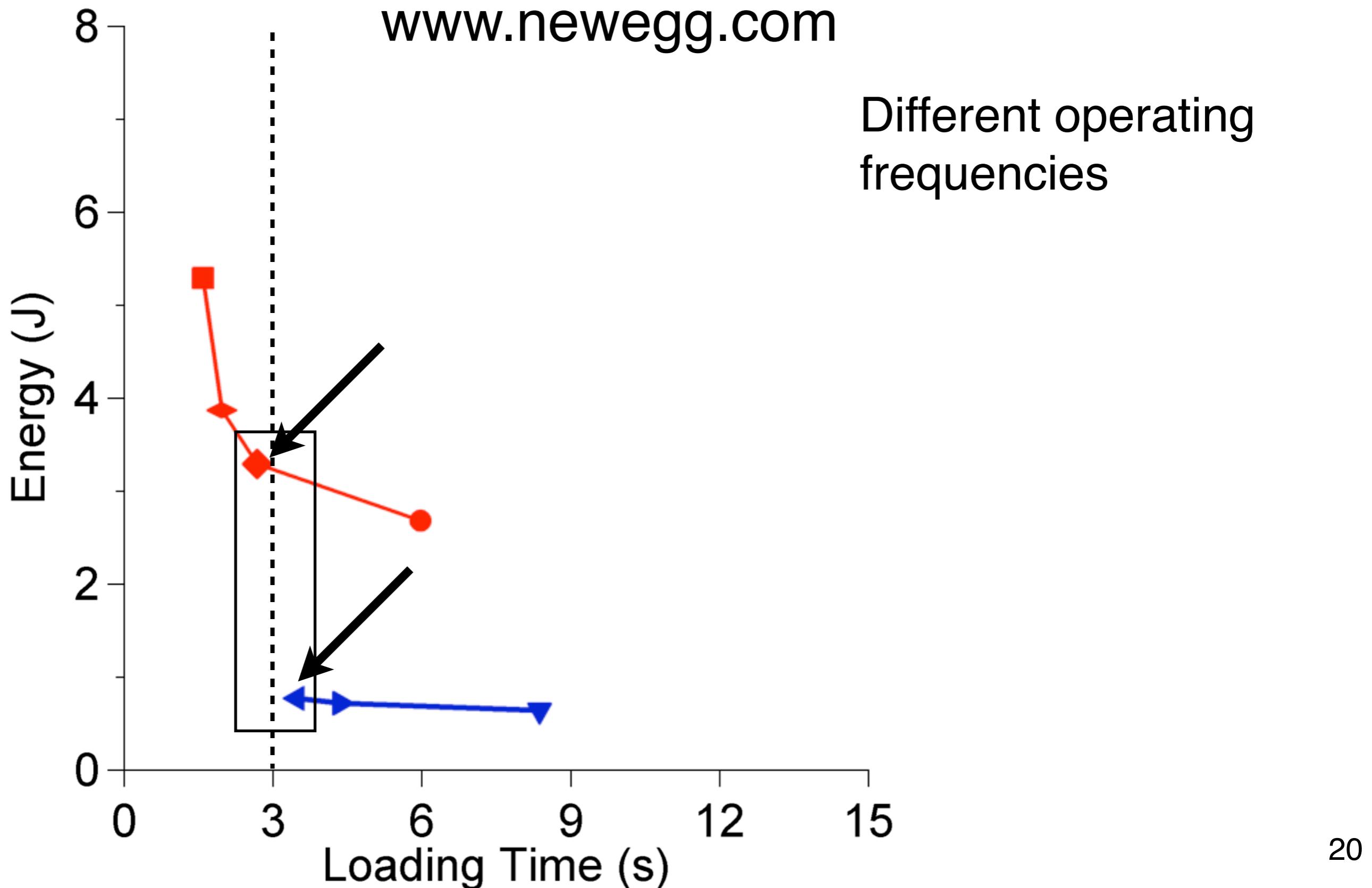
Why Big/Little Systems?



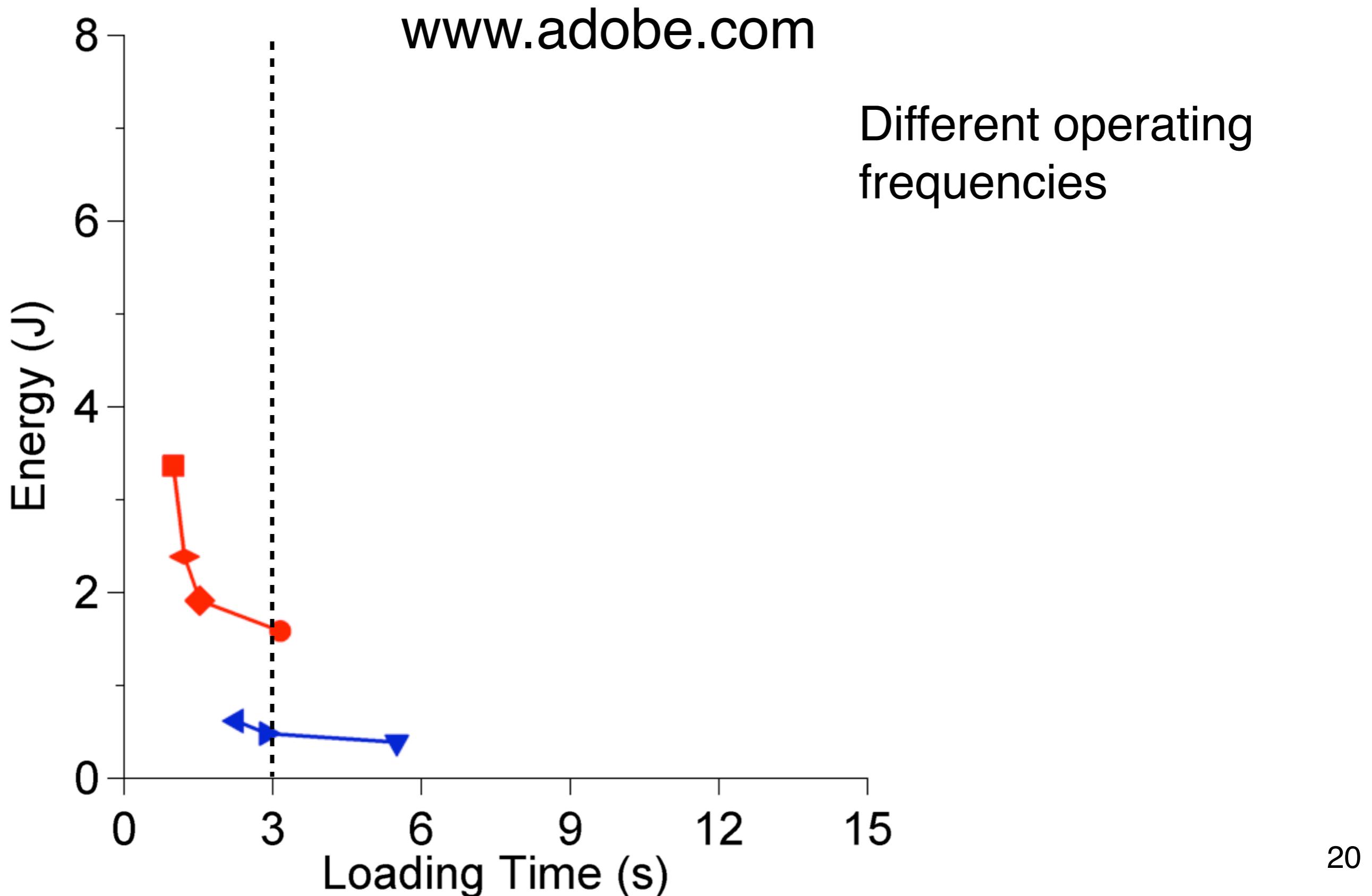
Why Big/Little Systems?



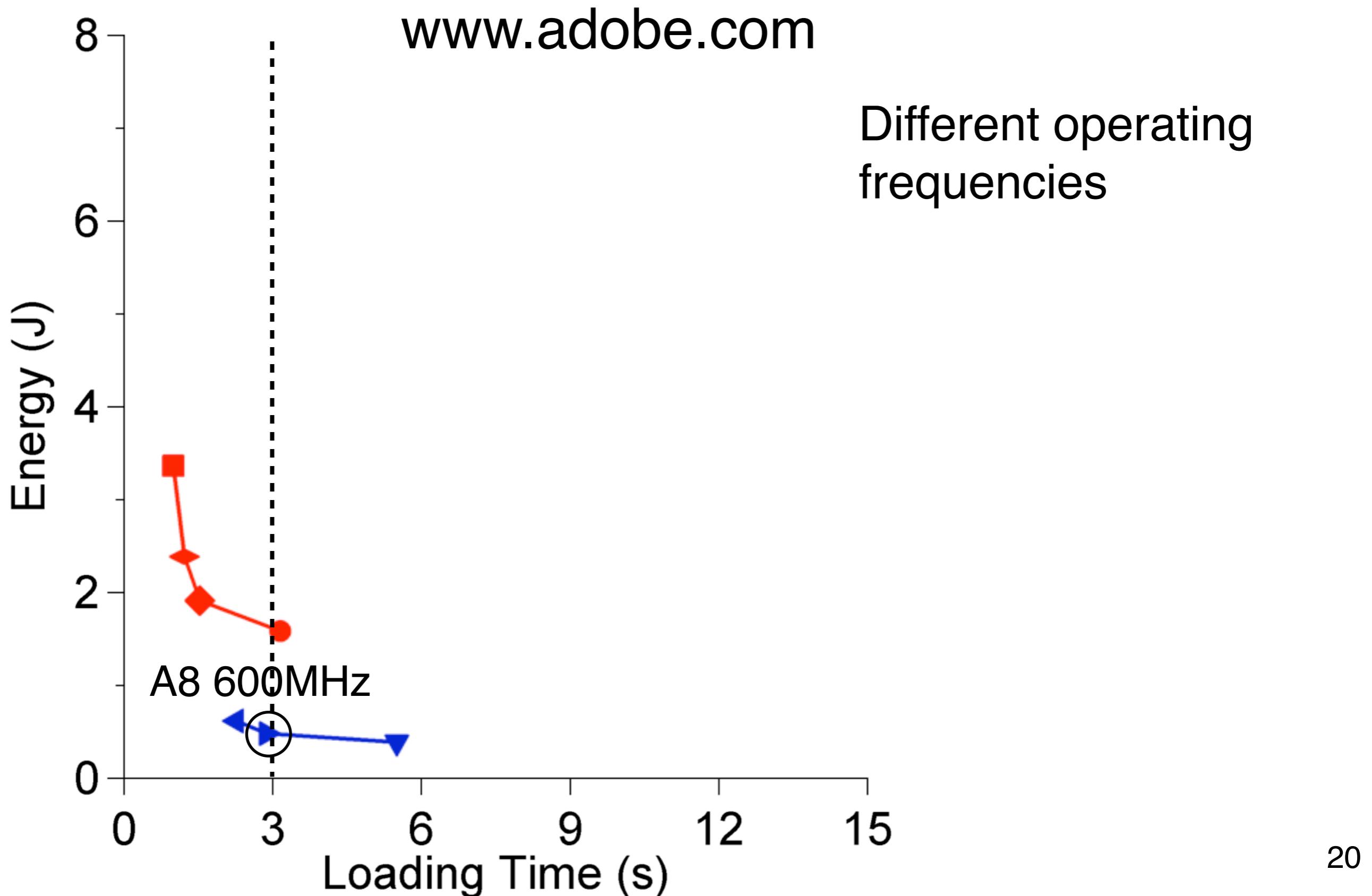
Why Big/Little Systems?



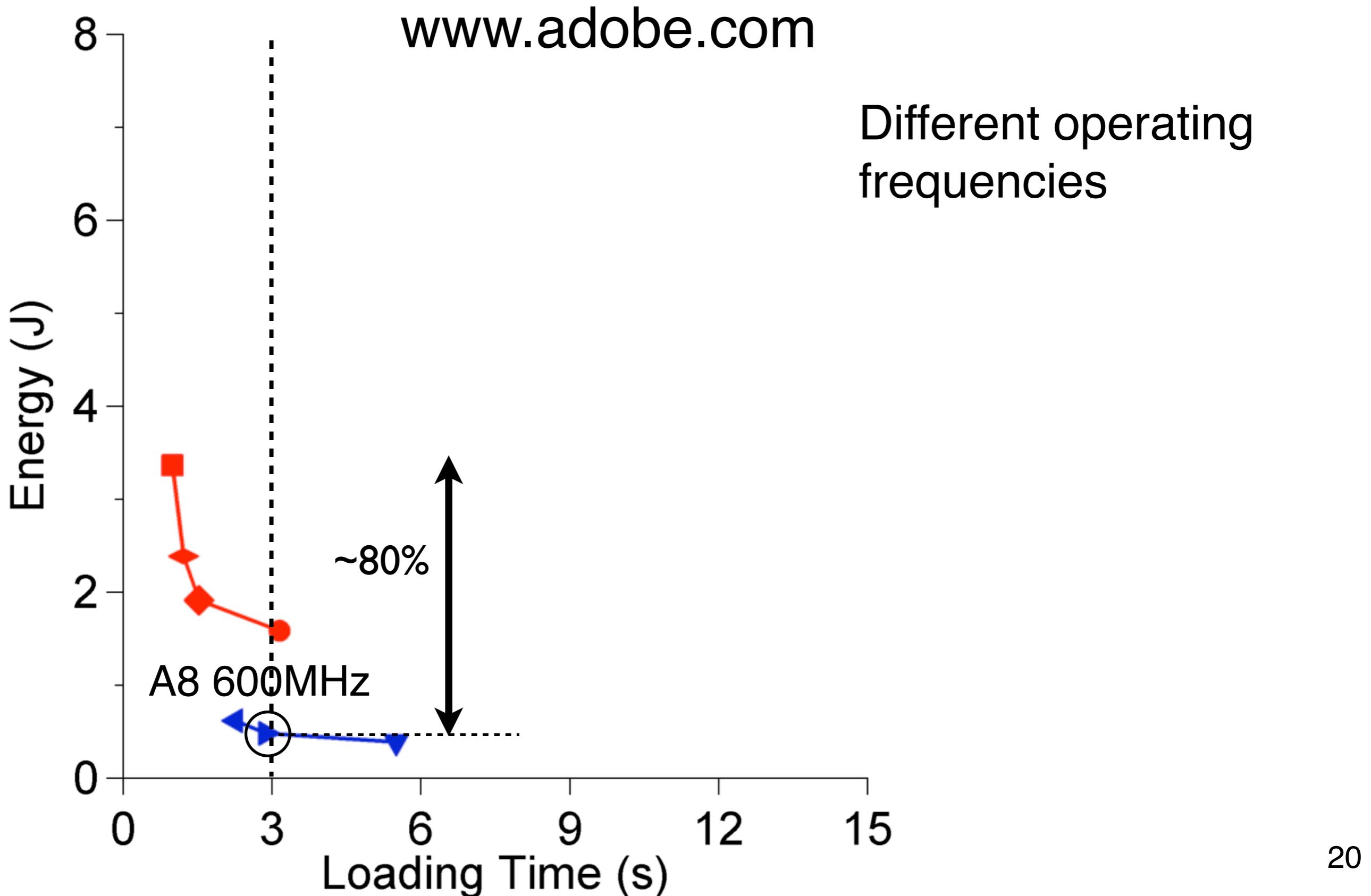
Why Big/Little Systems?



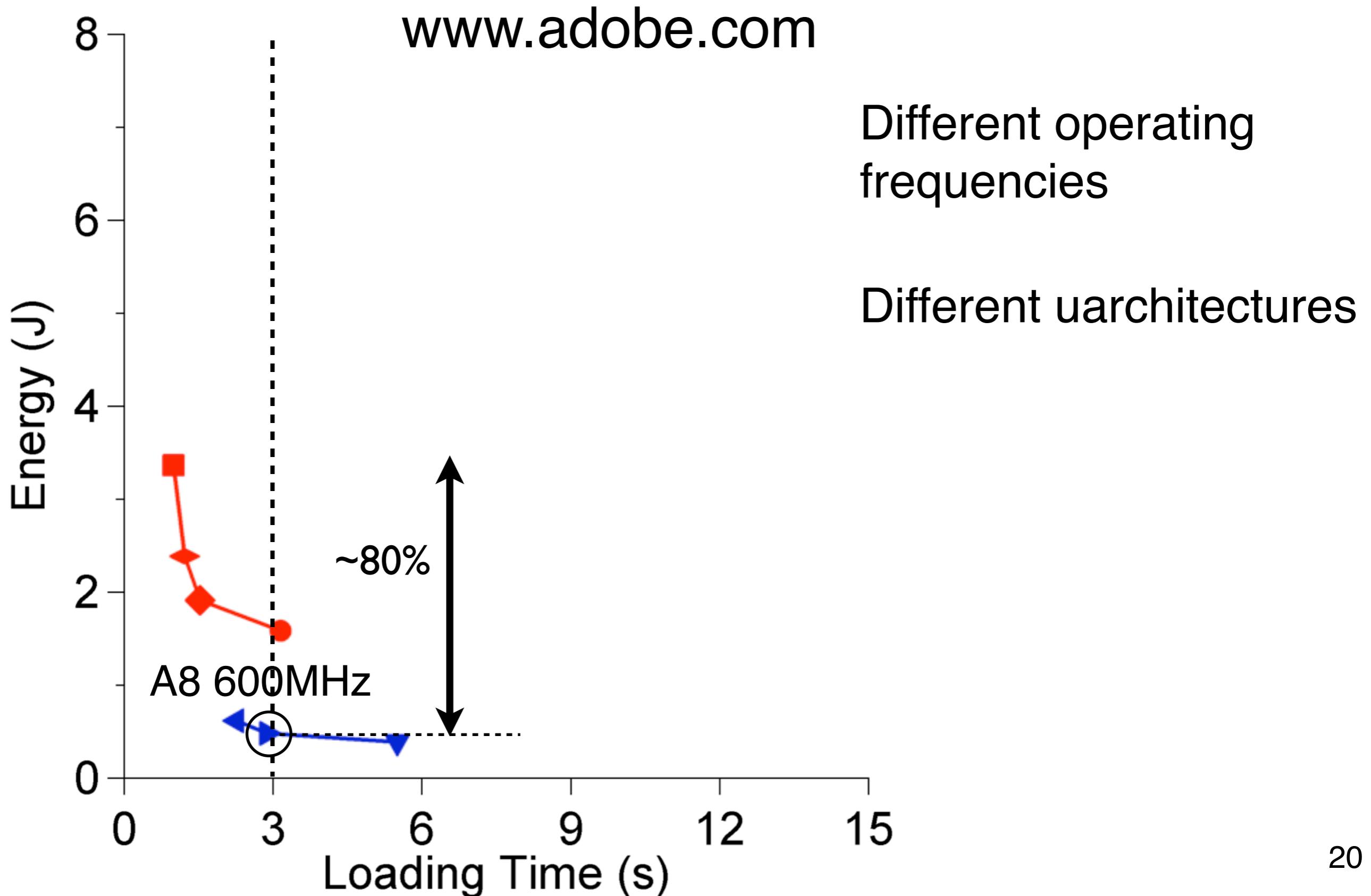
Why Big/Little Systems?



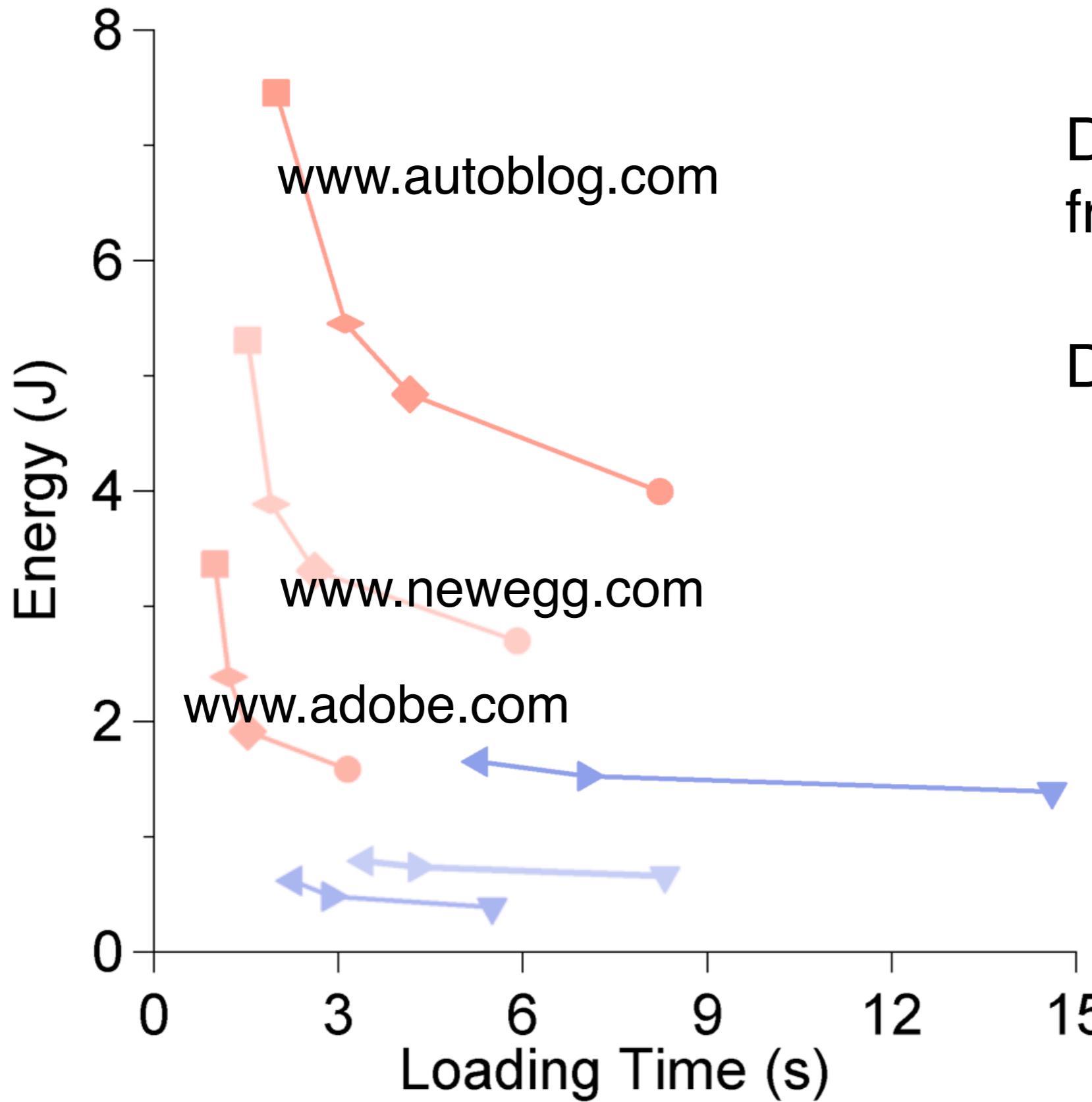
Why Big/Little Systems?



Why Big/Little Systems?



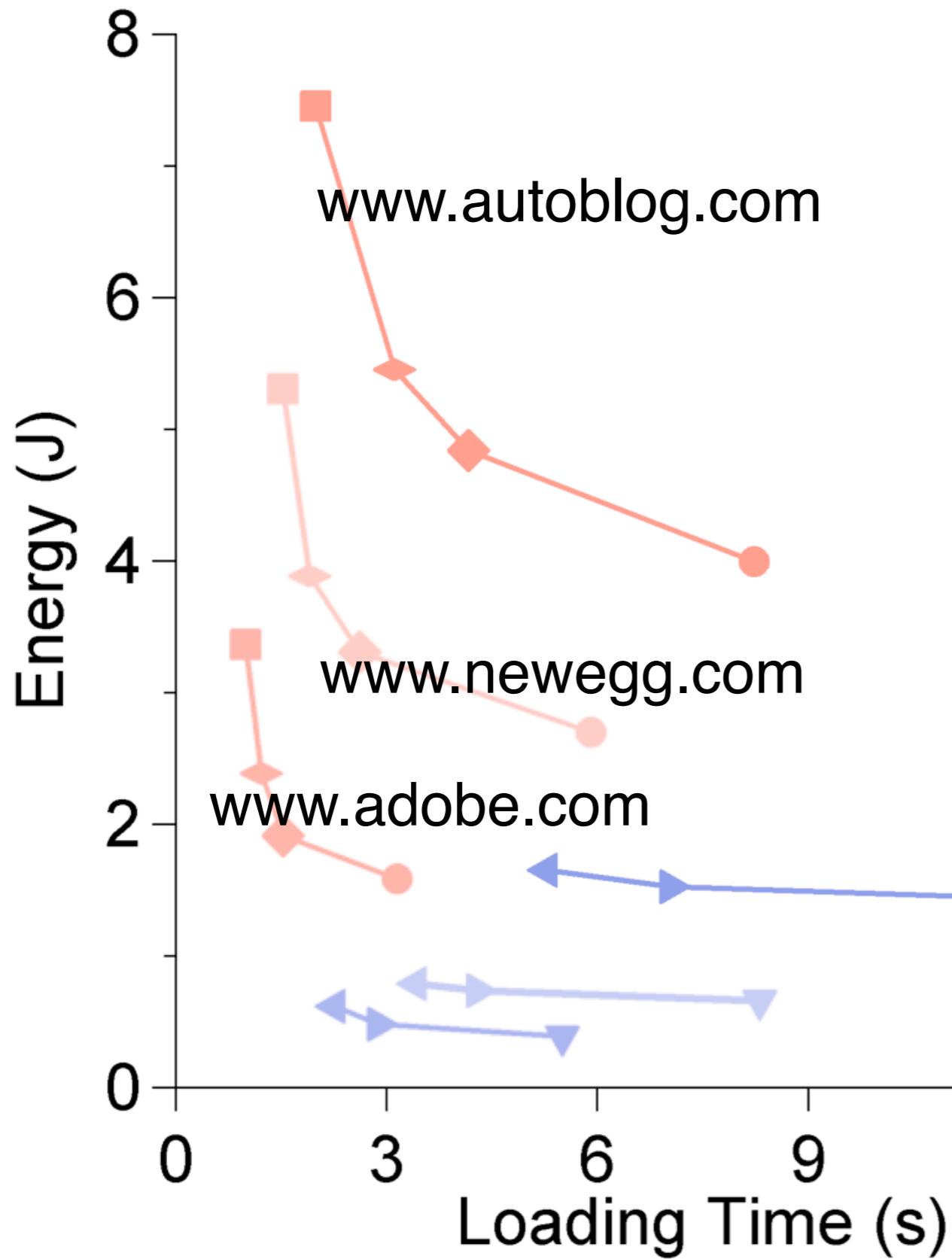
Why Big/Little Systems?



Different operating frequencies

Different uarchitectures

Why Big/Little Systems?



Different operating frequencies

Different uarchitectures

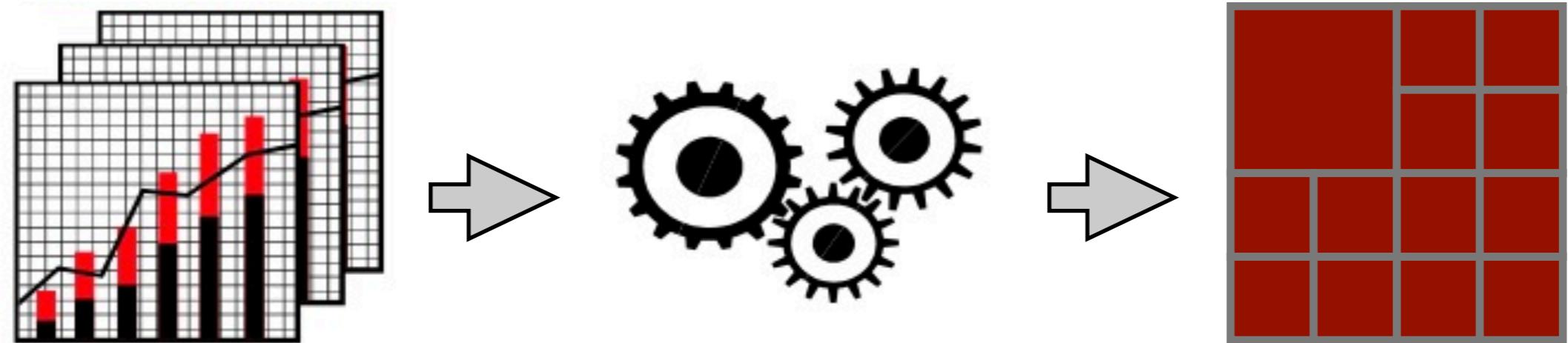
“Webpages variance” in load time and energy

Our Approach

How to leverage the big/little system to capture the webpage variance?

Our Approach

How to leverage the big/little system to capture the webpage variance?



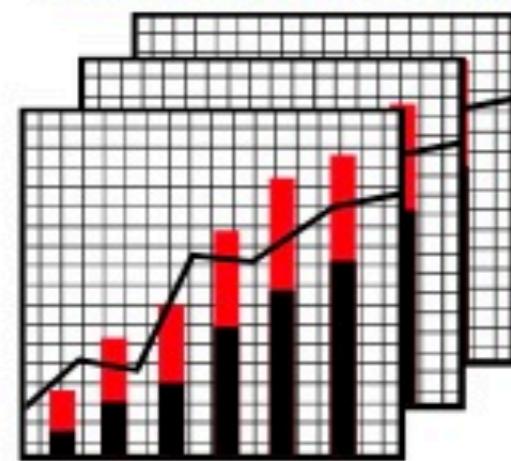
Workload
Characterization

Performance/Energy
Prediction

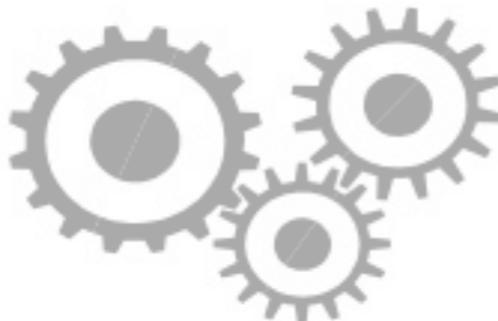
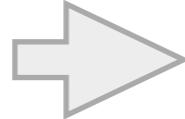
Resource
Management

Our Approach

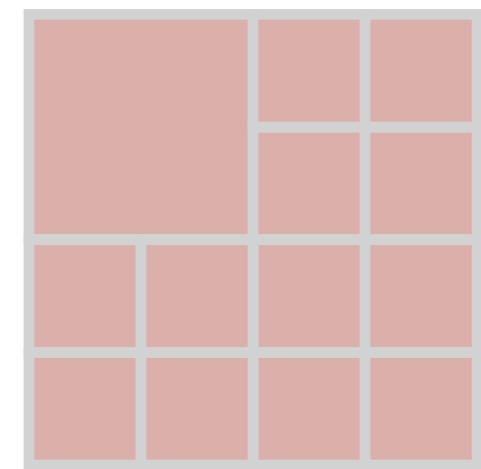
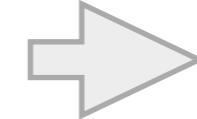
How to capture the dynamic behavior of web browsing?



**Workload
Characterization**

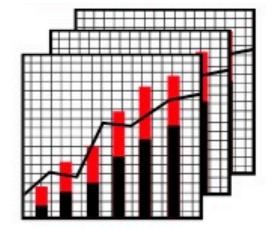


Performance/Energy
Prediction



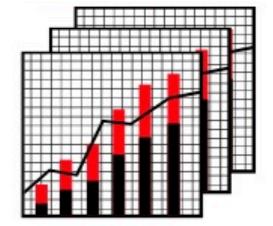
Resource
Management

Webpage Characterization



We treat webpages as the workload instead of the browser!

Webpage Characterization

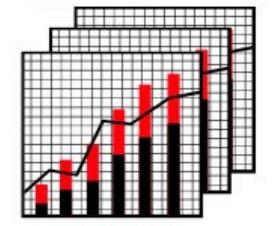


We treat webpages as the workload instead of the browser!

HTML (Structure)

CSS (Style)

Webpage Characterization



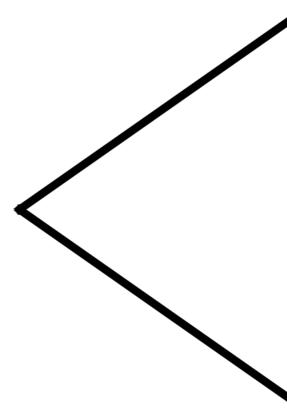
We treat webpages as the workload instead of the browser!

HTML (Structure)

CSS (Style)

Tag (h3, li, table, img)

Attribute



Webpage Characterization



We treat webpages as the workload instead of the browser!

HTML (Structure)

Tag (h3, li, table, img)

Attribute

CSS (Style)

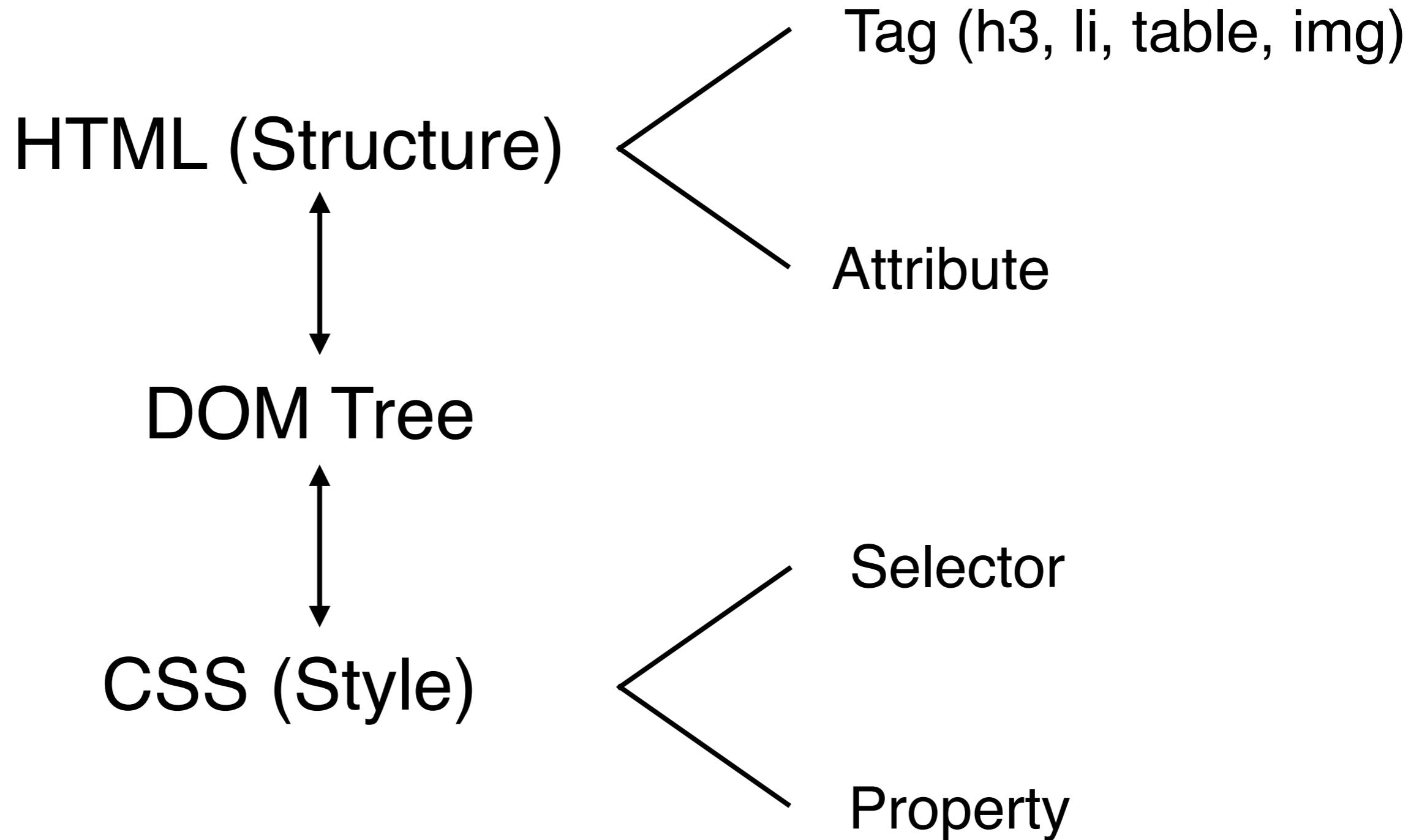
Selector

Property

Webpage Characterization



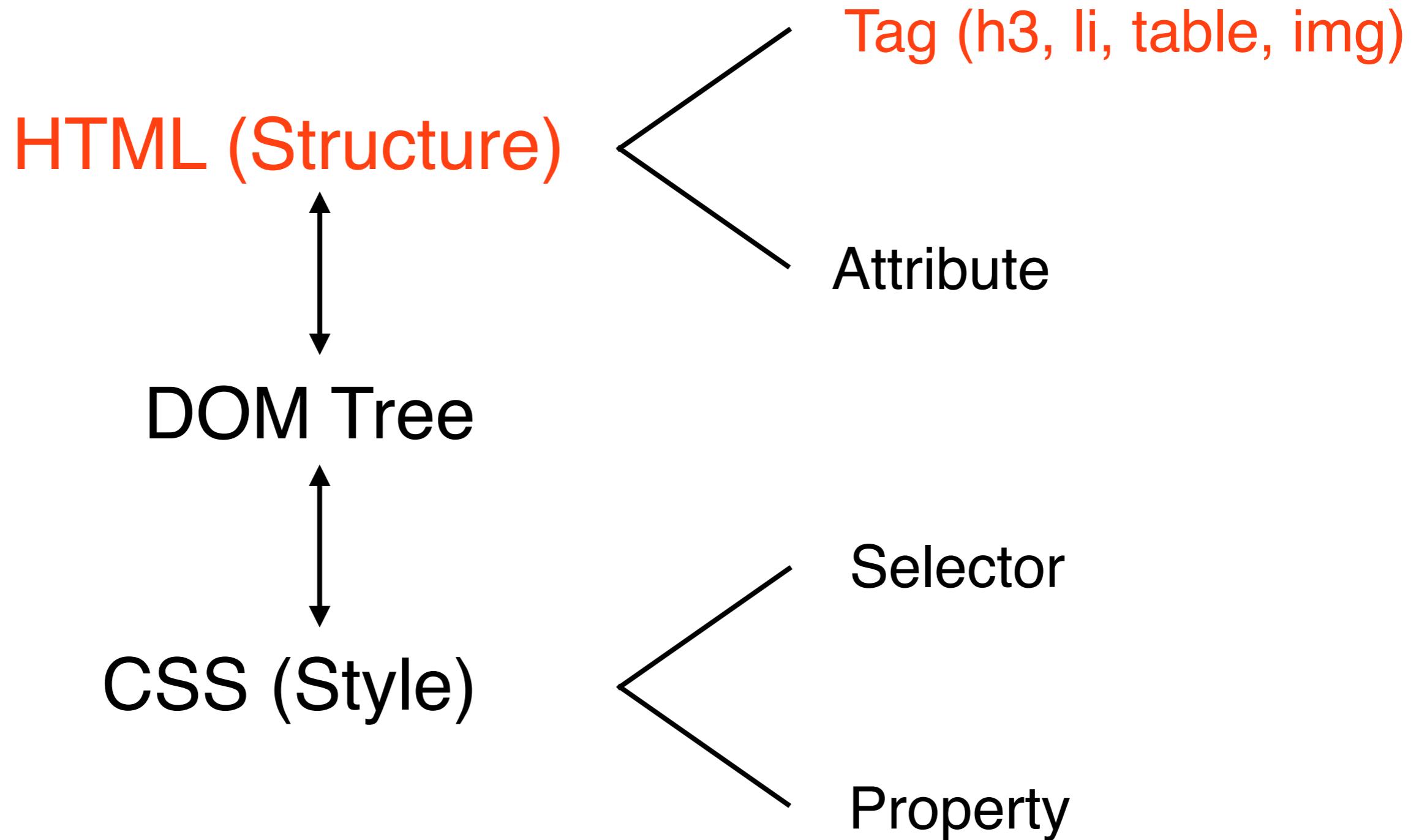
We treat webpages as the workload instead of the browser!



Webpage Characterization



We treat webpages as the workload instead of the browser!



HTML Tag Analysis



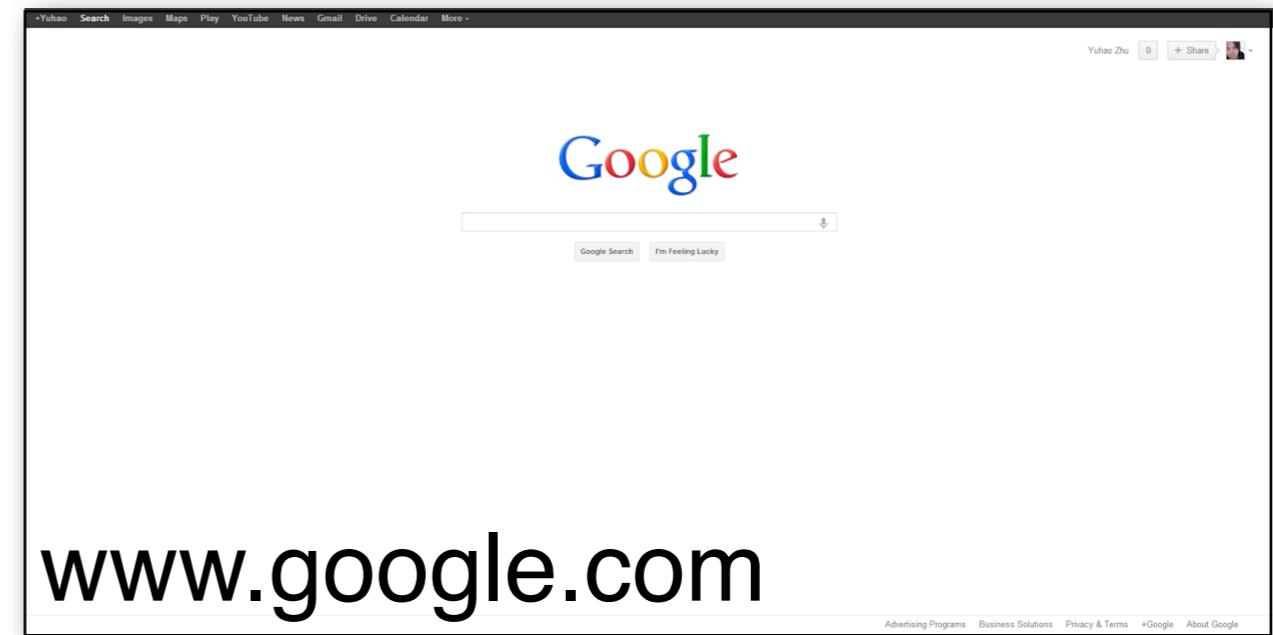
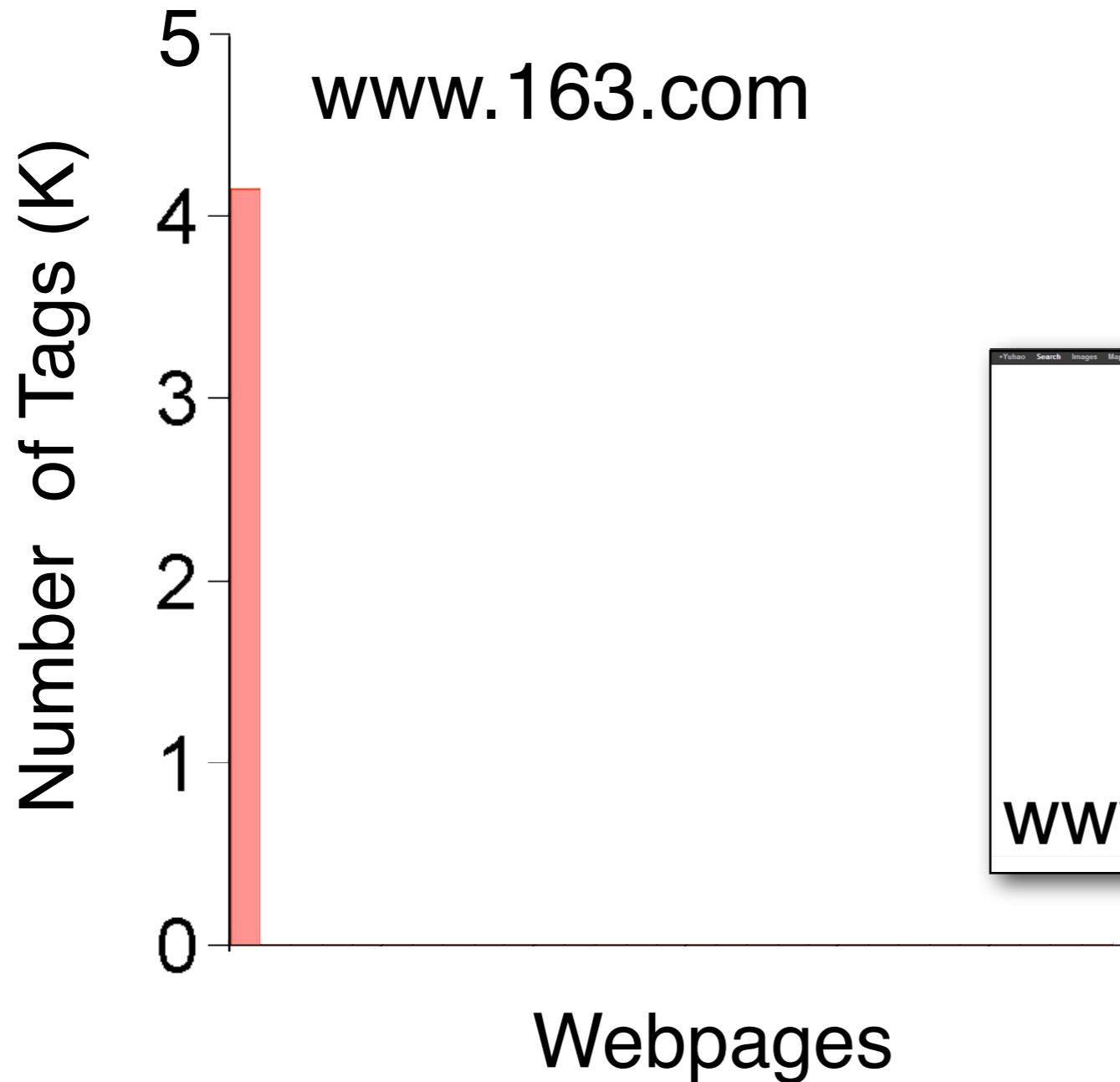
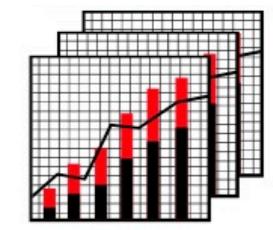
www.163.com

This image is a collage of numerous screenshots from Chinese news websites and forums, primarily from the 2013 Spring Festival period. It includes content from major platforms like NetEase, Sina, and China.org.cn. The screenshots feature various news stories, including reports on the 2013 CCTV Spring Festival Gala, host selection, and related events. The collage is dense with text, images, and video thumbnails, illustrating the extensive coverage of the topic across different media outlets.

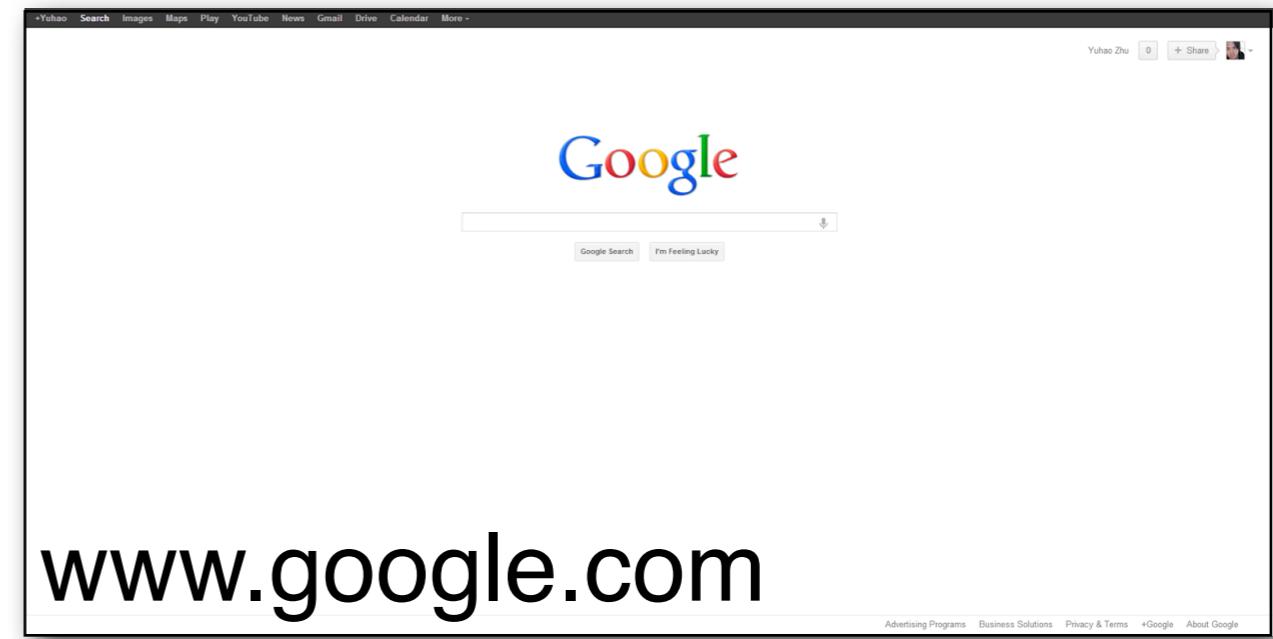
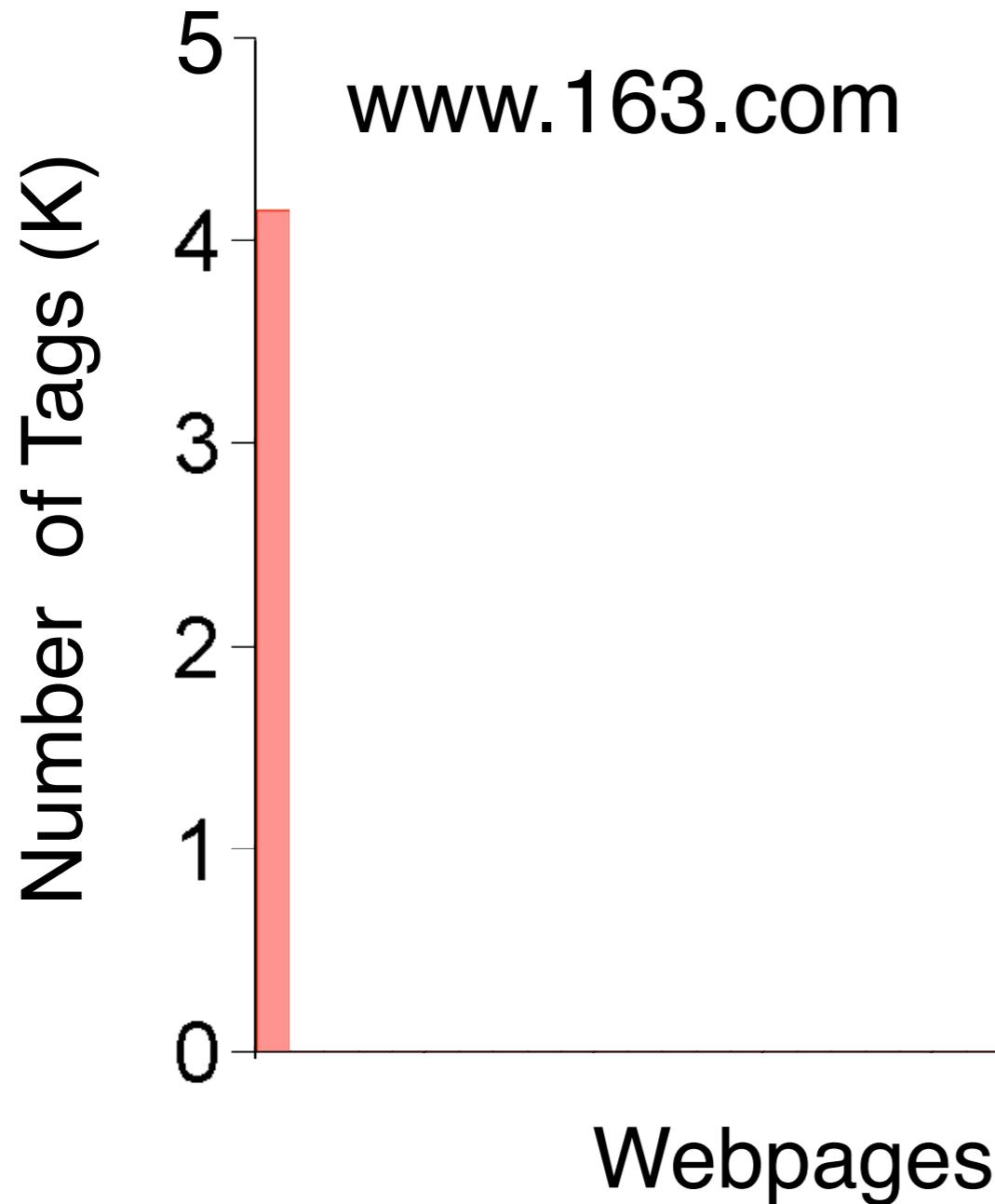
HTML Tag Analysis



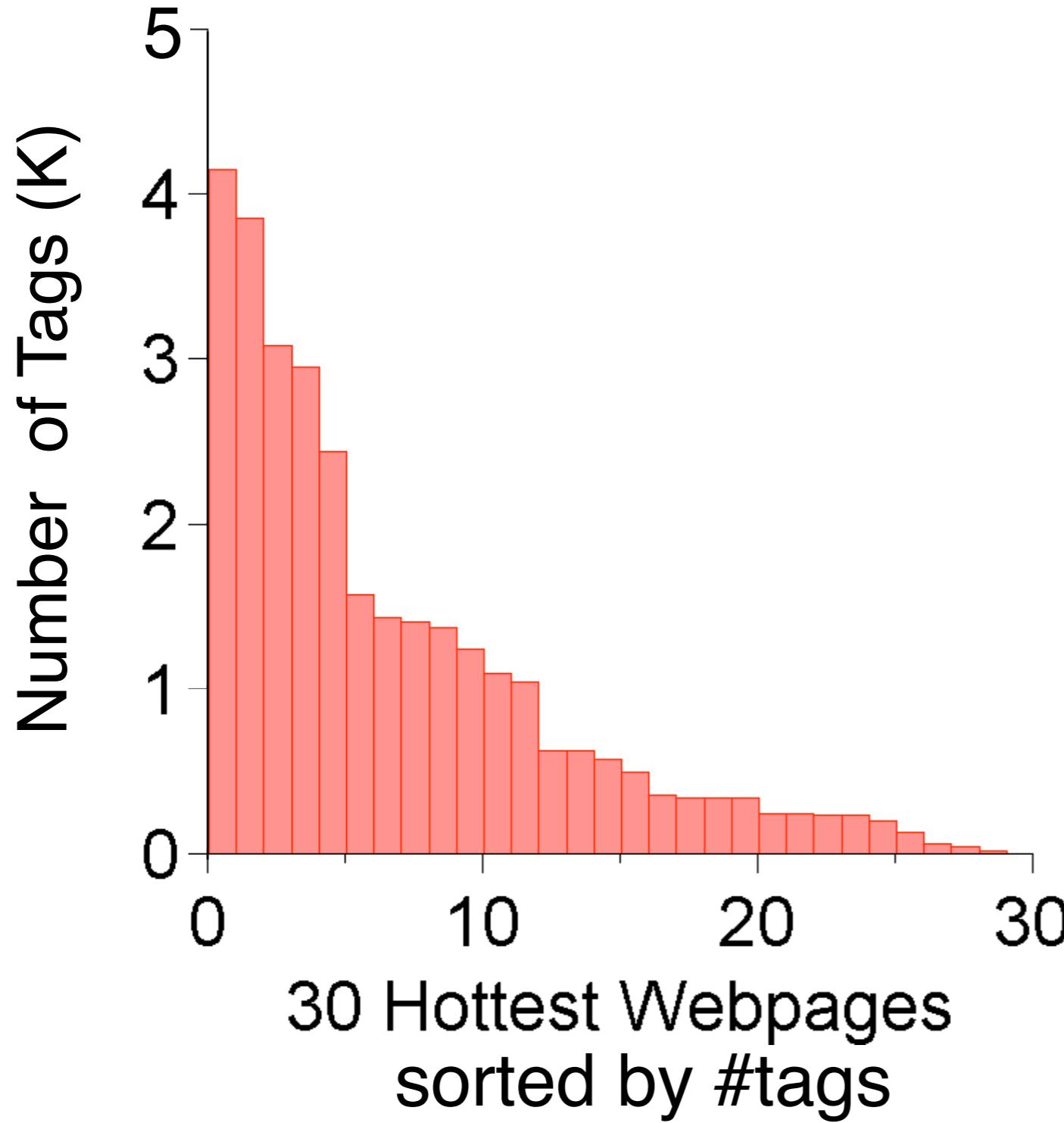
HTML Tag Analysis



HTML Tag Analysis

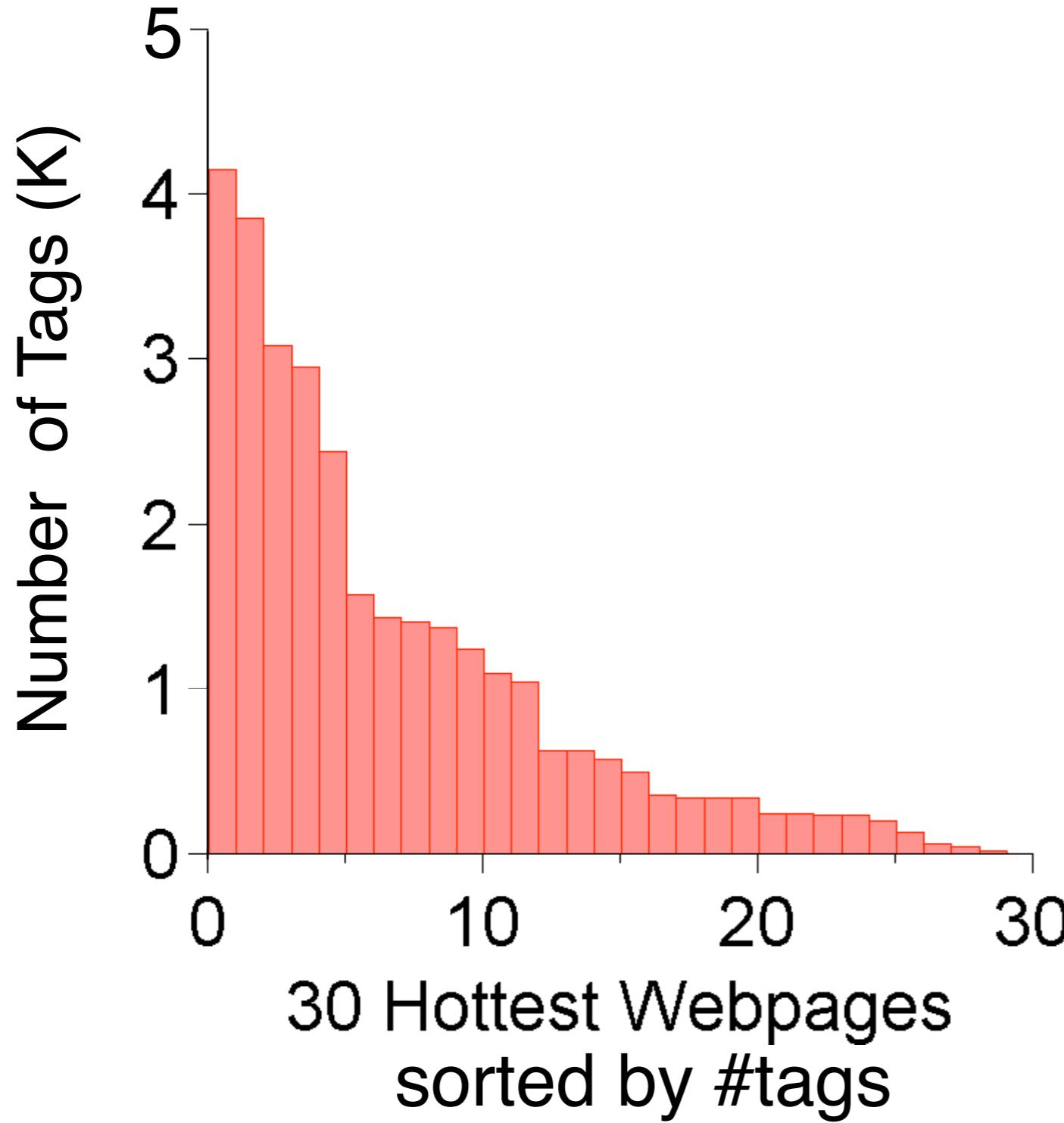


HTML Tag Analysis



Webpages have
different tag counts
(instruction counts)

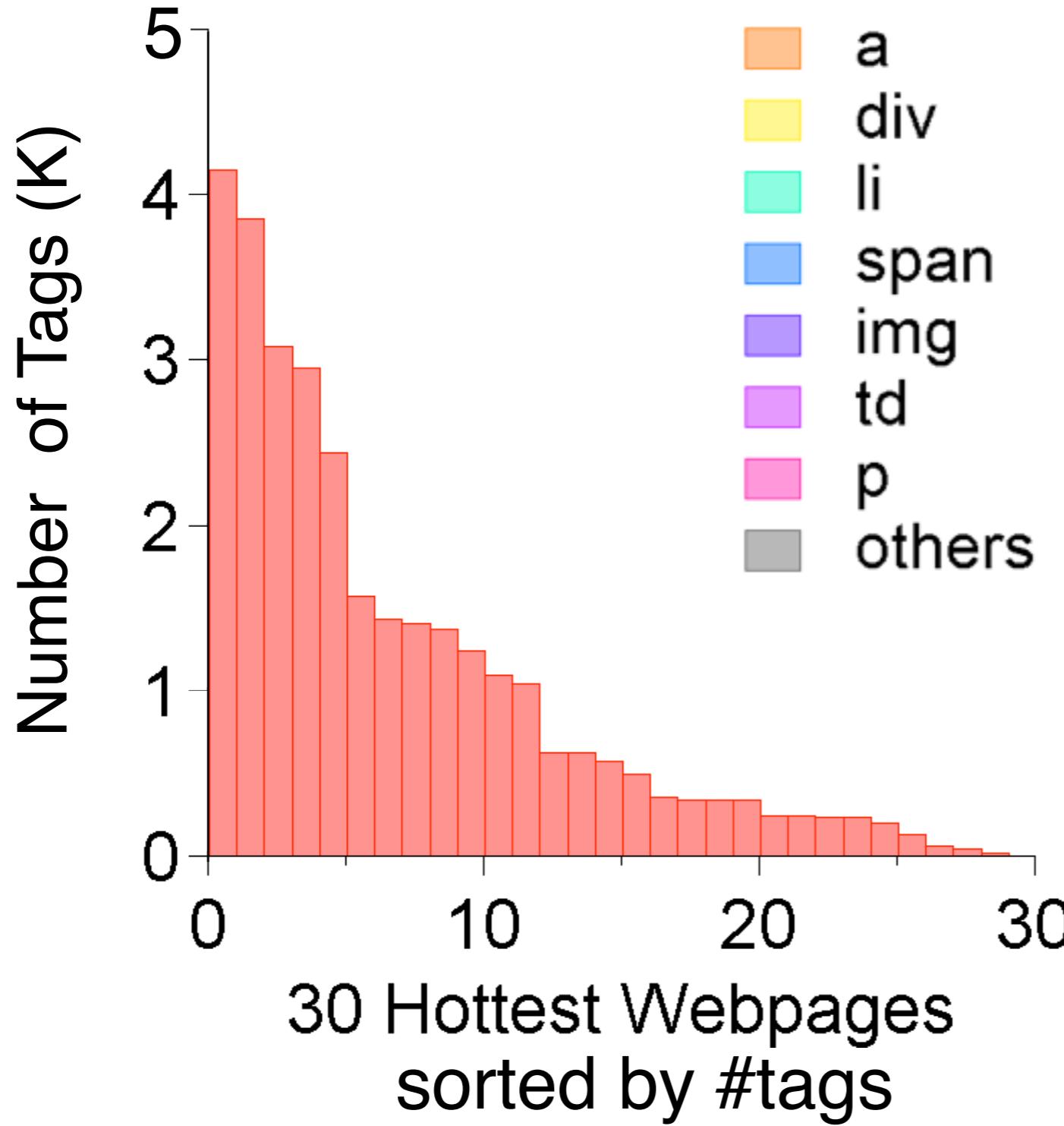
HTML Tag Analysis



Webpages have different tag counts (instruction counts)

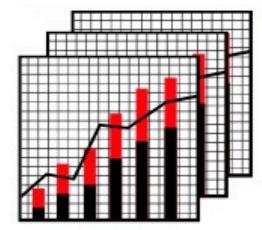
Webpages have a few hot HTML tags (hot instructions)

HTML Tag Analysis

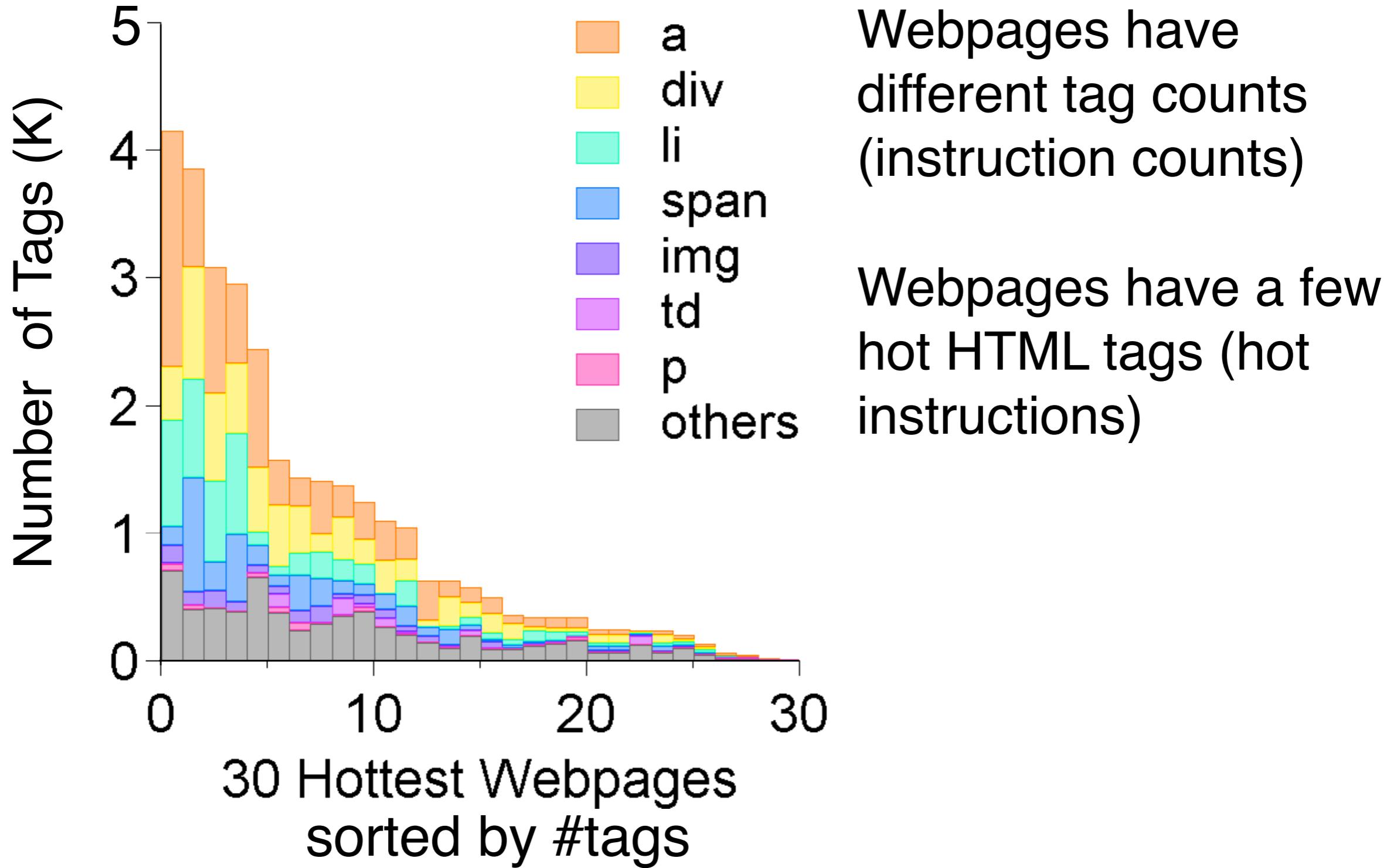


Webpages have different tag counts (instruction counts)

Webpages have a few hot HTML tags (hot instructions)

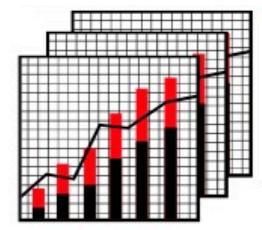


HTML Tag Analysis

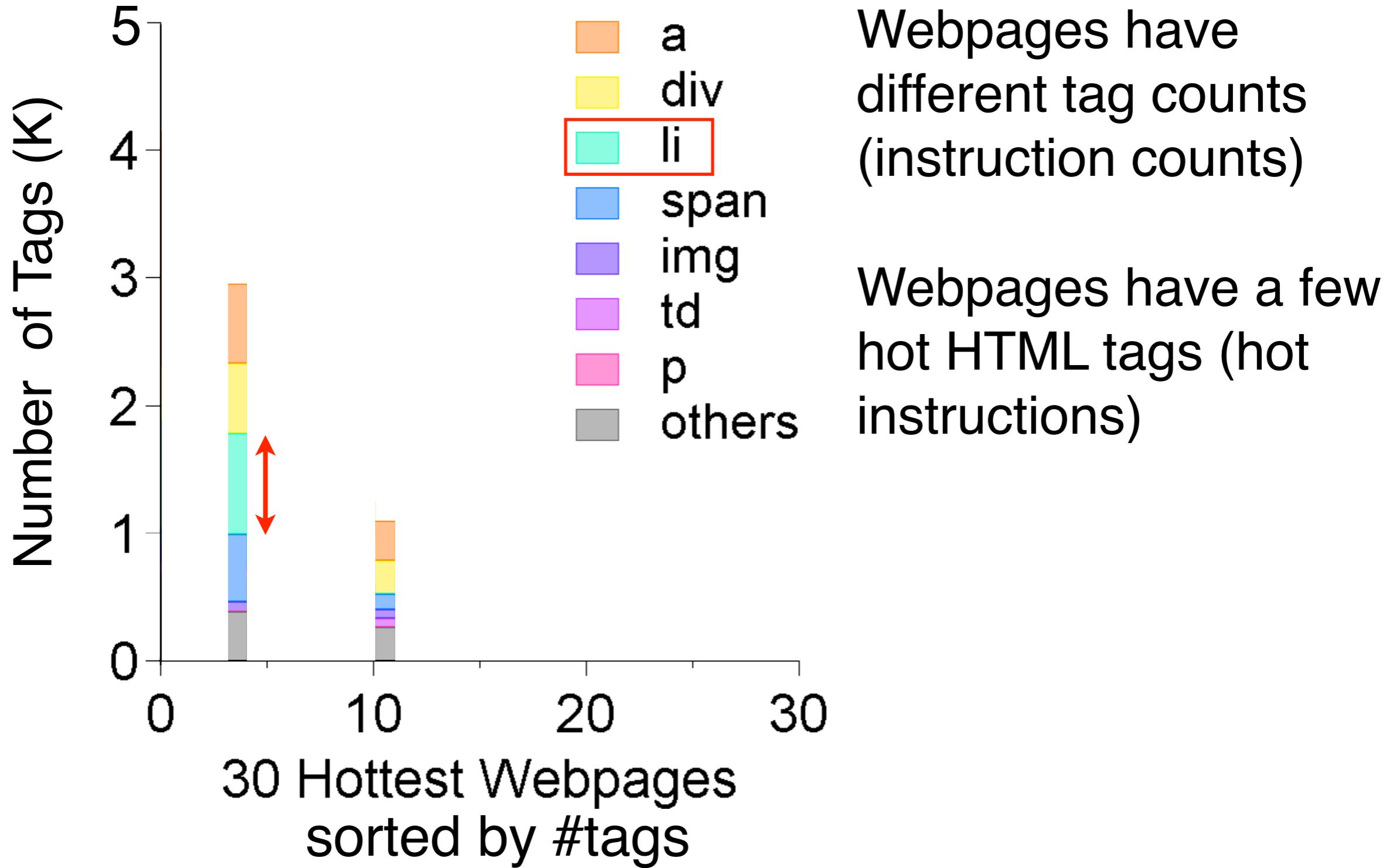


Webpages have different tag counts (instruction counts)

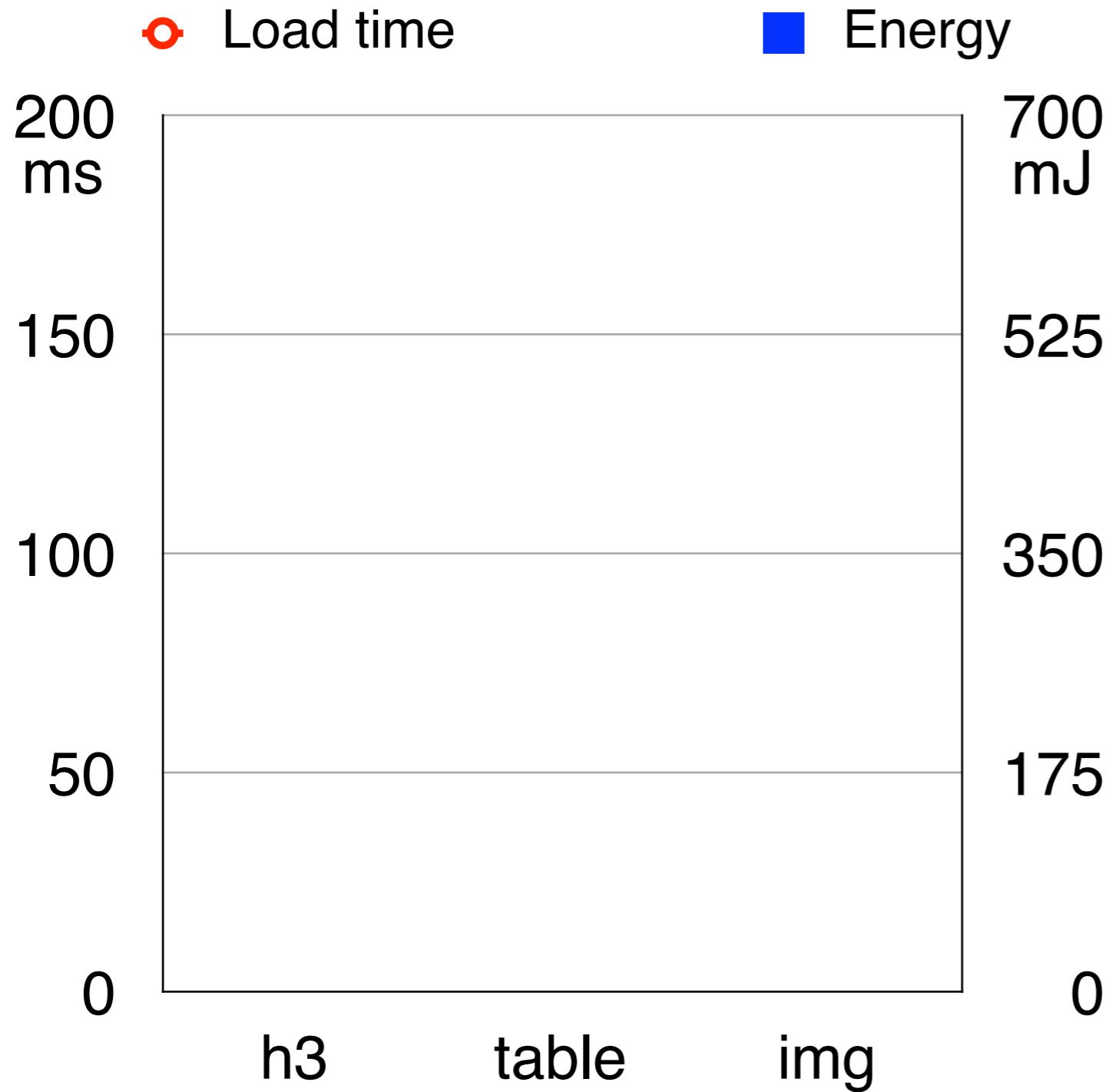
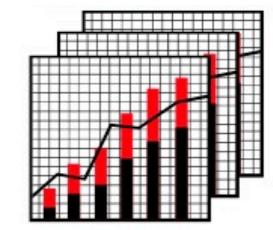
Webpages have a few hot HTML tags (hot instructions)



HTML Tag Analysis



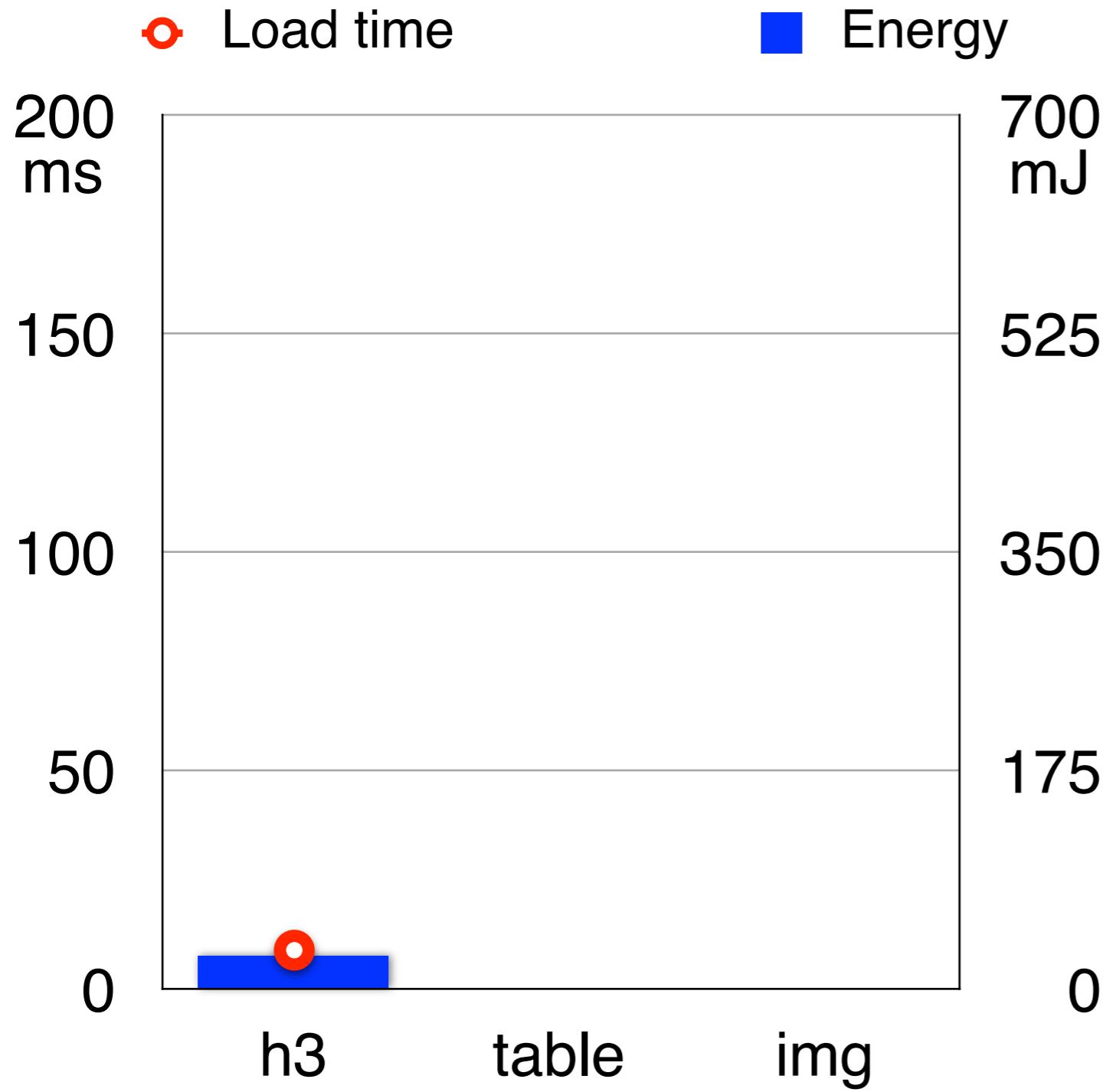
Tag Processing Overhead



Webpages have different tag counts (instruction counts)

Webpages have a few hot HTML tags (hot instructions)

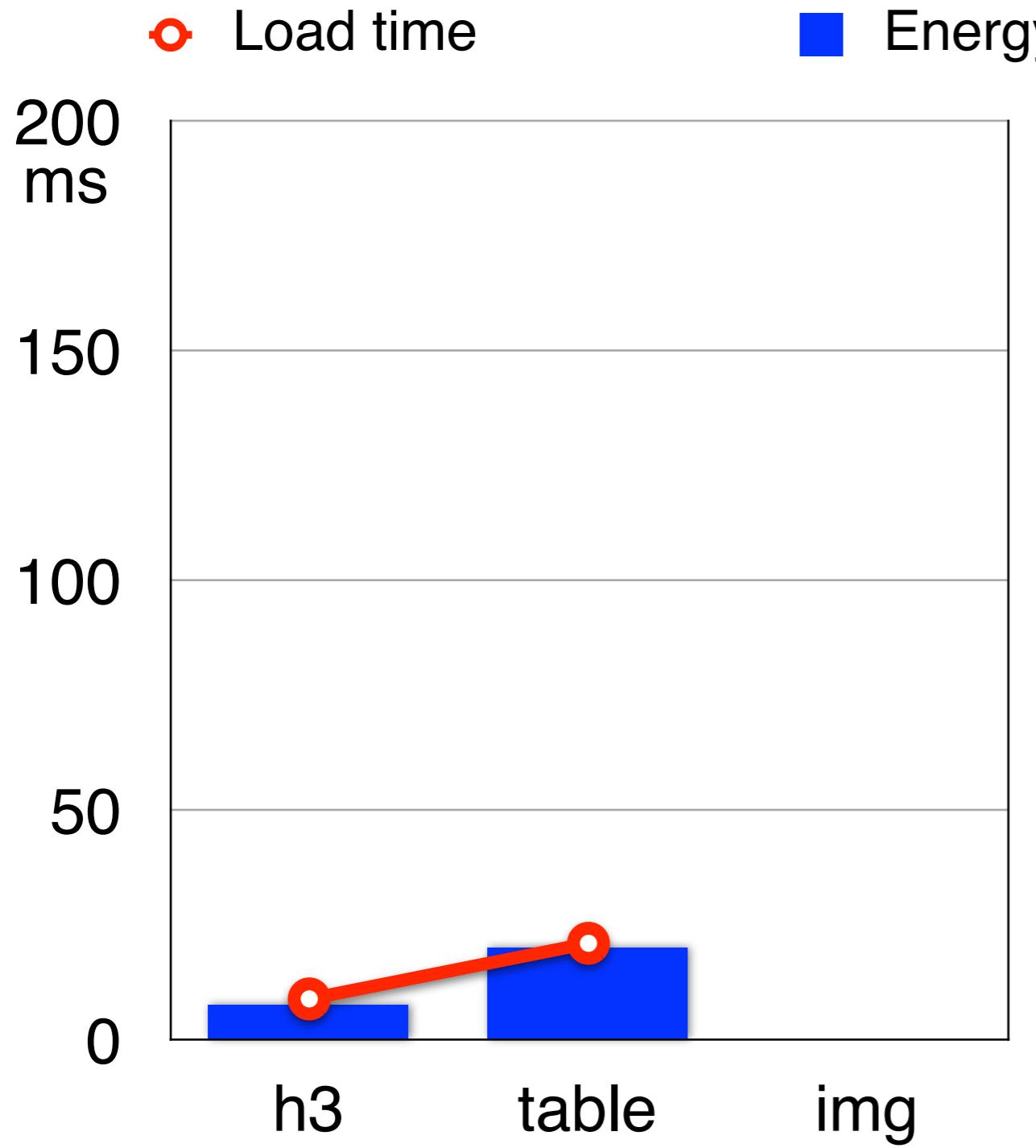
Tag Processing Overhead



Webpages have different tag counts (instruction counts)

Webpages have a few hot HTML tags (hot instructions)

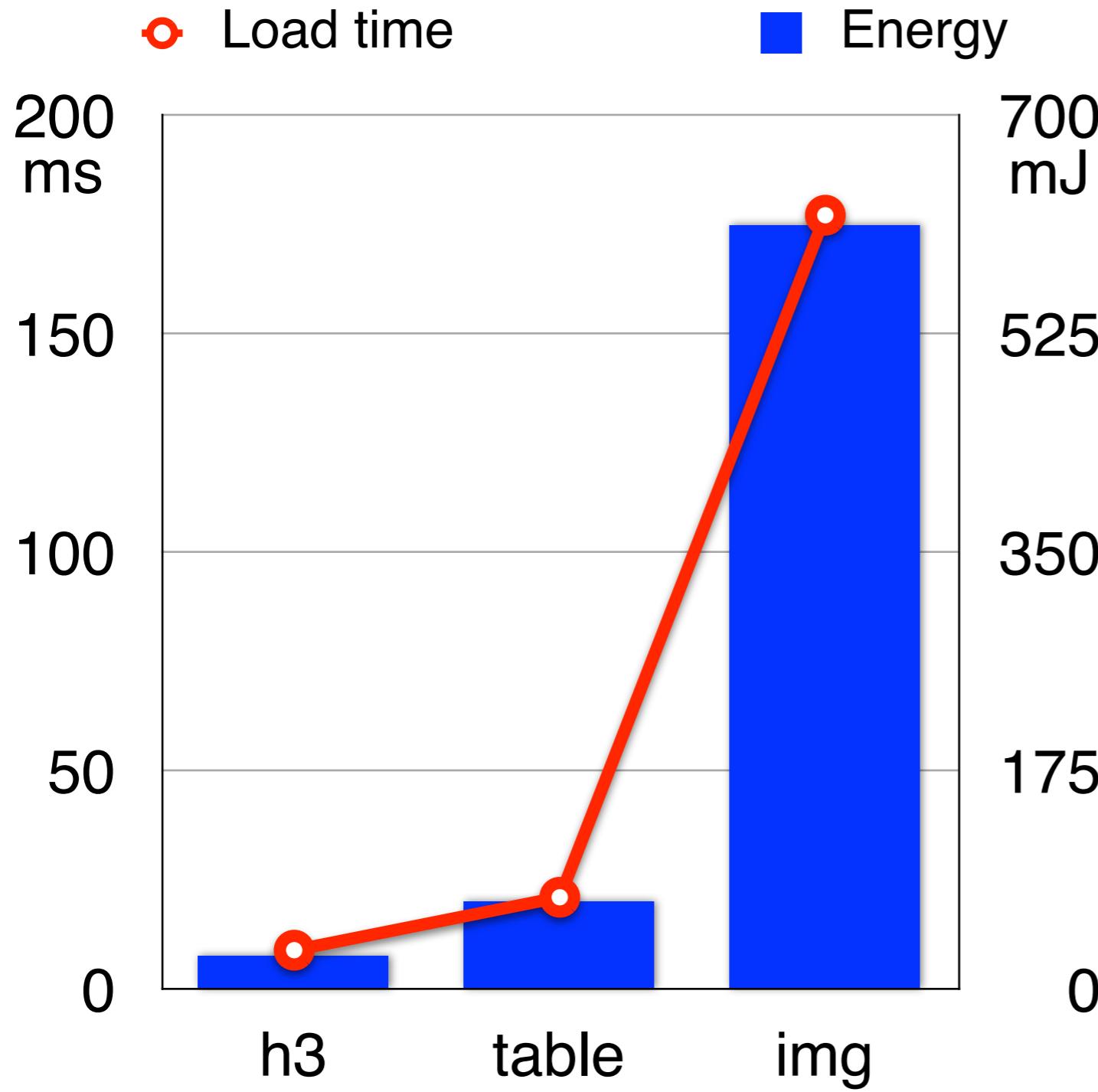
Tag Processing Overhead



Webpages have different tag counts (instruction counts)

Webpages have a few hot HTML tags (hot instructions)

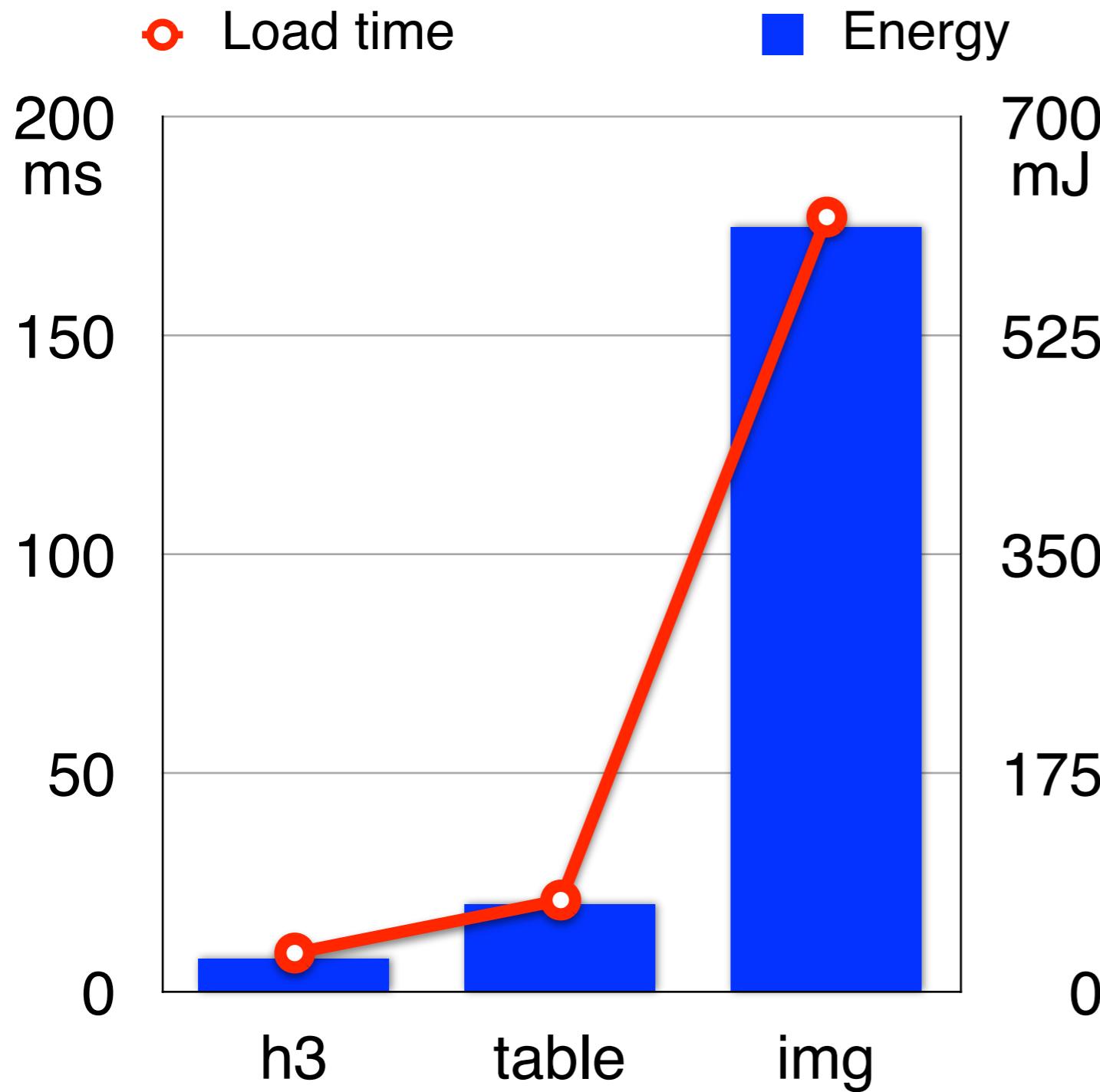
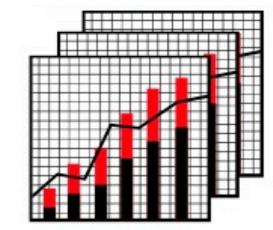
Tag Processing Overhead



Webpages have different tag counts (instruction counts)

Webpages have a few hot HTML tags (hot instructions)

Tag Processing Overhead

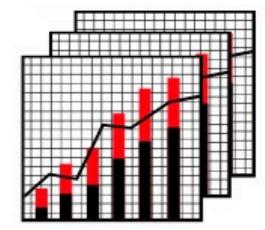


Webpages have different tag counts (instruction counts)

Webpages have a few hot HTML tags (hot instructions)

HTML tags have different processing overhead (time & energy)

Characterization Conclusions



Characterization Conclusions



- **Root cause** of load time and energy variance

Characterization Conclusions



- **Root cause** of load time and energy variance
 - 1. Webpages have different HTML tag **counts and mixes**

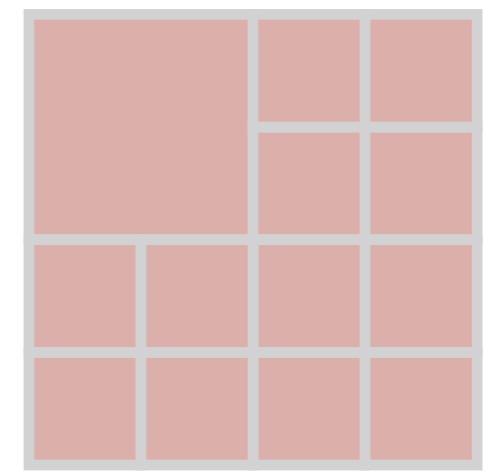
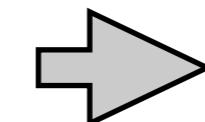
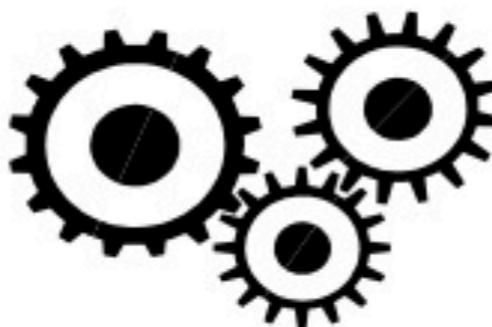
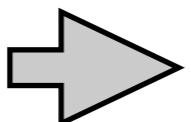
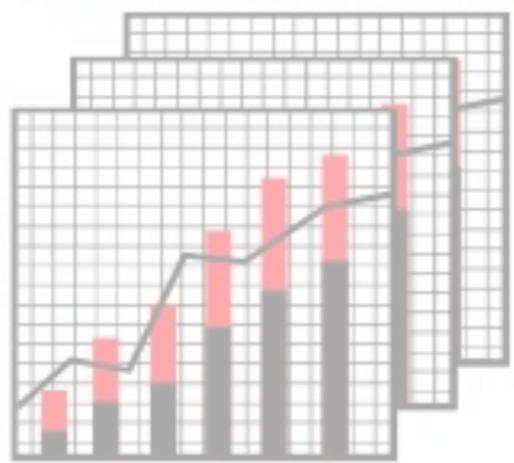
Characterization Conclusions



- **Root cause** of load time and energy variance
 - 1. Webpages have different HTML tag **counts and mixes**
 - 2. Individual HTML tags involve different **processing overheads**

Our Approach

How to predict the time and energy of webpage loading?



Workload
Characterization

**Performance/Energy
Prediction**

Resource
Management

Regression Modeling Strategy



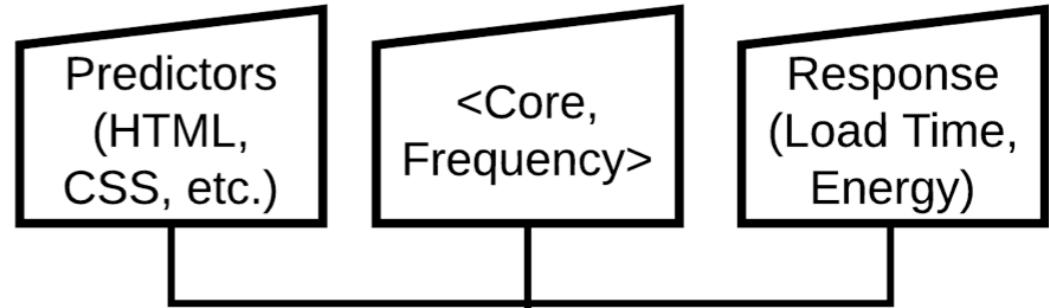
Idea: predict webpage load time and energy consumption (responses) using webpage characteristics (predictors)

Regression Modeling Strategy



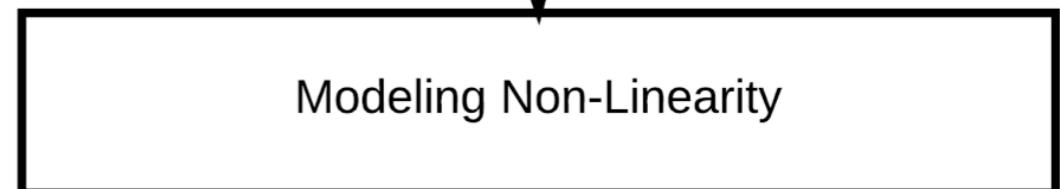
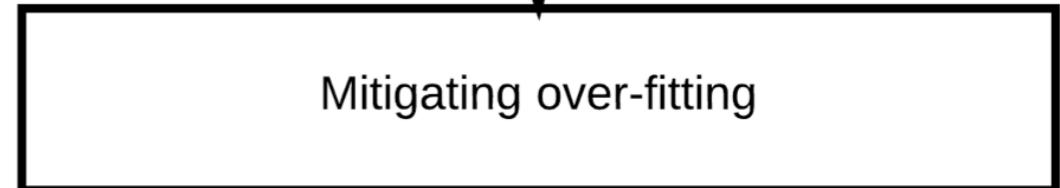
Identify Training Predictors and Responses

Training using hottest 2,500 webpages



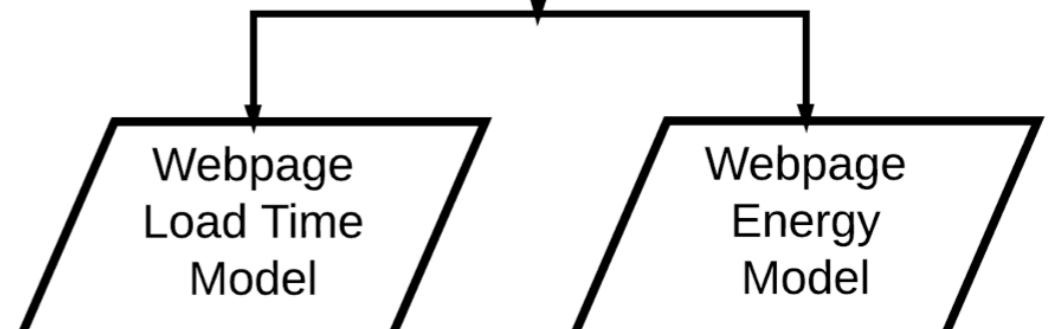
Model Construction and Refinement

Start from the linear model and progressively refine it

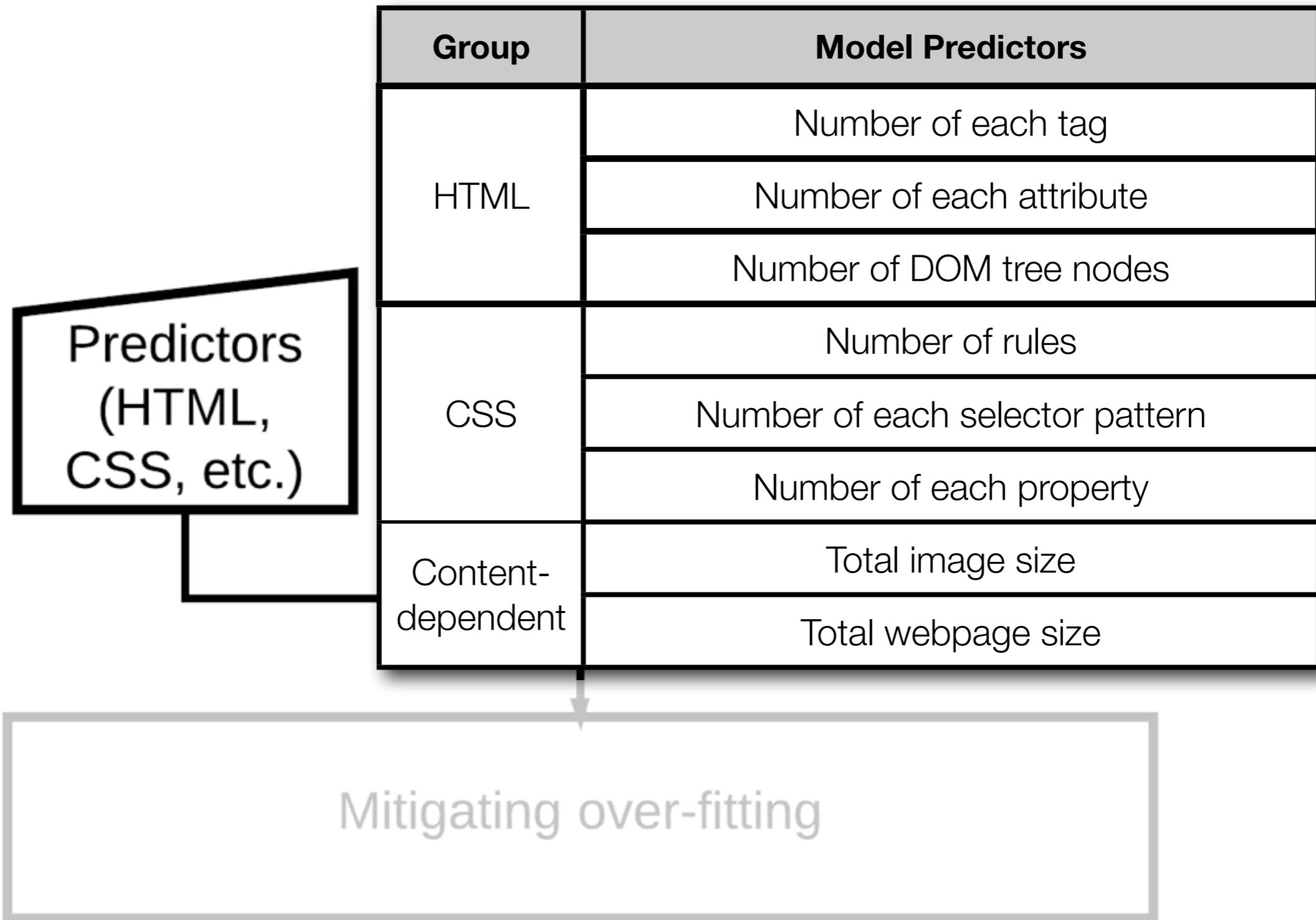
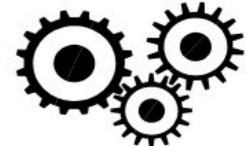


Model Validation

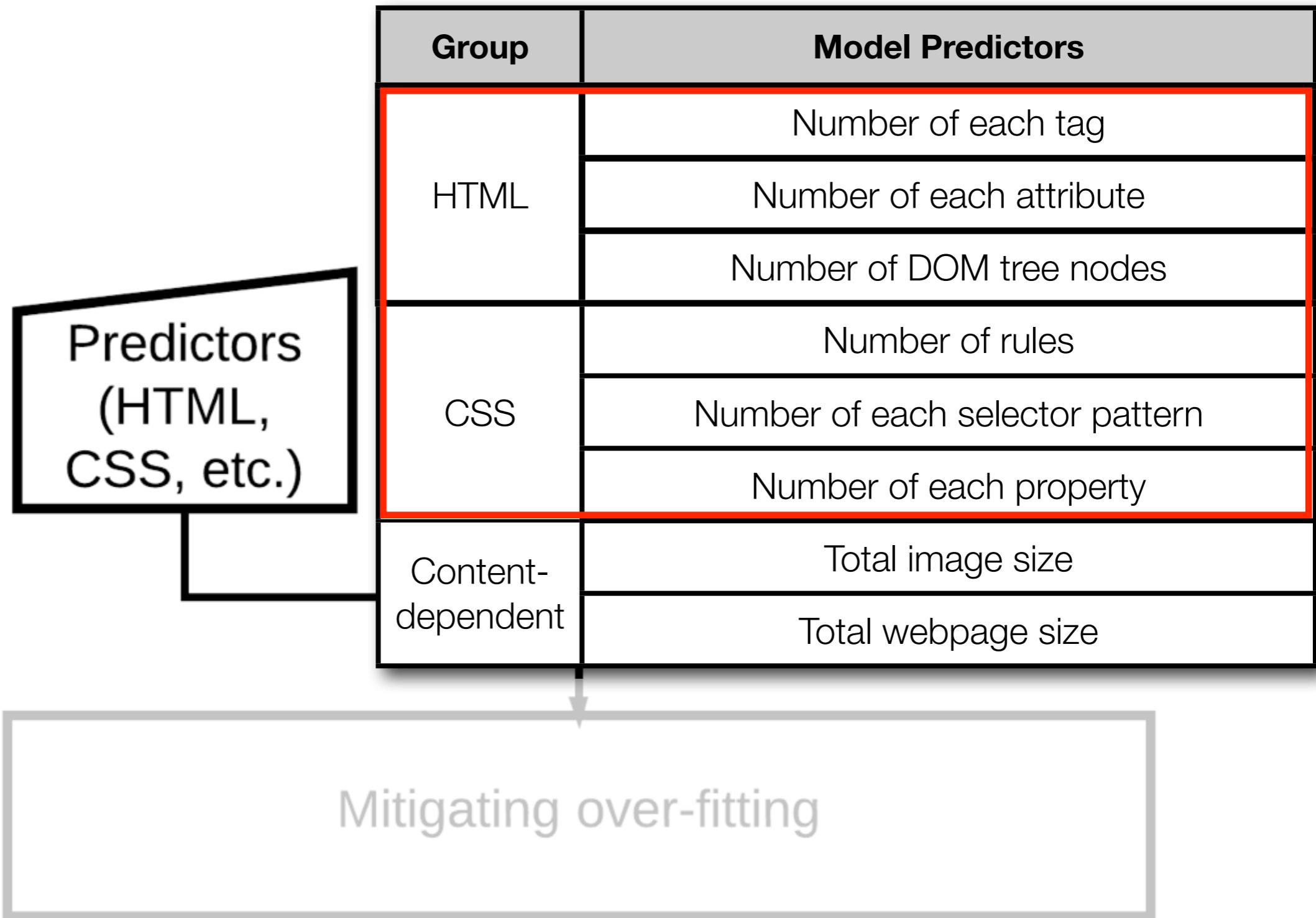
Validating on another 2,500 webpages



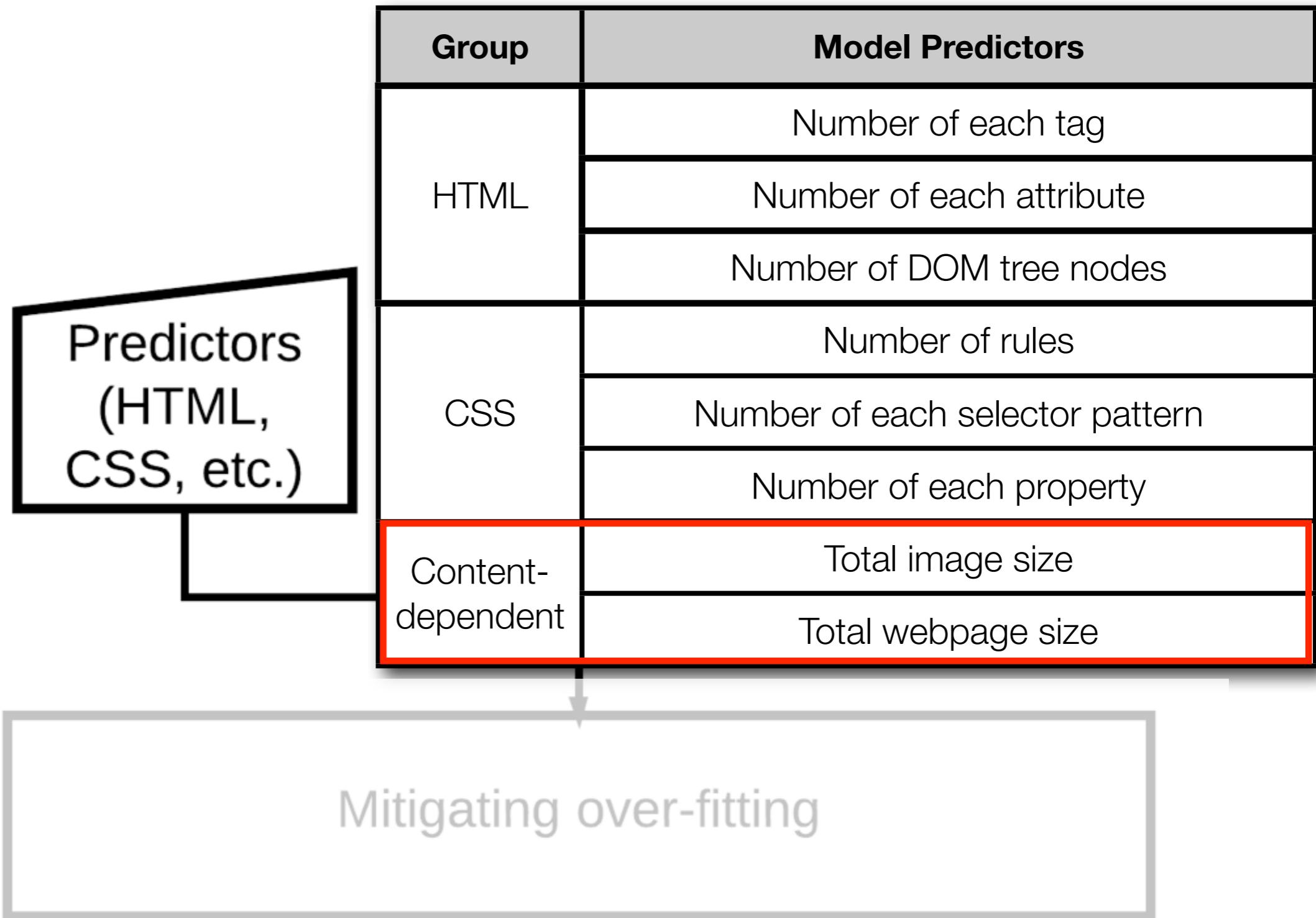
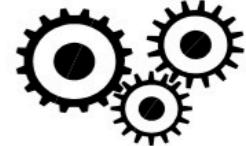
Regression Modeling



Regression Modeling



Regression Modeling





Regression Modeling

Mitigating over-fitting

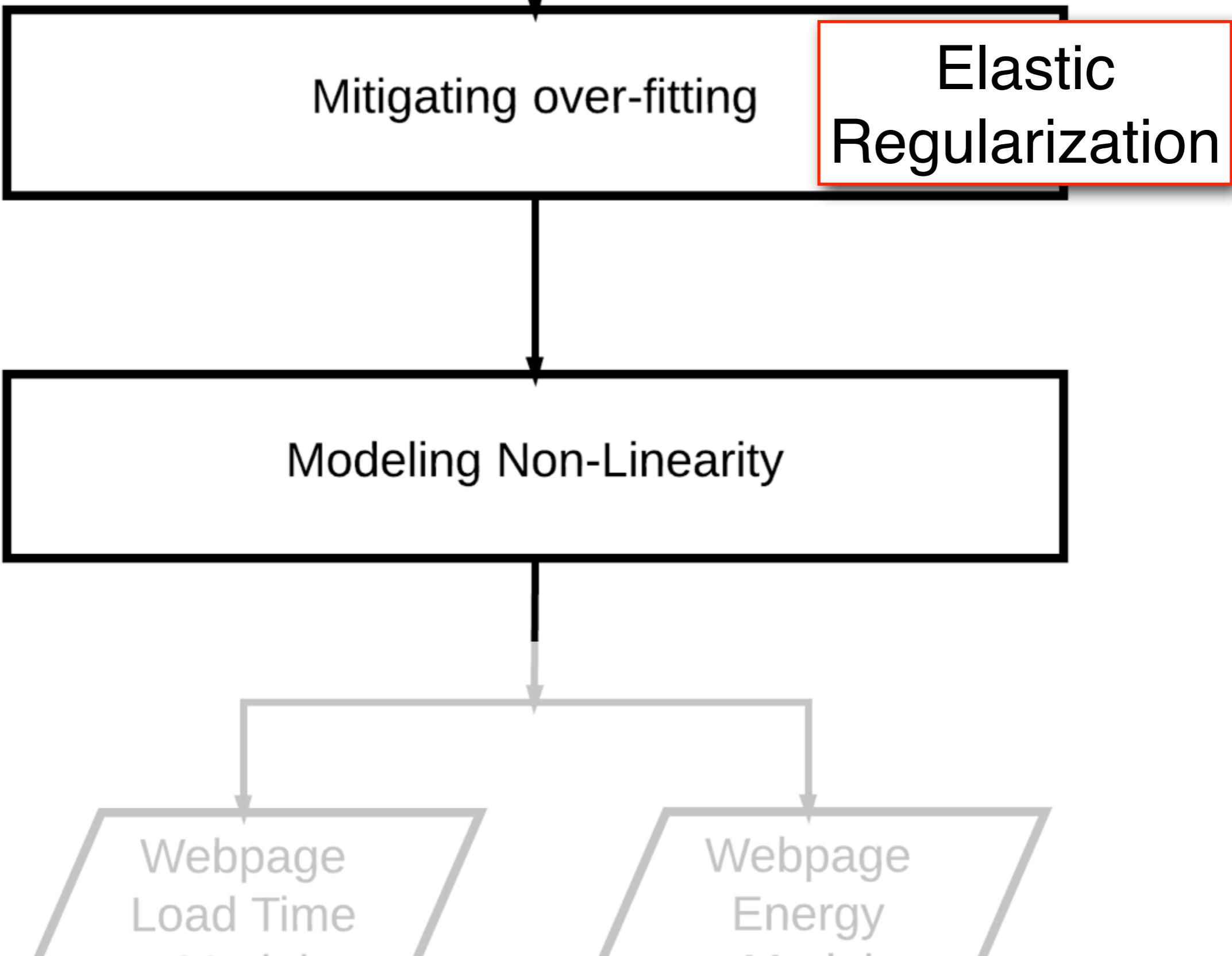
Modeling Non-Linearity

Webpage
Load Time

Webpage
Energy

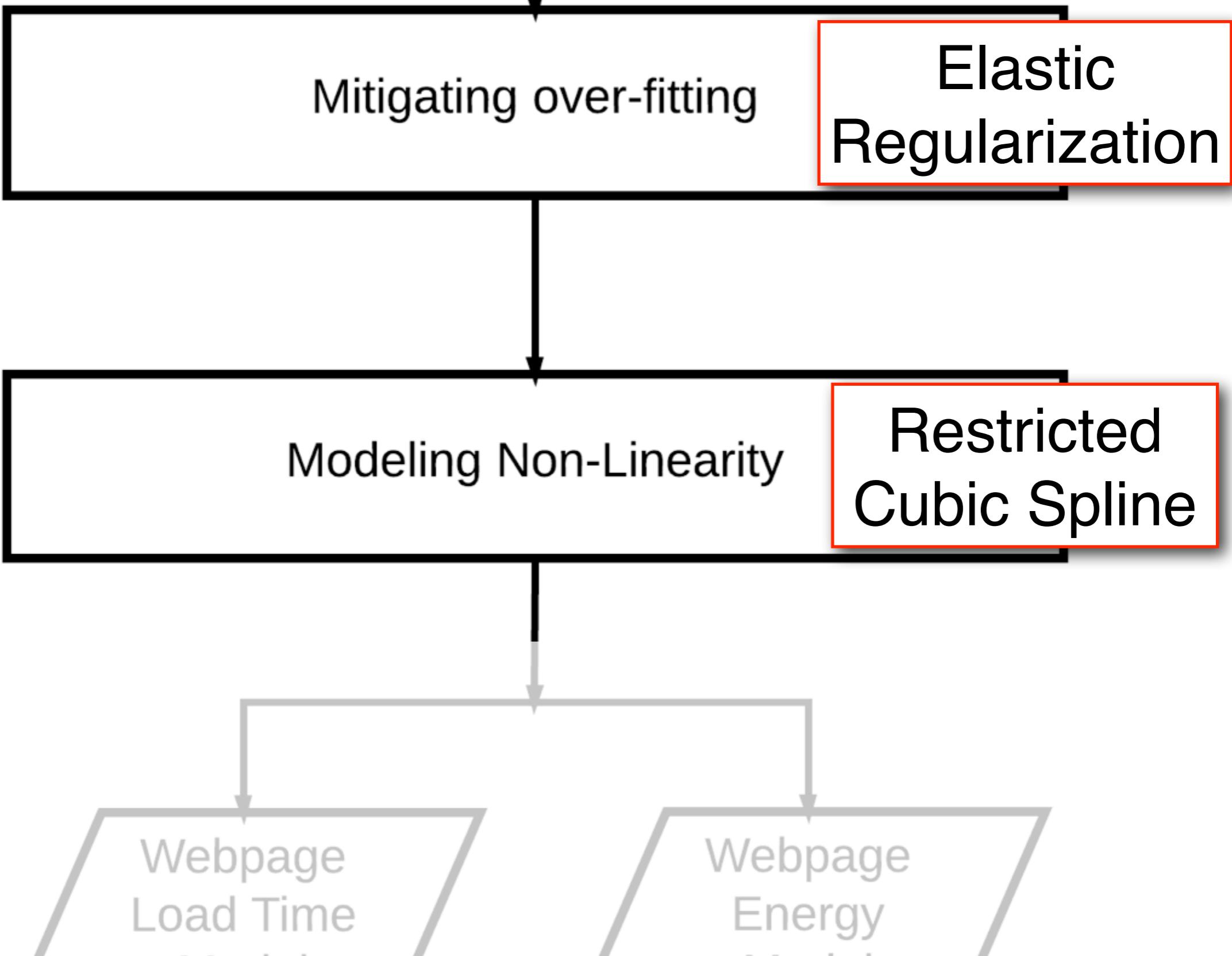


Regression Modeling

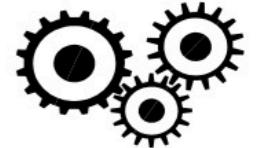




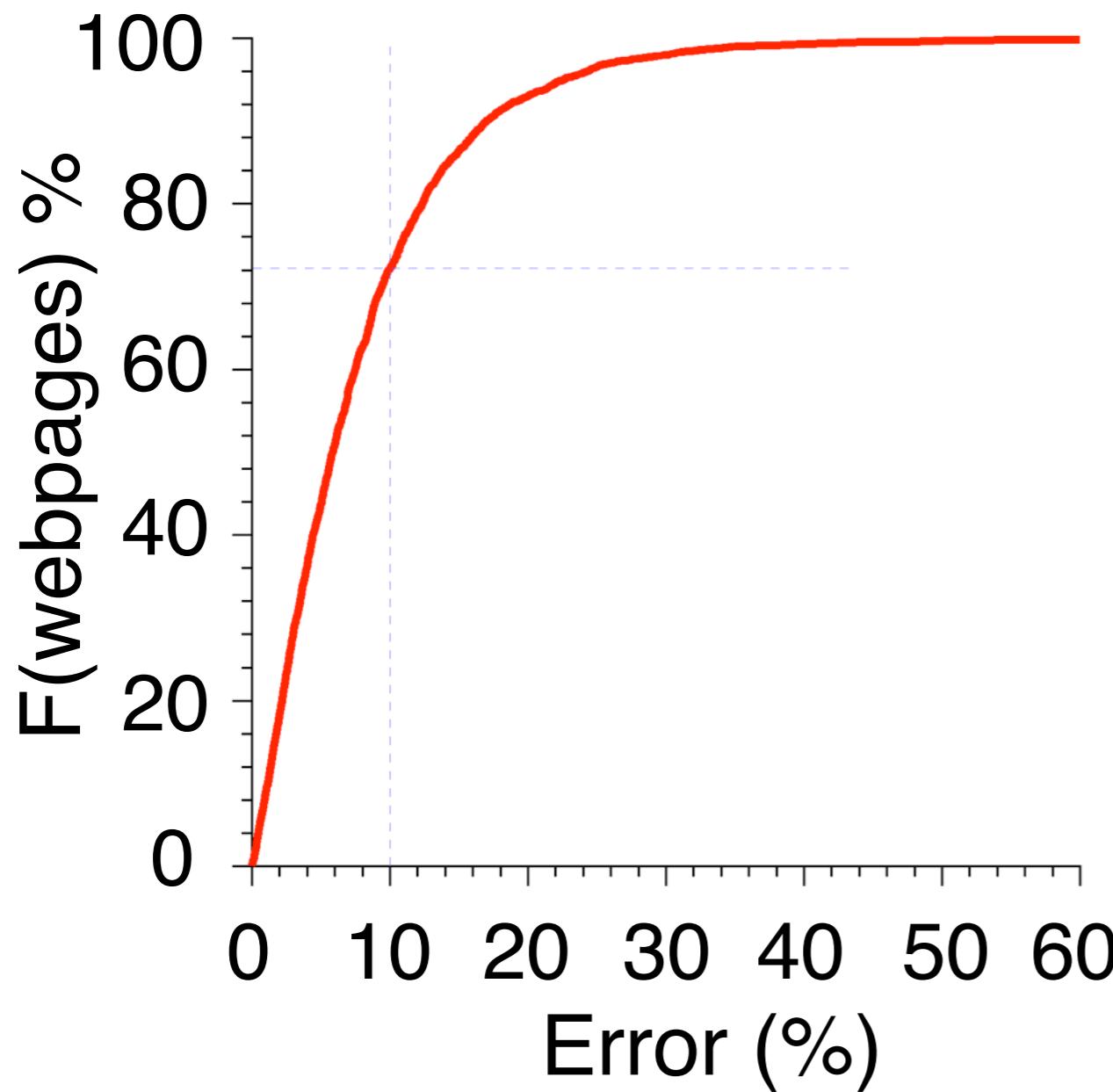
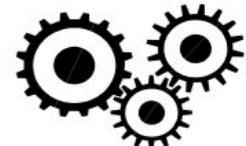
Regression Modeling



Regression Modeling



Regression Modeling

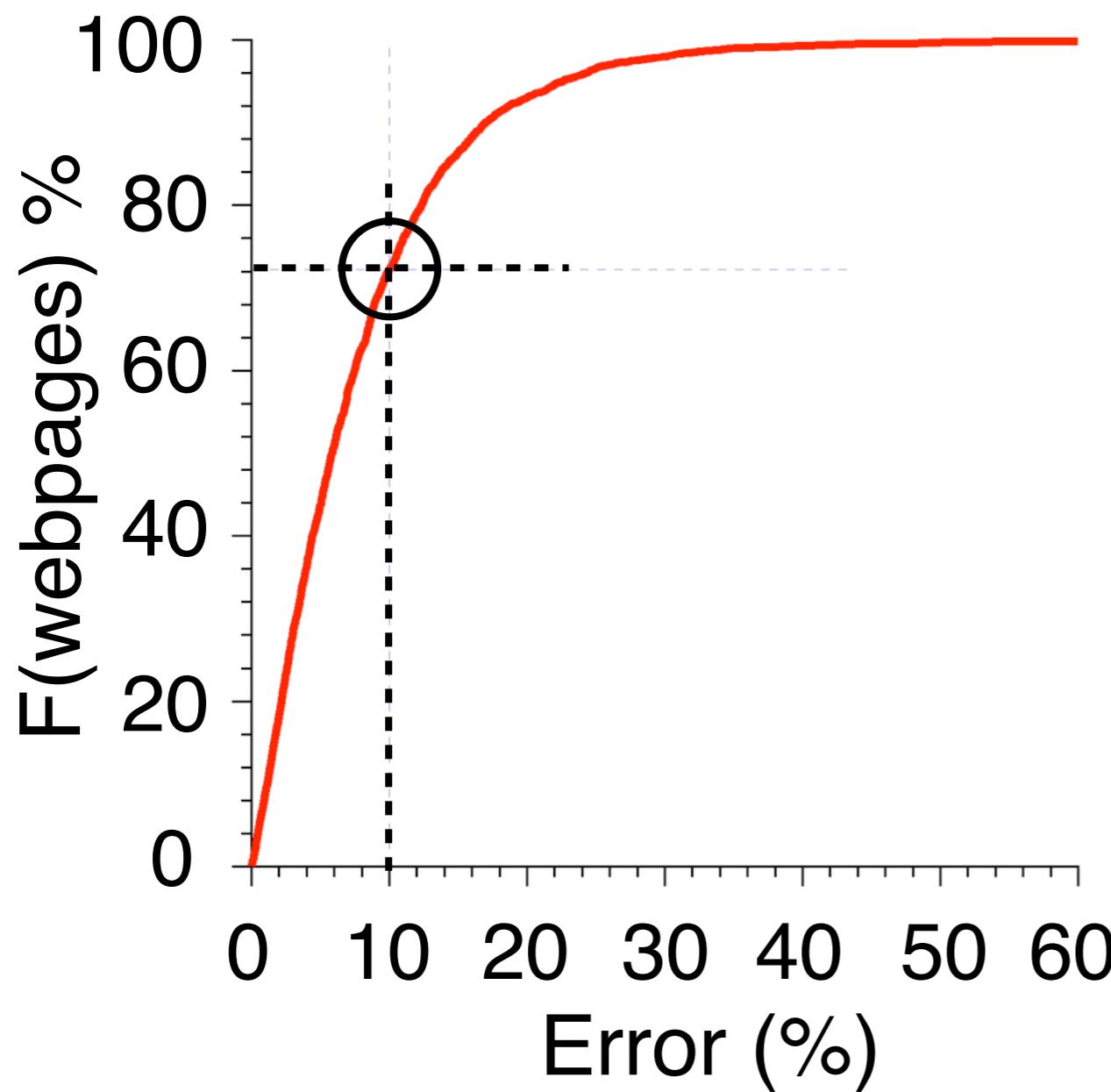


Regression Modeling

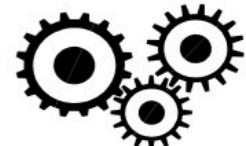


Webpage
Load Time
Model

Webpage
Energy
Model

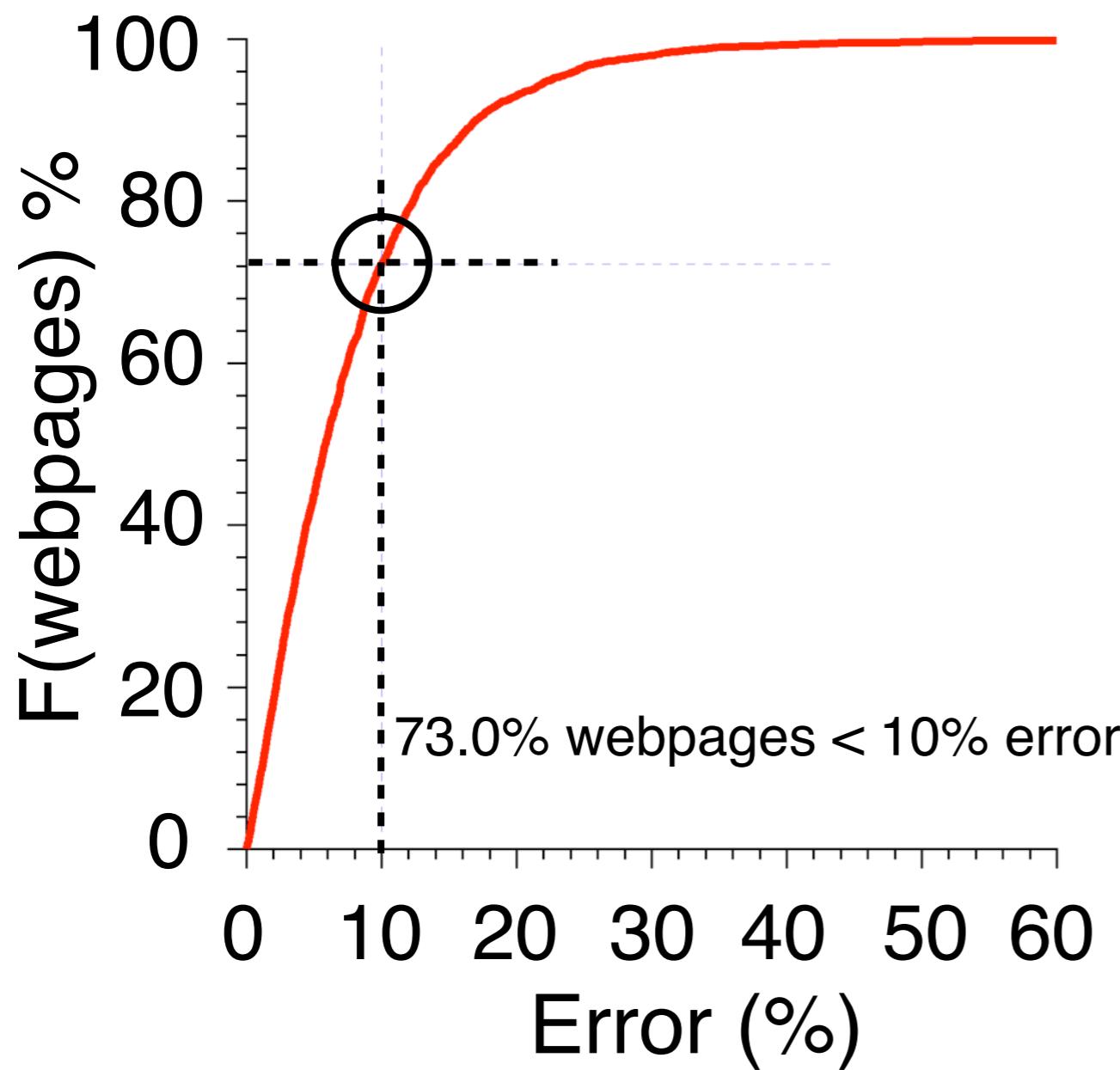


Regression Modeling



Webpage
Load Time
Model

Webpage
Energy
Model

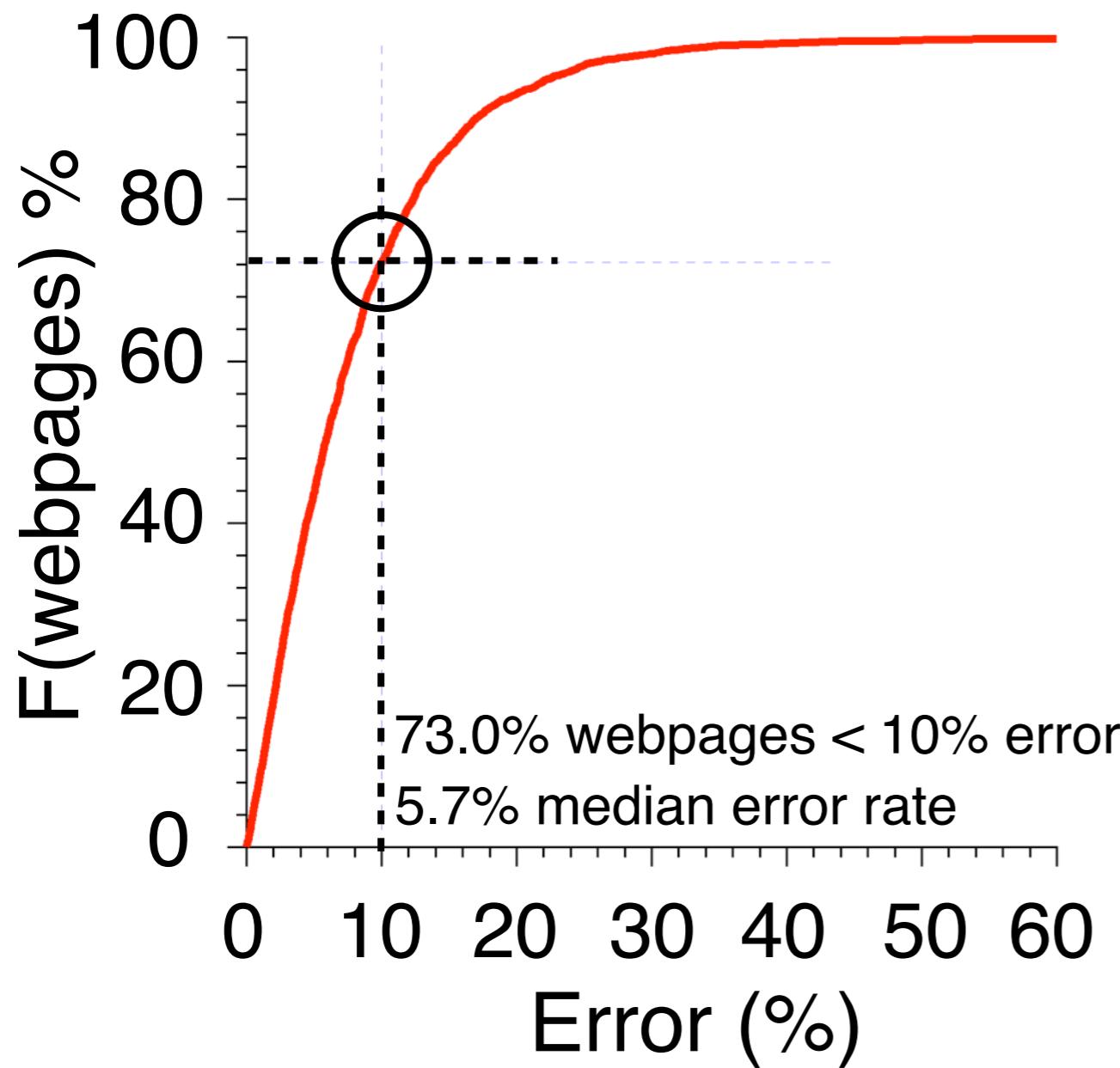


Regression Modeling



Webpage
Load Time
Model

Webpage
Energy
Model

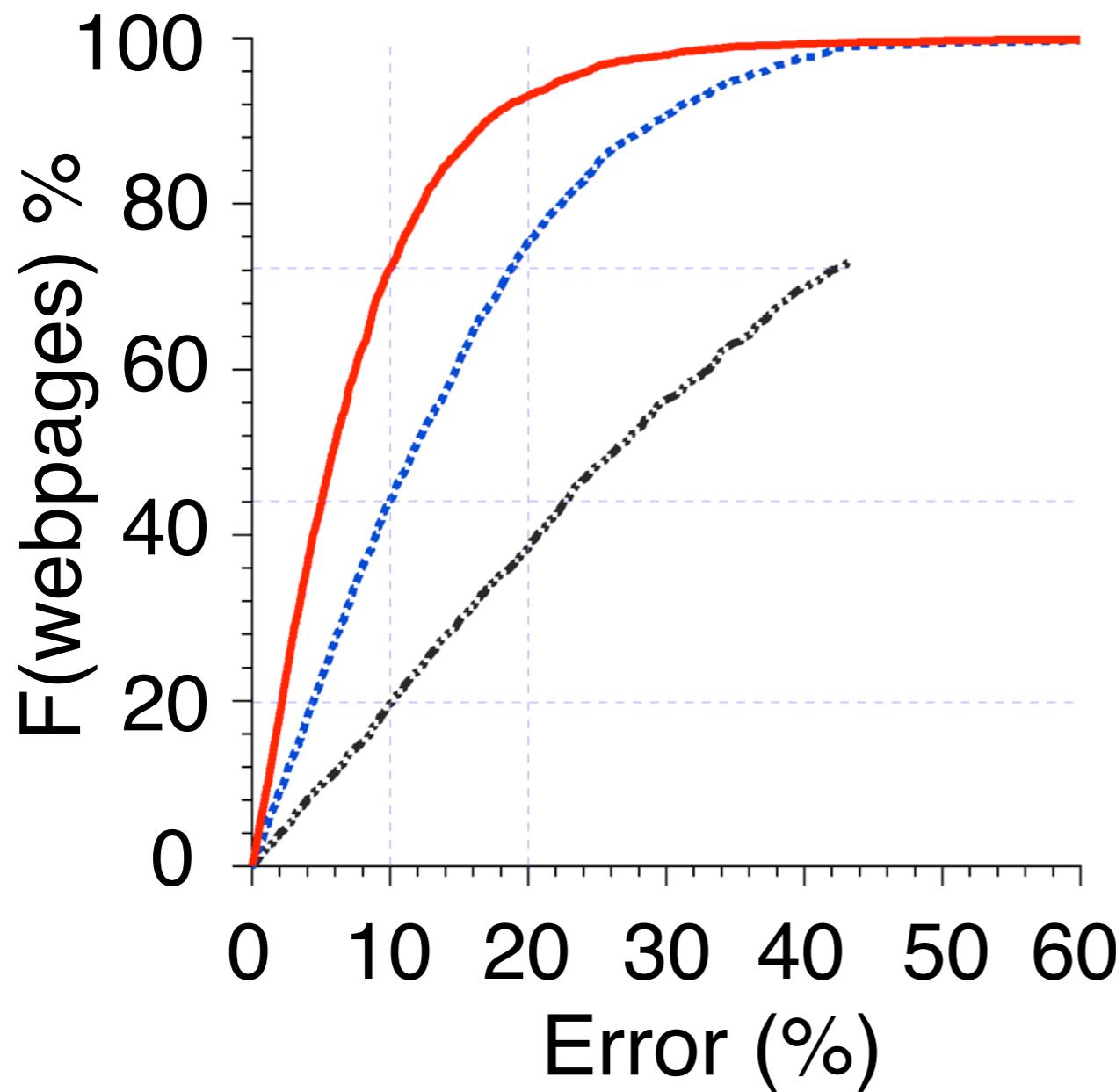


Regression Modeling



Webpage
Load Time
Model

Webpage
Energy
Model

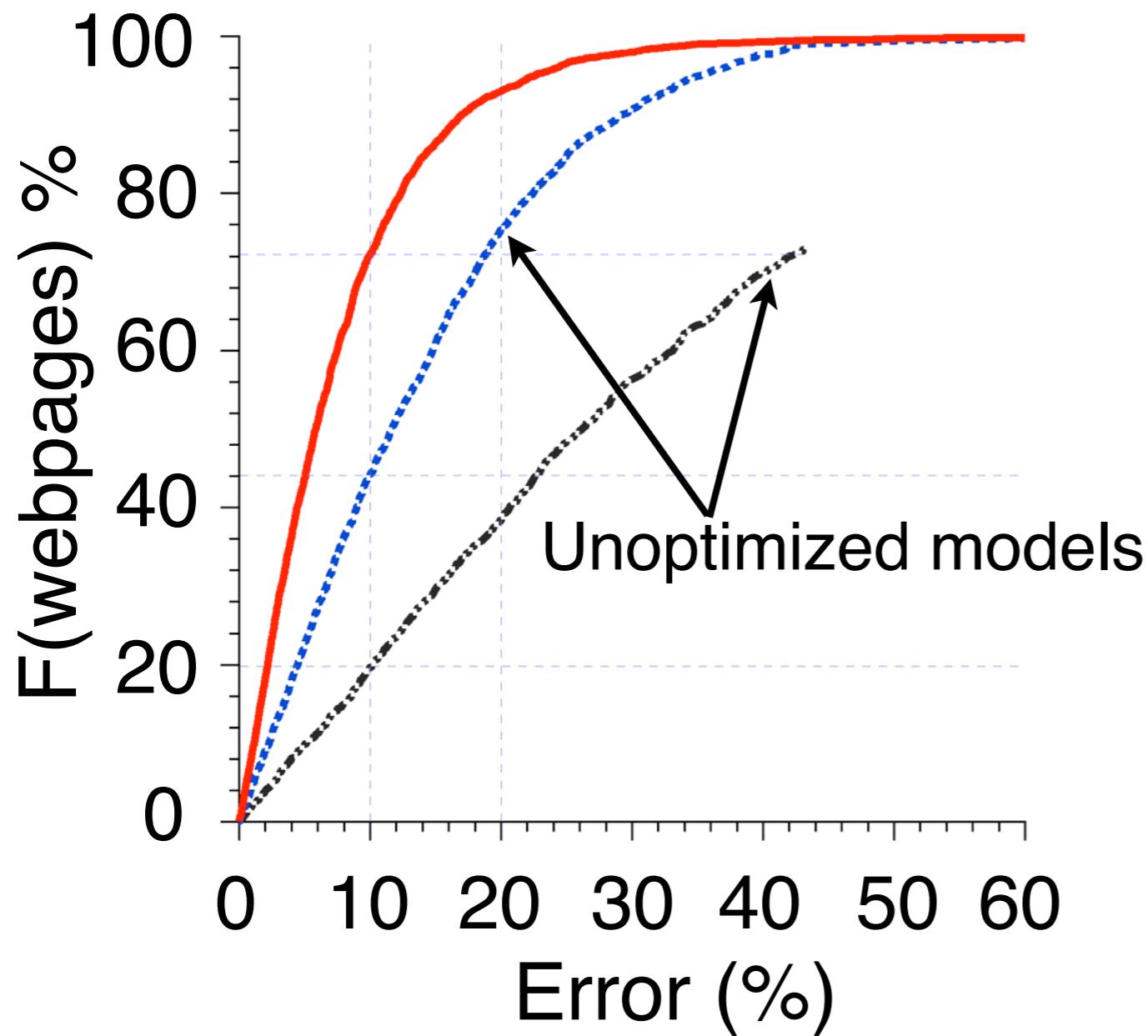


Regression Modeling

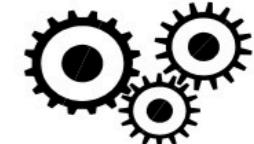


Webpage
Load Time
Model

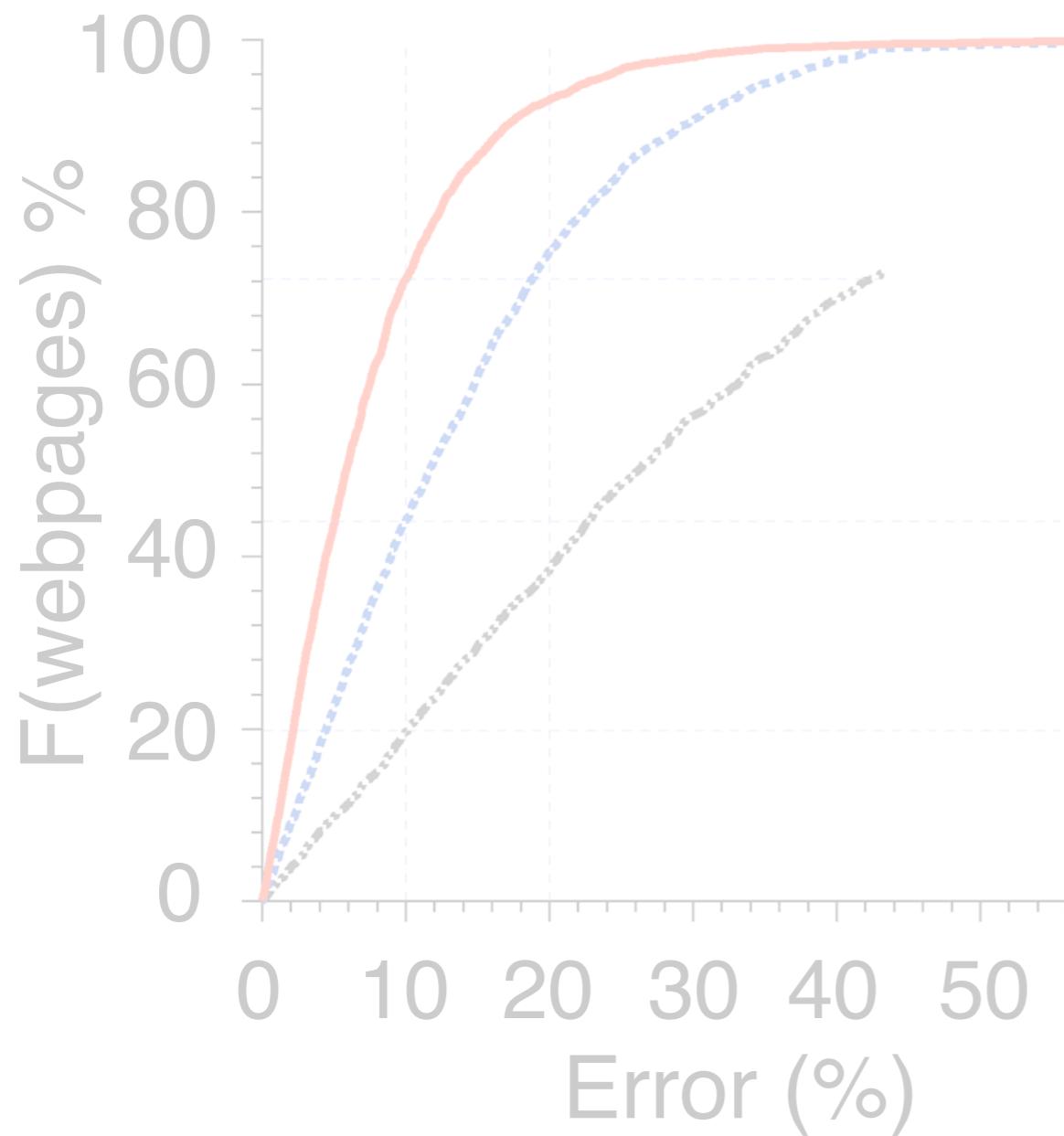
Webpage
Energy
Model



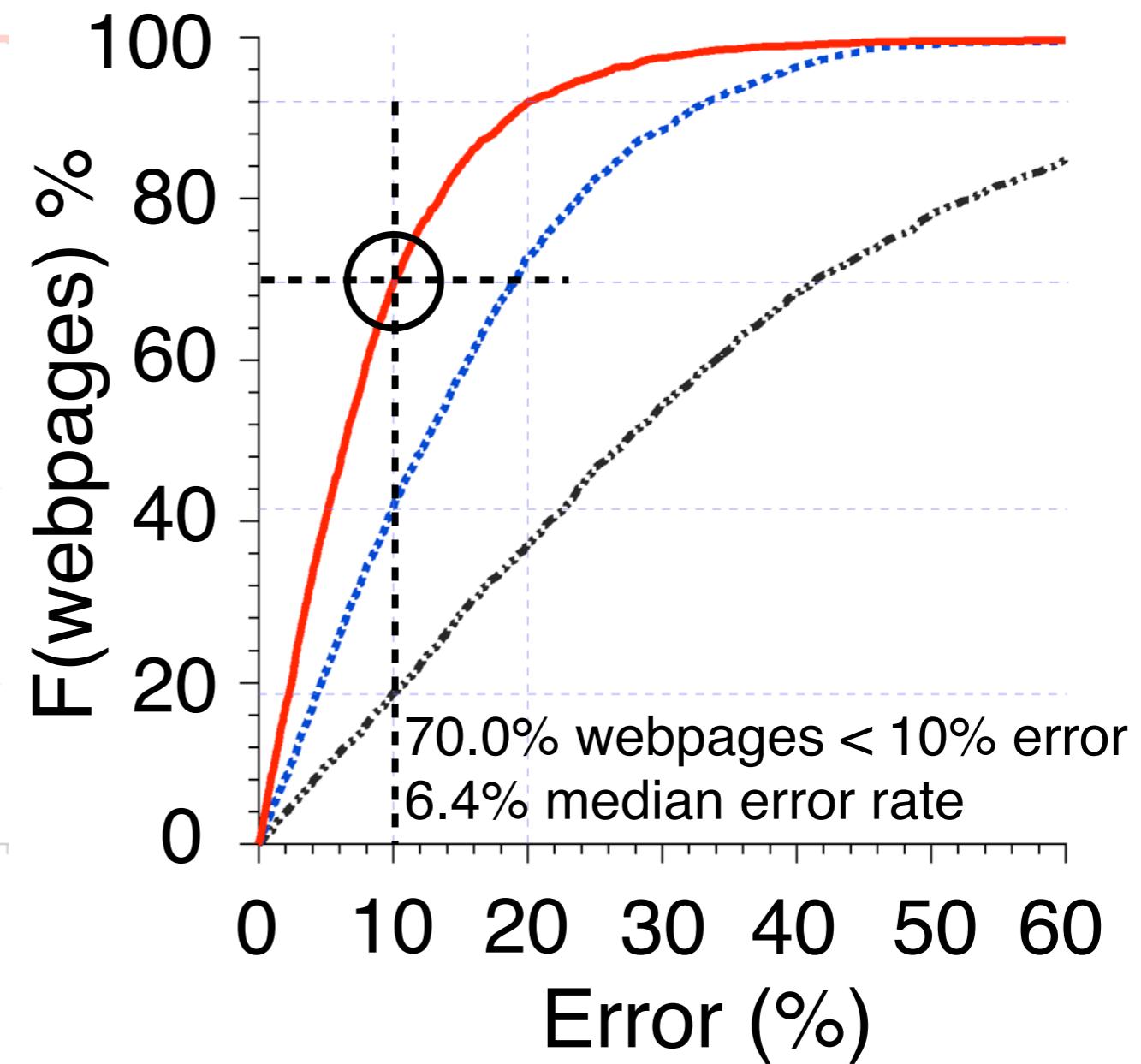
Regression Modeling



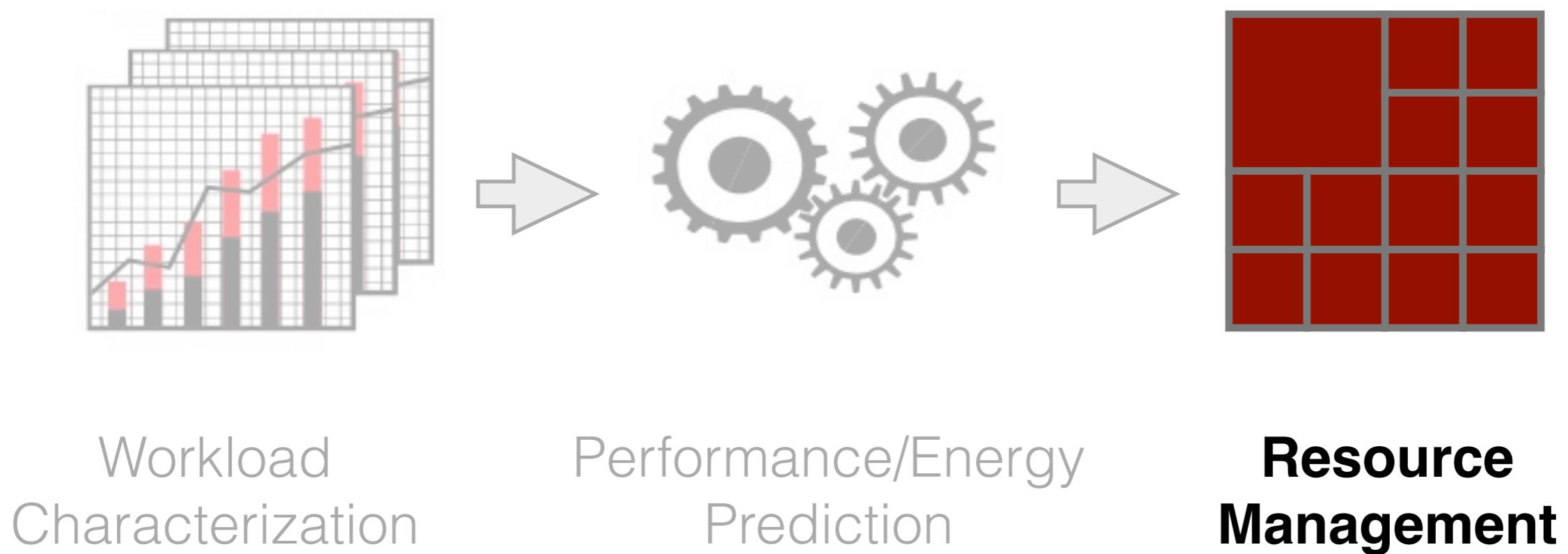
Webpage
Load Time
Model



Webpage
Energy
Model



Big/Little Scheduling

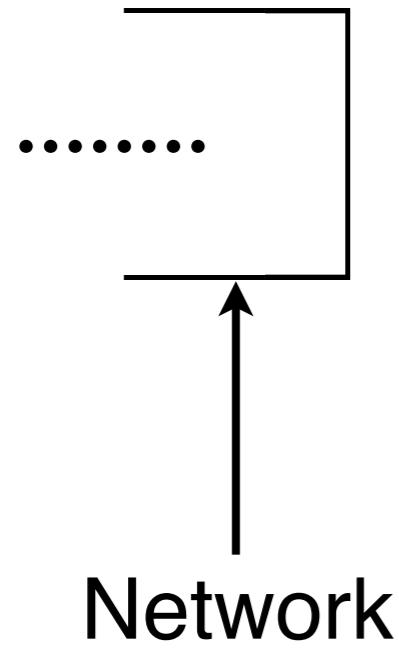
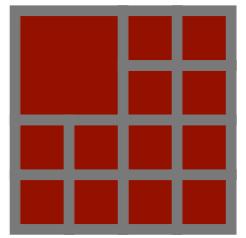


Workload
Characterization

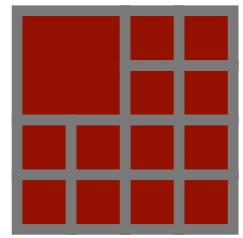
Performance/Energy
Prediction

**Resource
Management**

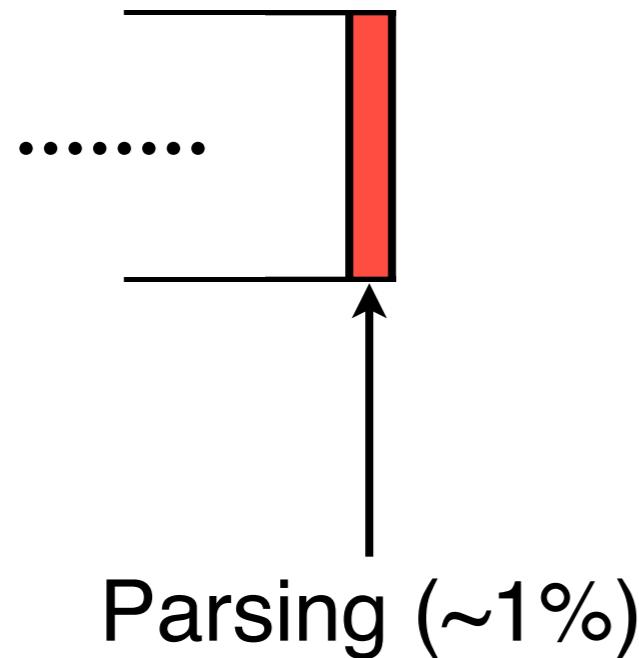
Webpage-aware Scheduling



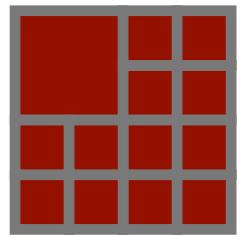
Webpage-aware Scheduling



Normal webpage rendering



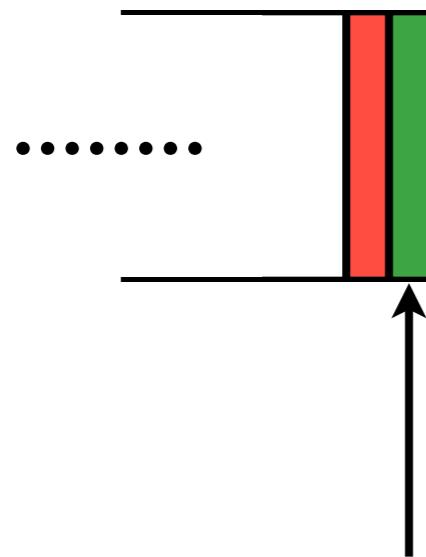
Webpage-aware Scheduling



Normal webpage rendering

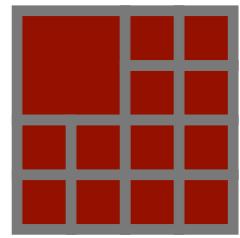


Webpage-aware scheduling



Prediction (Minimal overhead)

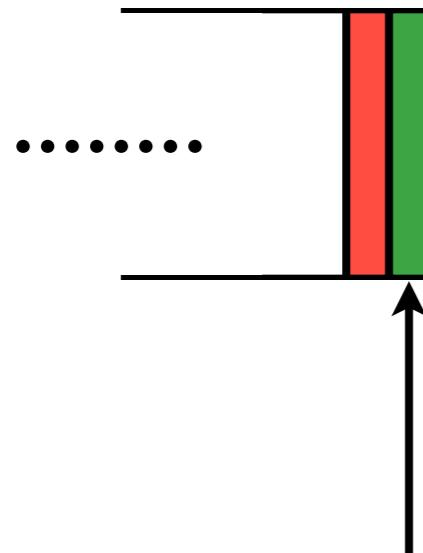
Webpage-aware Scheduling



Normal webpage rendering



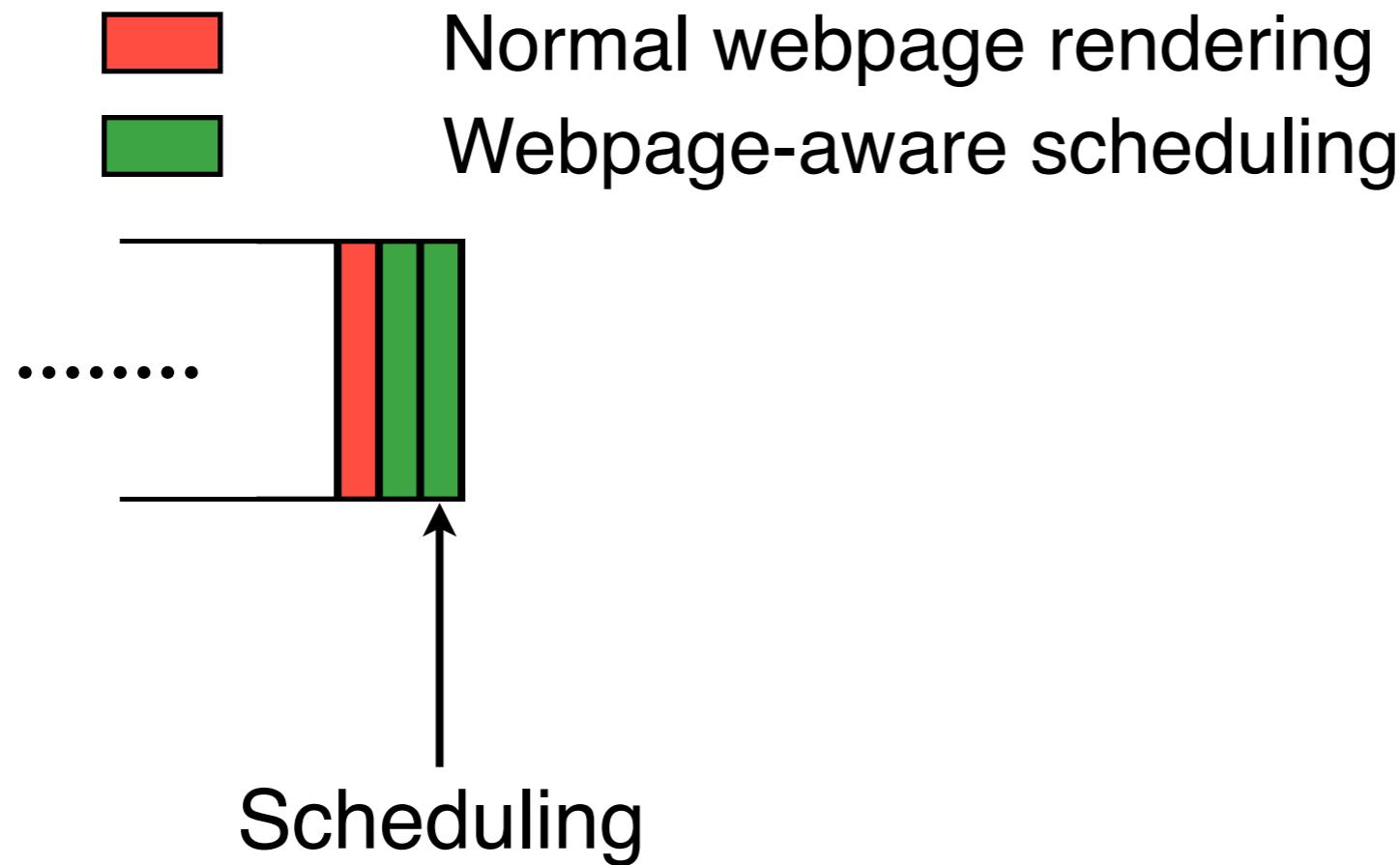
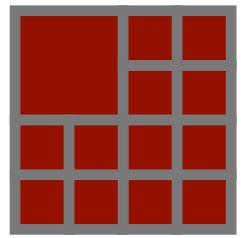
Webpage-aware scheduling



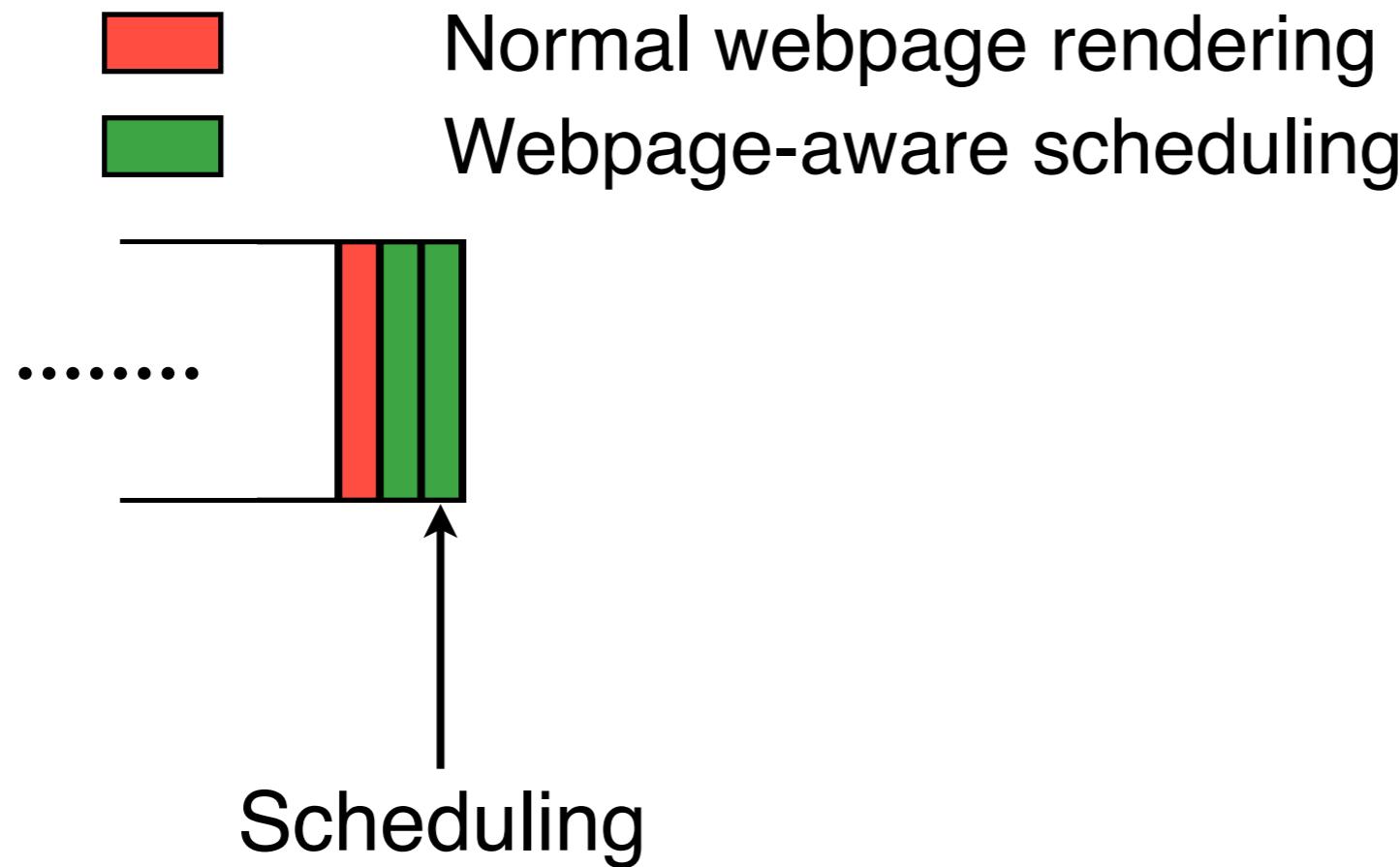
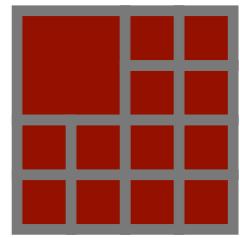
Prediction (Minimal overhead)

Predict the load time and energy for each $\langle \text{core}, \text{freq} \rangle$ conf.

Webpage-aware Scheduling



Webpage-aware Scheduling

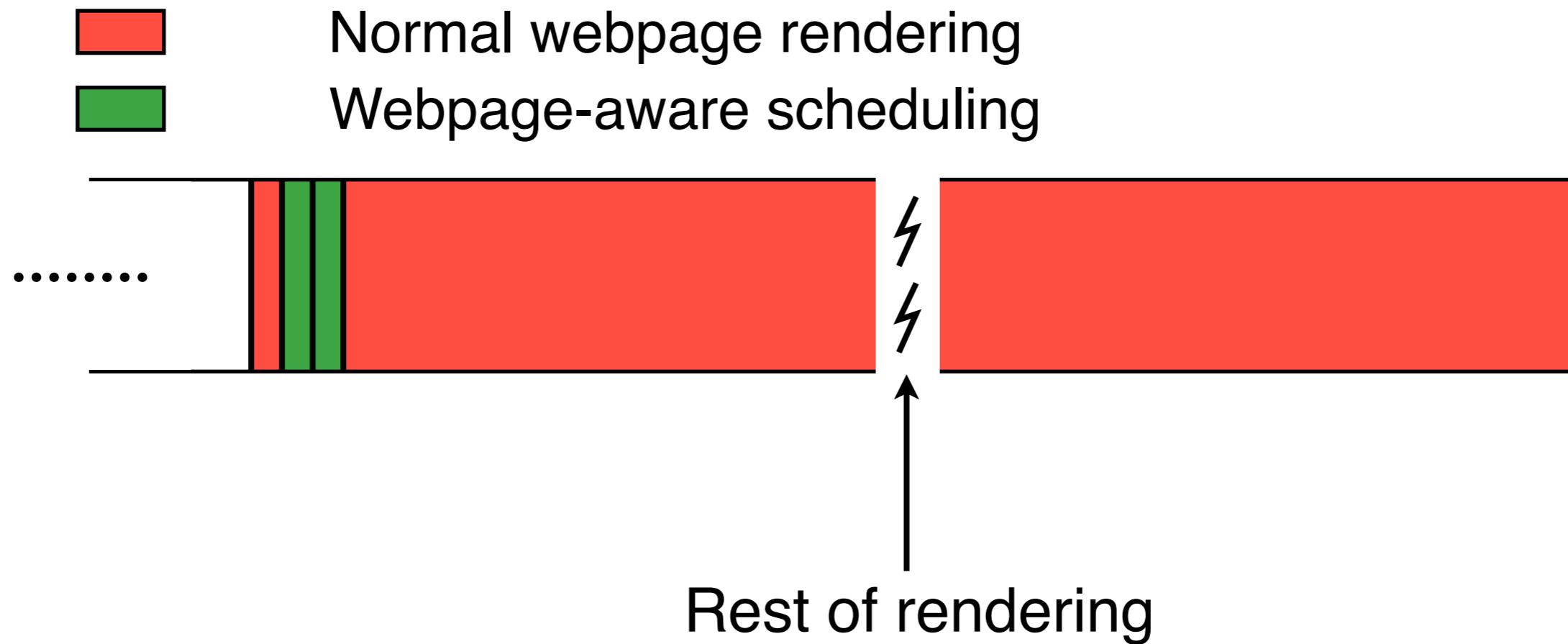
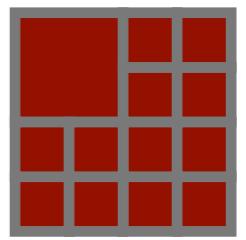


Scheduling overhead

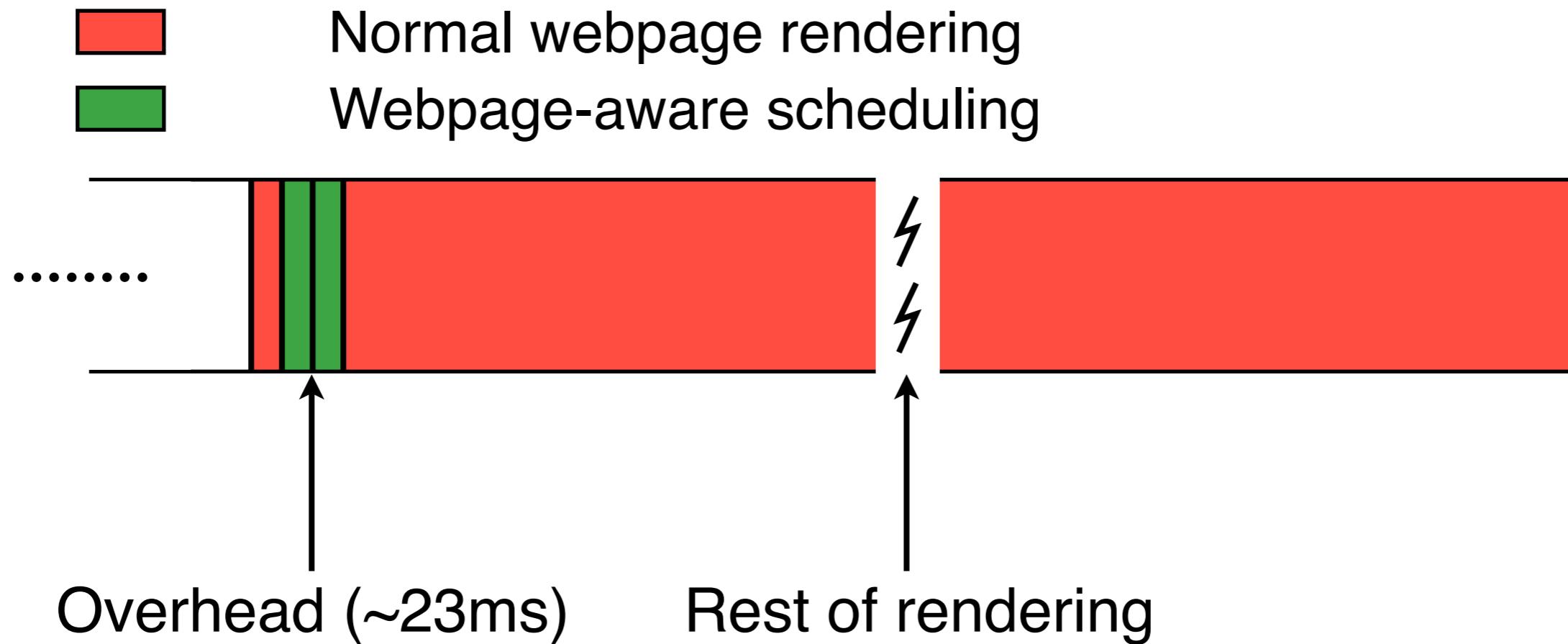
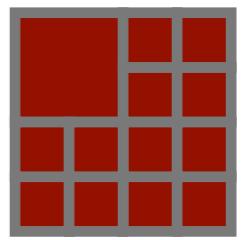
- Big/little migration: ~20ms [1]
- Frequency scaling: ~3ms considering both HW/SW time

[1] Big.LITTLE Processing with ARM Cortex-A15 & Cortex-A7. <http://goo.gl/7mgbL>

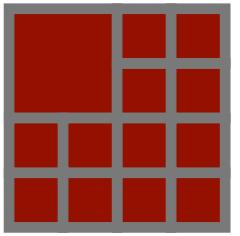
Webpage-aware Scheduling



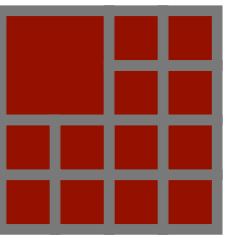
Webpage-aware Scheduling



Evaluation Methodology

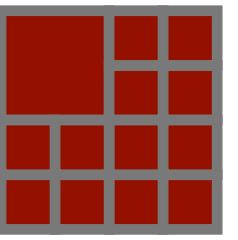


- Today's system: performance-oriented strategy, i.e., highest frequency on the big core (baseline)



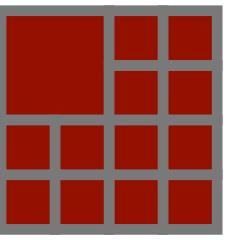
Evaluation Methodology

- Today's system: performance-oriented strategy, i.e., highest frequency on the big core (baseline)
- Our proposal: Webpage-aware scheduling (WS)



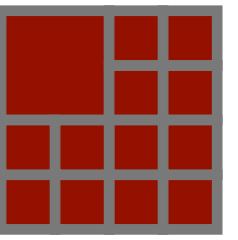
Evaluation Methodology

- Today's system: performance-oriented strategy, i.e., highest frequency on the big core (baseline)
- Our proposal: Webpage-aware scheduling (WS)
- OS DVFS strategies (OS)



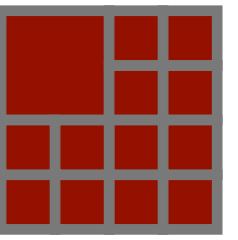
Evaluation Methodology

- Today's system: performance-oriented strategy, i.e., highest frequency on the big core (baseline)
- Our proposal: Webpage-aware scheduling (WS)
- OS DVFS strategies (OS)
 - OS-Big



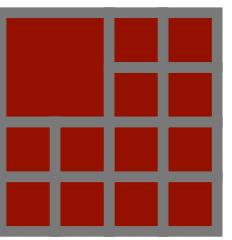
Evaluation Methodology

- Today's system: performance-oriented strategy, i.e., highest frequency on the big core (baseline)
- Our proposal: Webpage-aware scheduling (WS)
- OS DVFS strategies (OS)
 - OS-Big
 - OS-Little



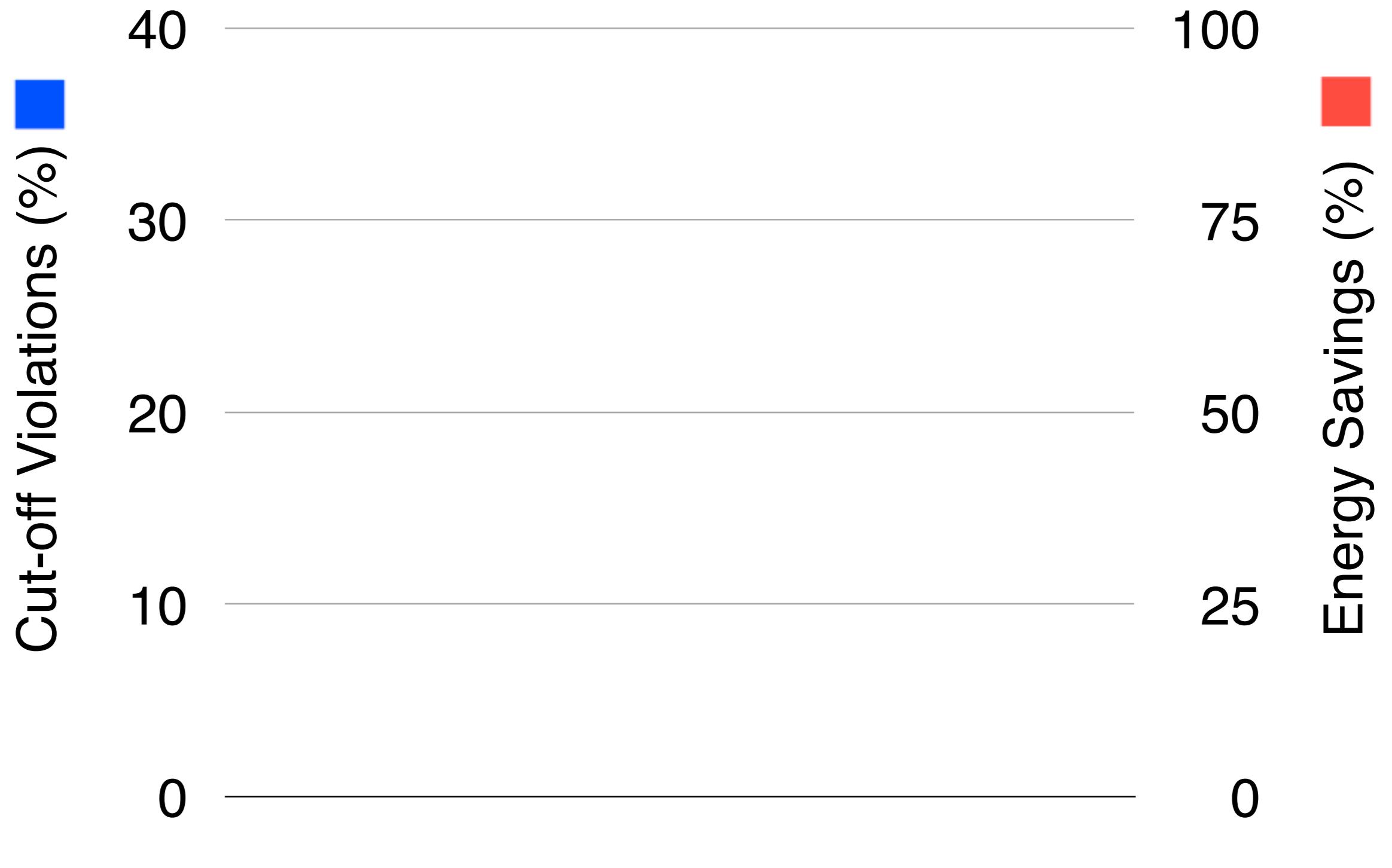
Evaluation Methodology

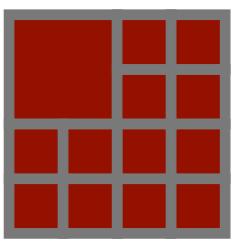
- Today's system: performance-oriented strategy, i.e., highest frequency on the big core (baseline)
- Our proposal: Webpage-aware scheduling (WS)
- OS DVFS strategies (OS)
 - OS-Big
 - OS-Little
 - (Hypothetical) OS-Big/Little



Results

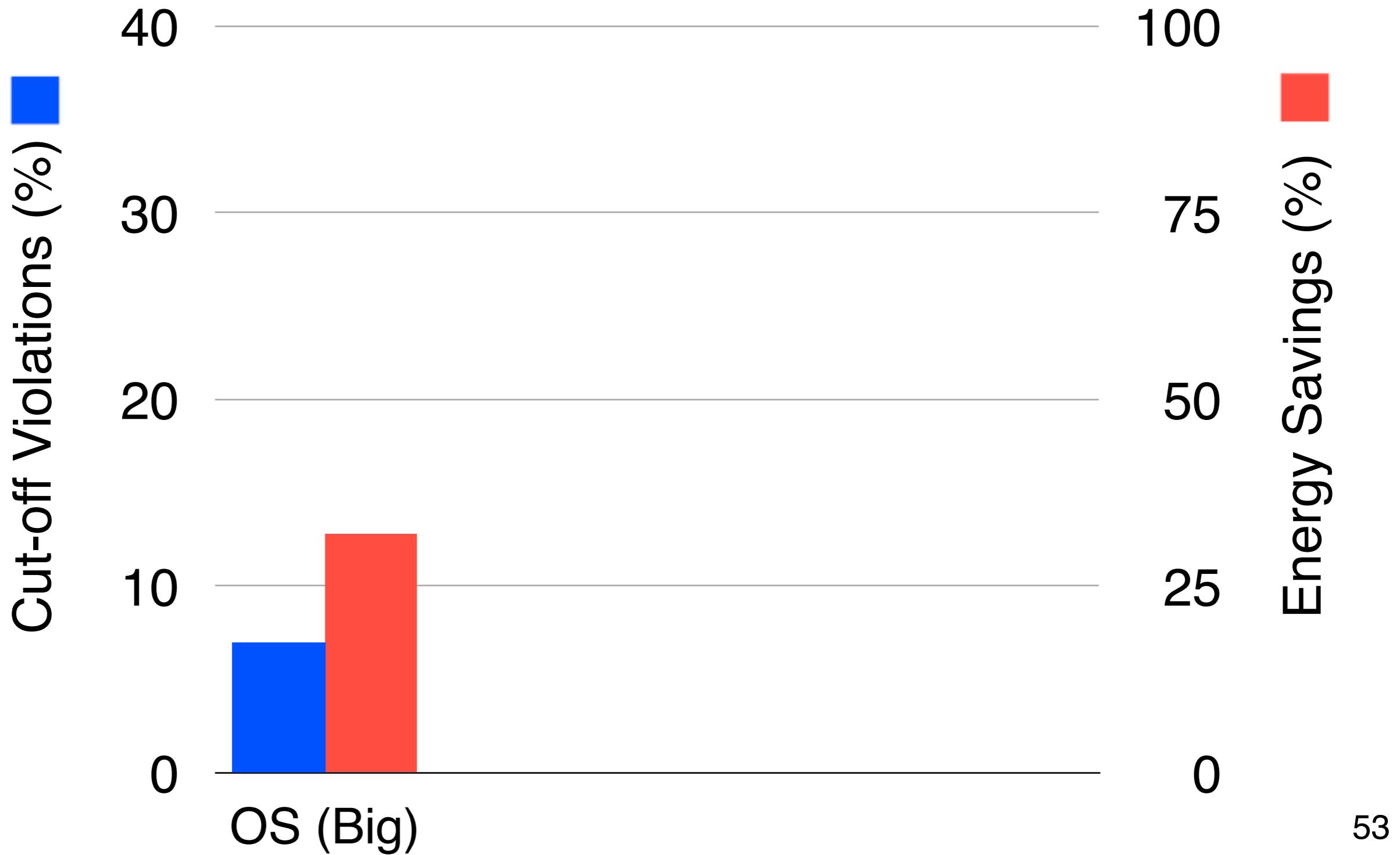
Perf-oriented strategy as the baseline

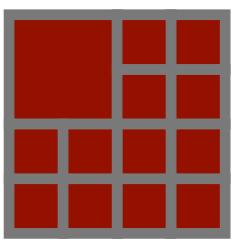




Results

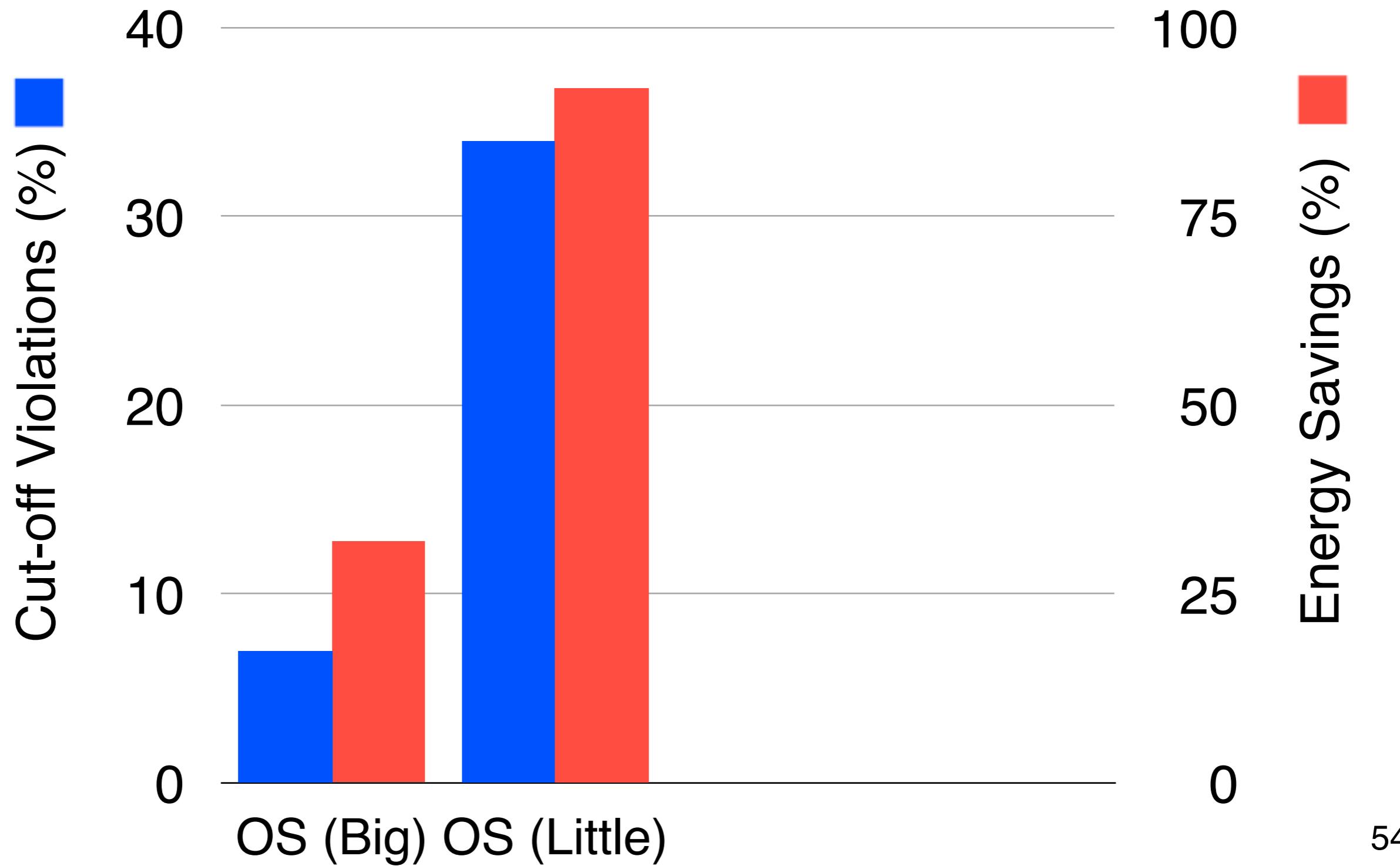
Perf-oriented strategy as the baseline



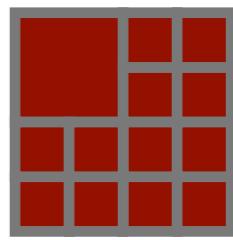


Results

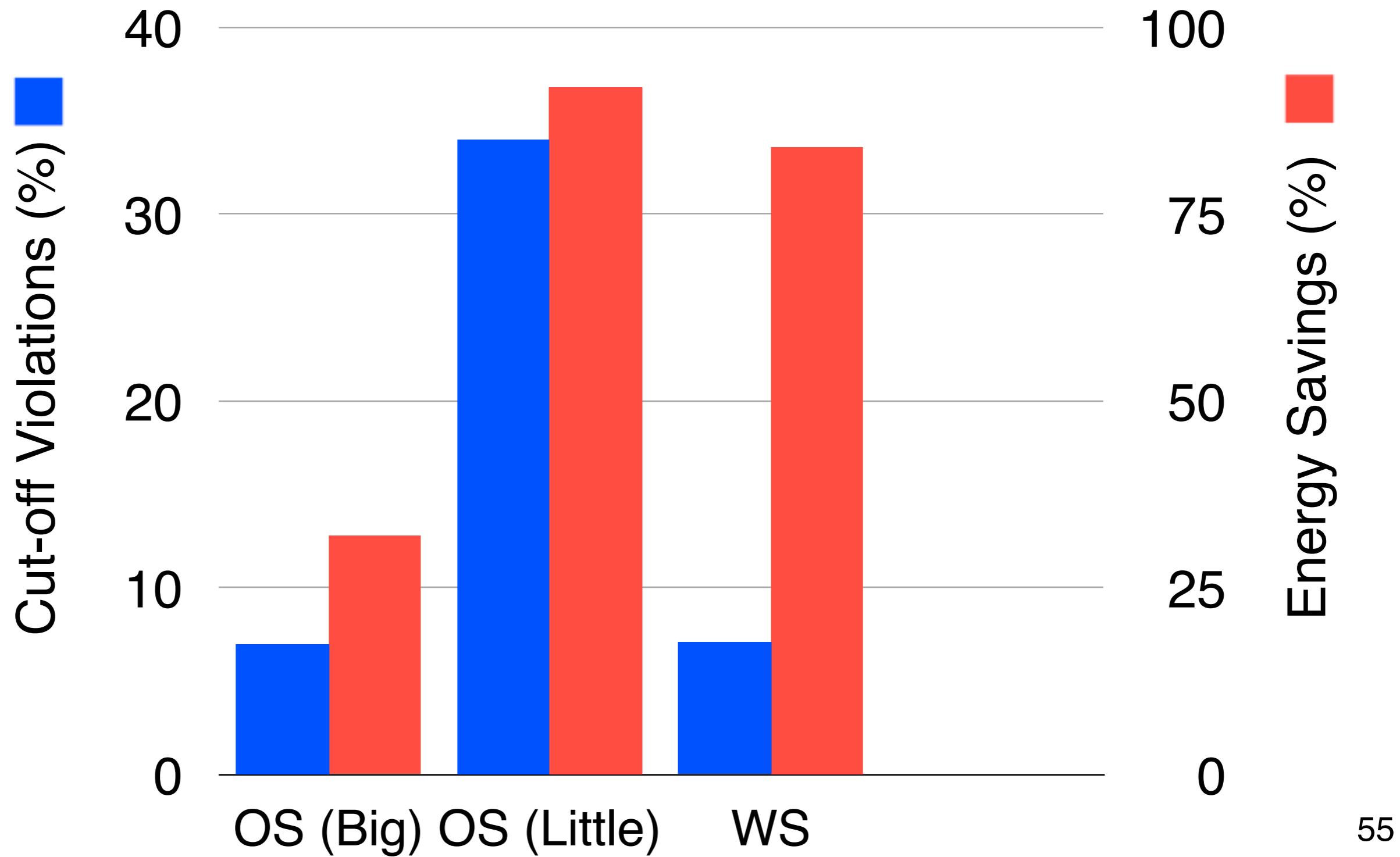
Perf-oriented strategy as the baseline



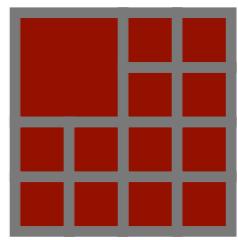
Results



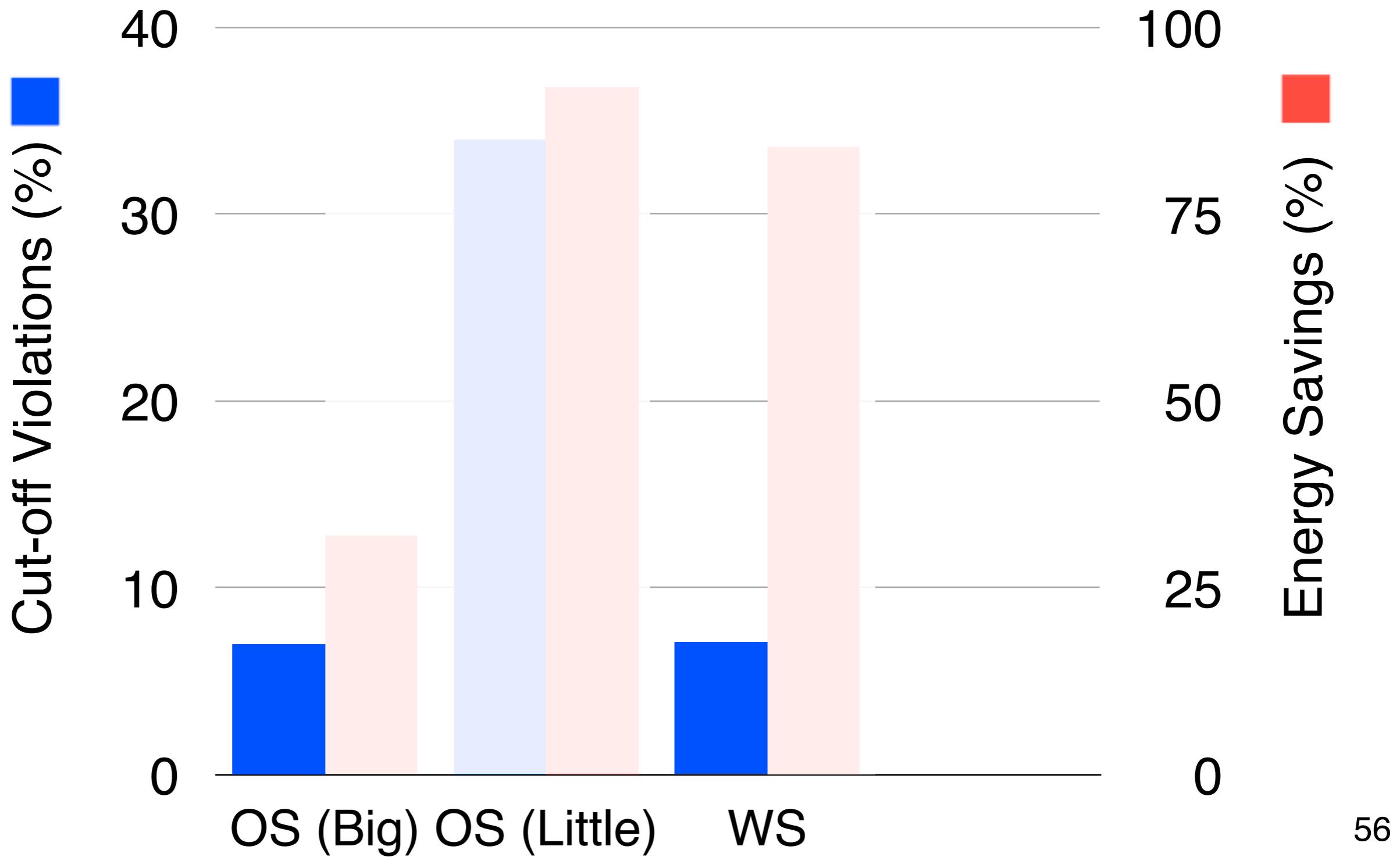
Perf-oriented strategy as the baseline



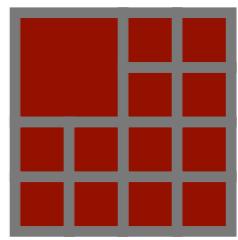
Results



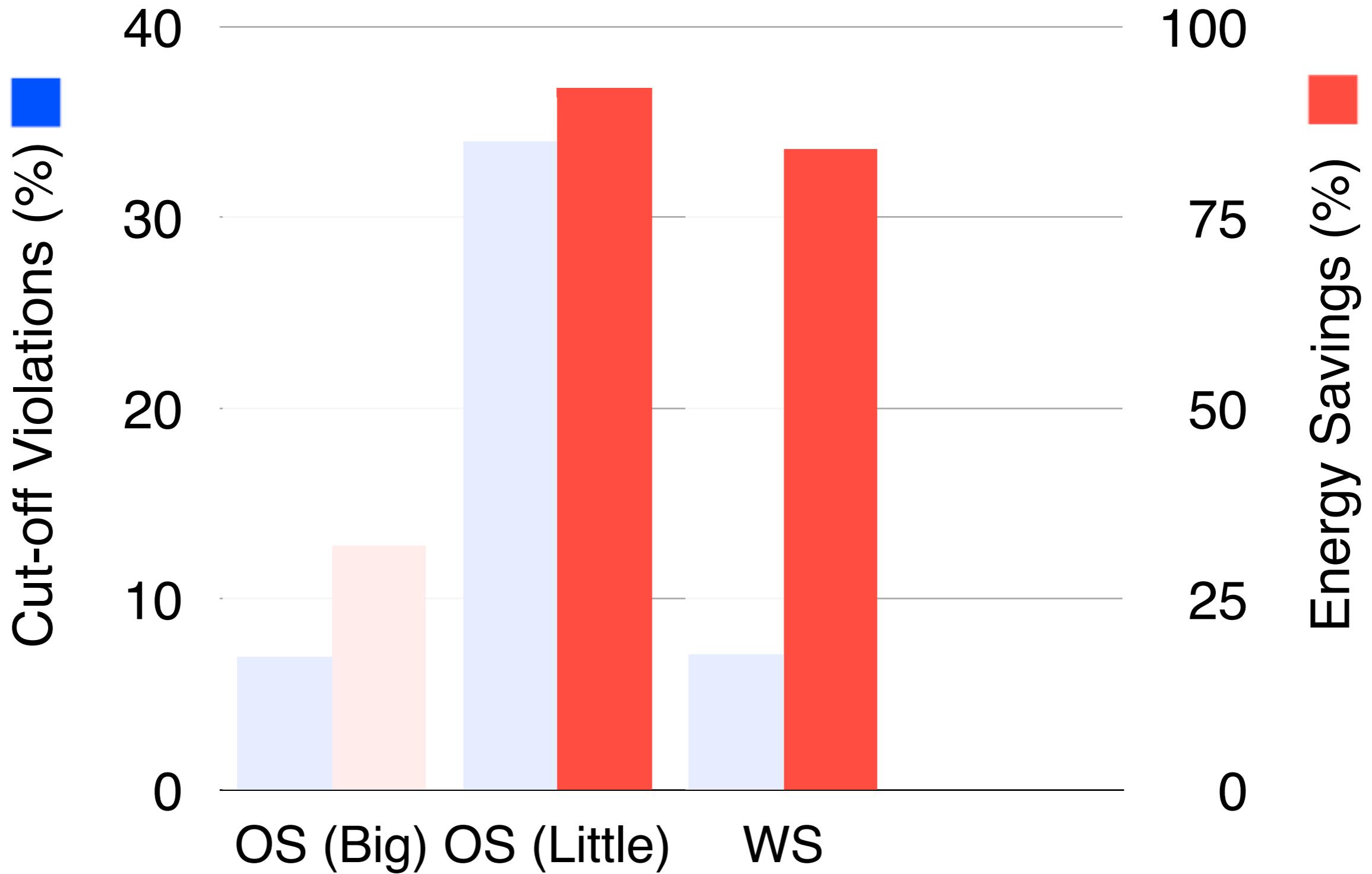
Perf-oriented strategy as the baseline



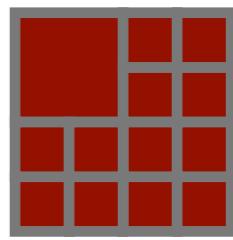
Results



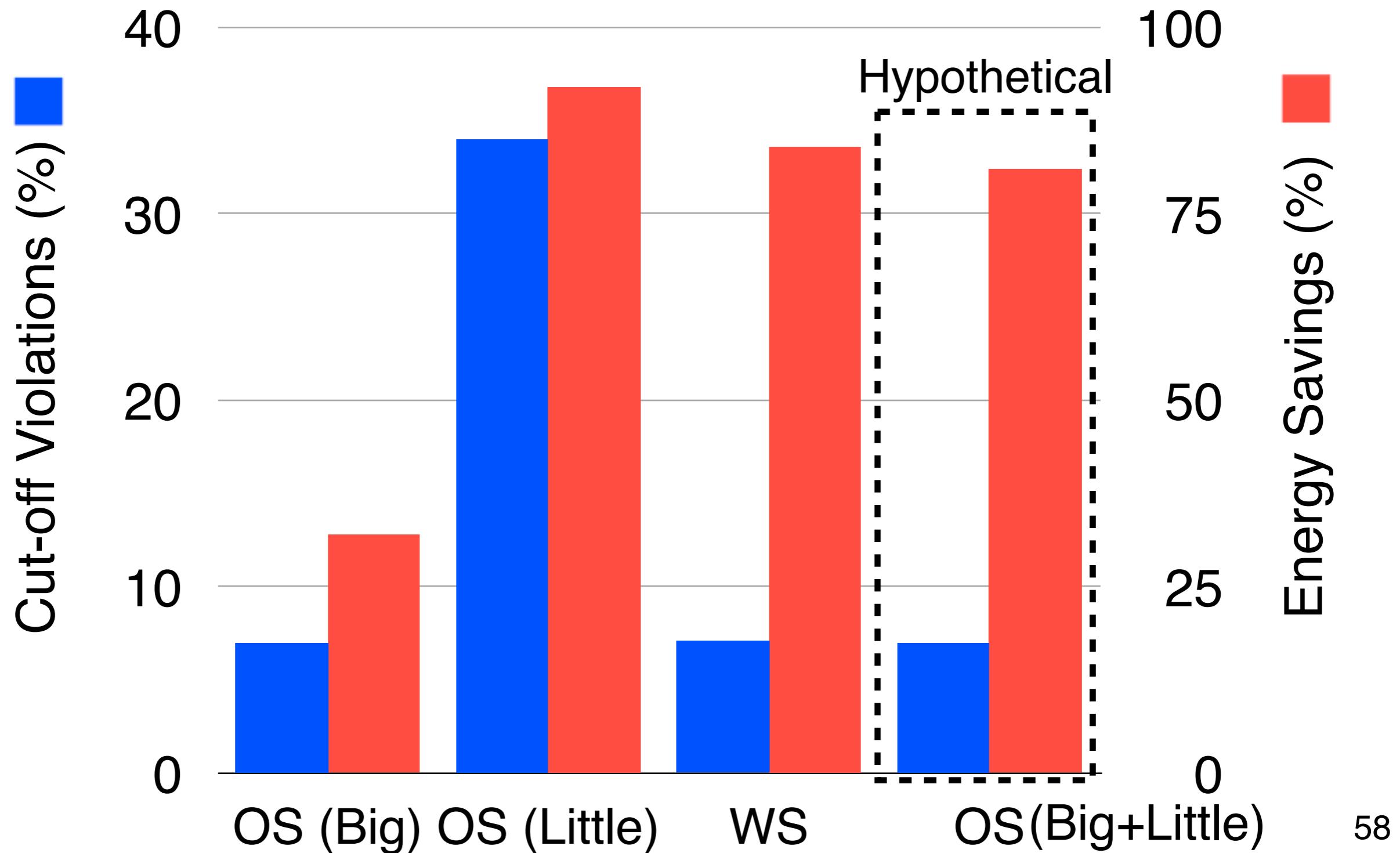
Perf-oriented strategy as the baseline



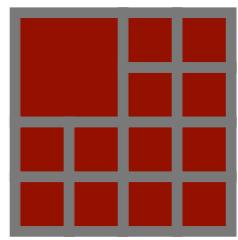
Results



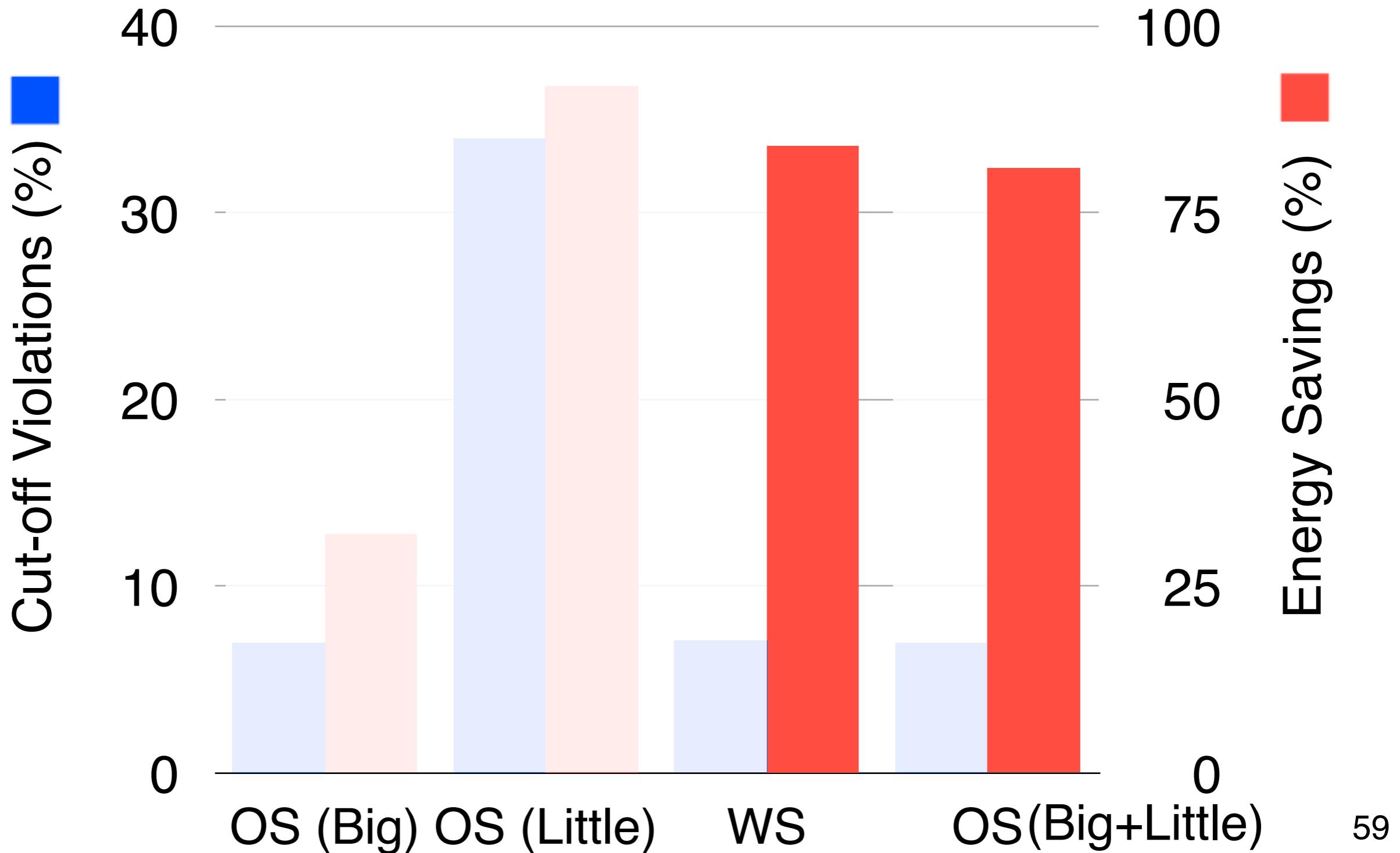
Perf-oriented strategy as the baseline



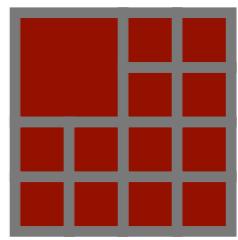
Results



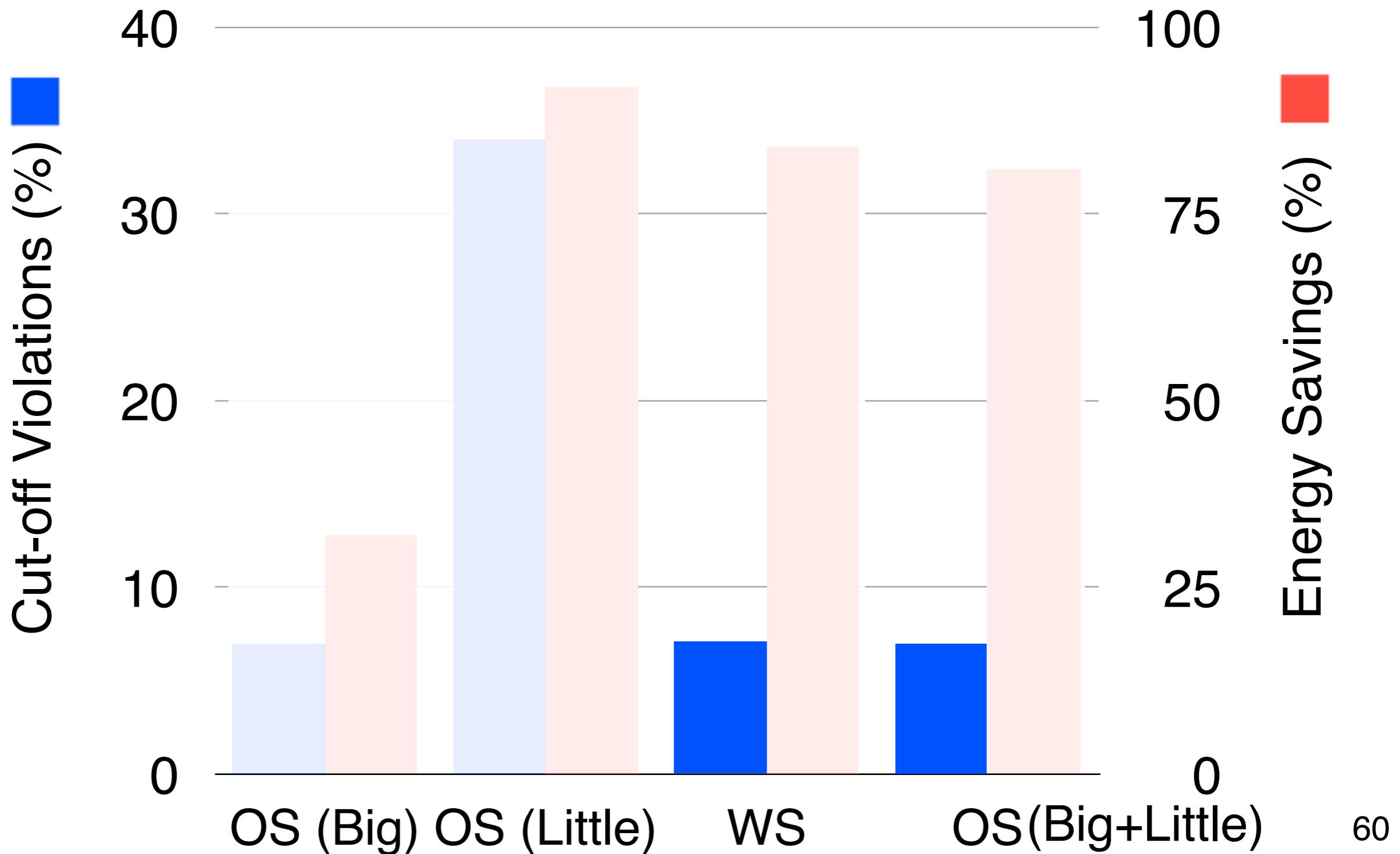
Perf-oriented strategy as the baseline



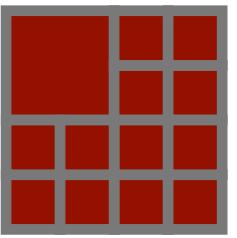
Results



Perf-oriented strategy as the baseline



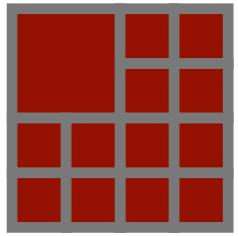
Results



Webpage-aware Scheduler

- VS. Performance-oriented strategy
 - 83% energy saving
 - 4.1% more cut-off violations

Results



Webpage-aware Scheduler

- VS. Performance-oriented strategy
 - 83% energy saving
 - 4.1% more cut-off violations
- VS. (Hypothetical) Big/Little OS DVFS
 - 8.6% energy saving with minimal additional cut-off violations
 - 4.0% performance improvement

Conclusions



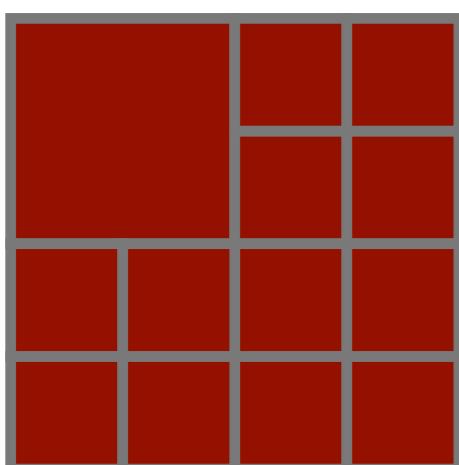
Webpage-inherent characterization

- Webpages are drastically different in load time and energy consumption



Platform-dependent load time/energy prediction

- 94.3% and 93.6% accuracy, respectively



Big/little scheduling to effectively utilize the hardware resources

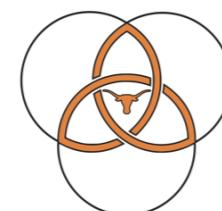
- Significant energy saving over the performance-oriented strategy
- Improve energy and performance over the Big/Little OS DVFS strategy



High-Performance and Energy-Efficient Mobile Web Browsing on Big/Little Systems

Yuhao Zhu (朱禹皓)

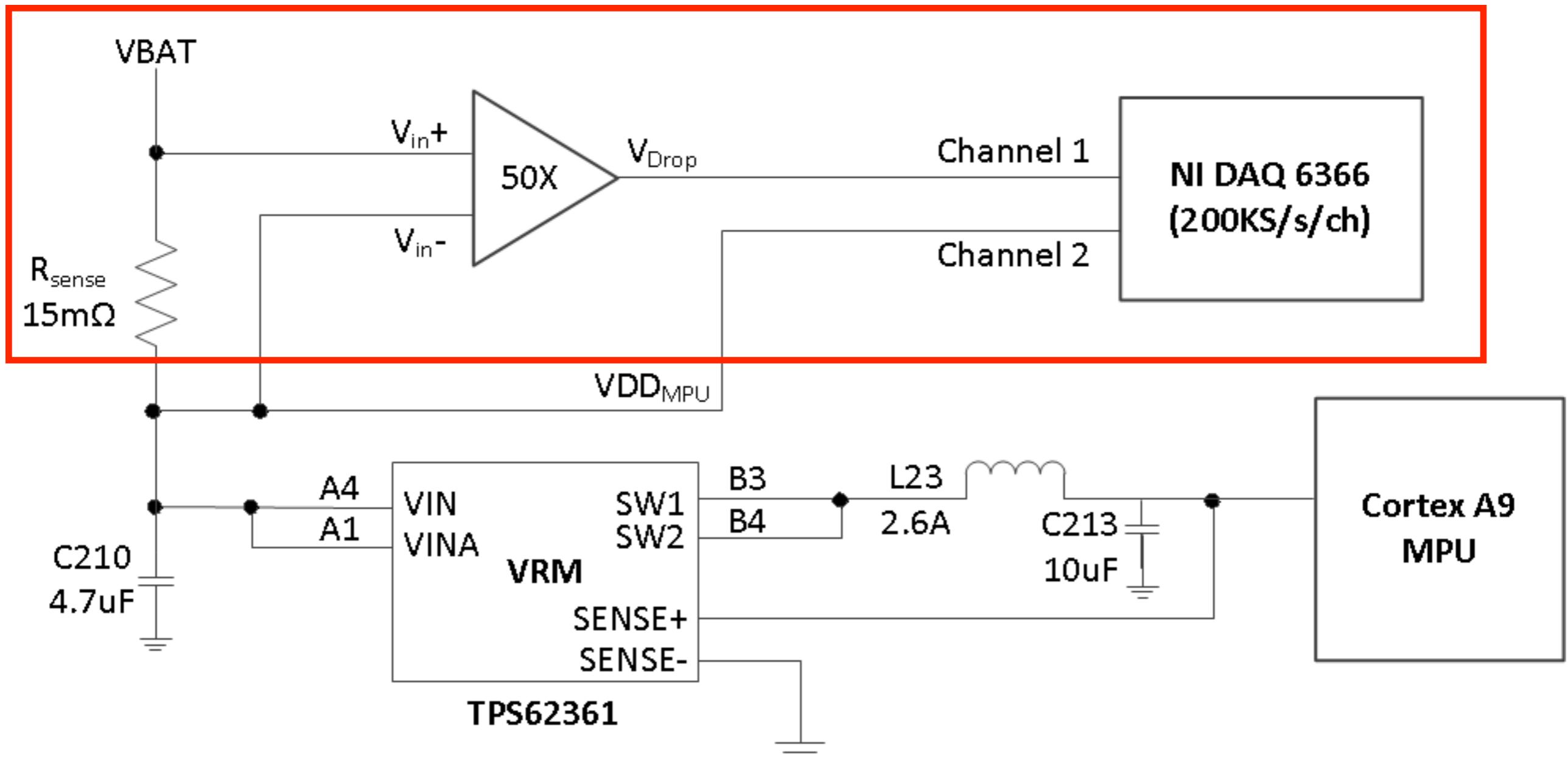
Vijay Janapa Reddi



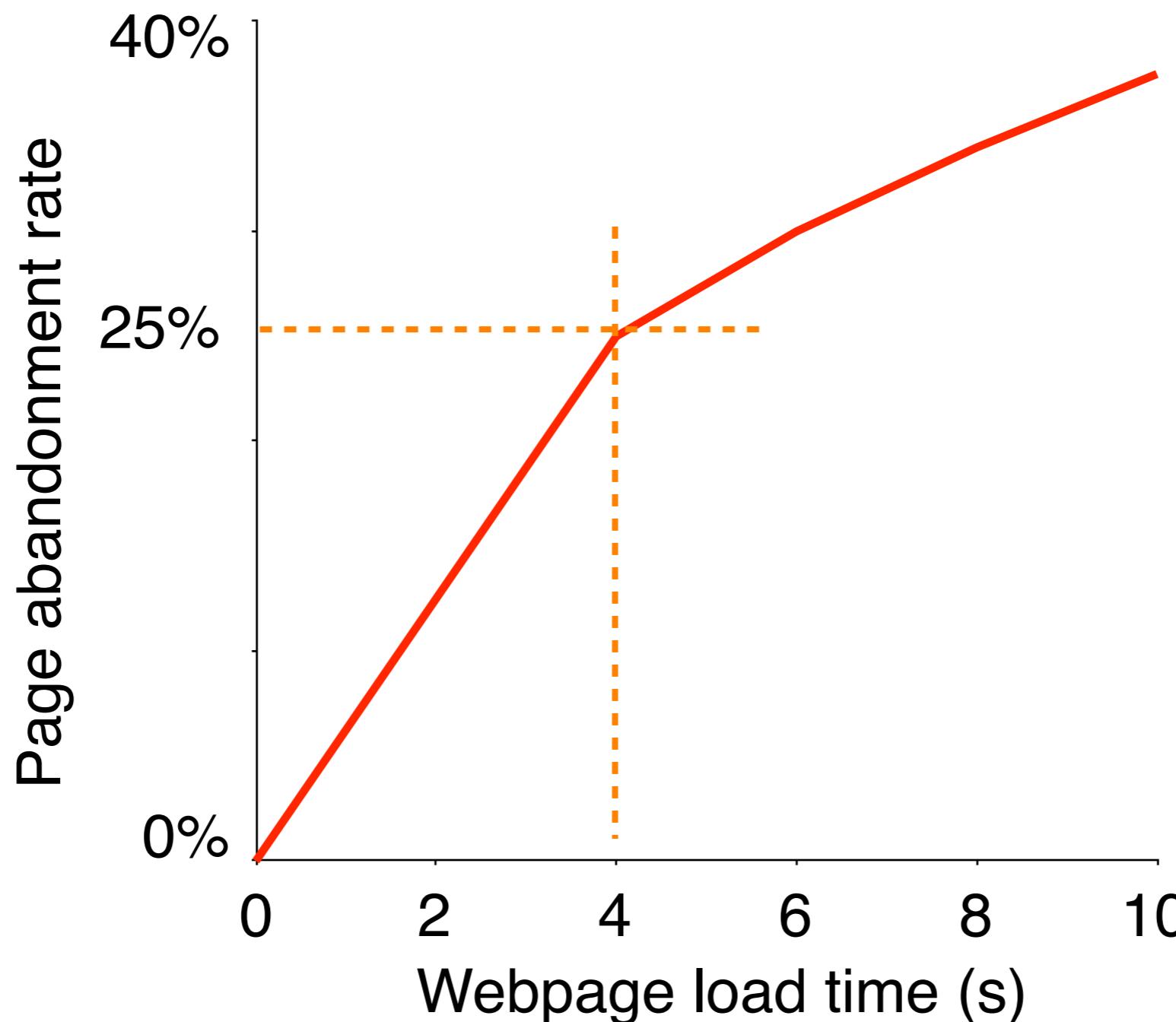
Trinity Research Group
The University of Texas at Austin

Backup Slides

Hardware Setup

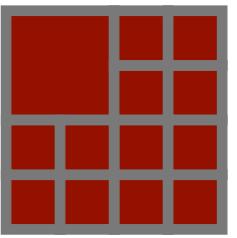


Page Abandonment



Source: Reproduced from Kissmetrics, 2011

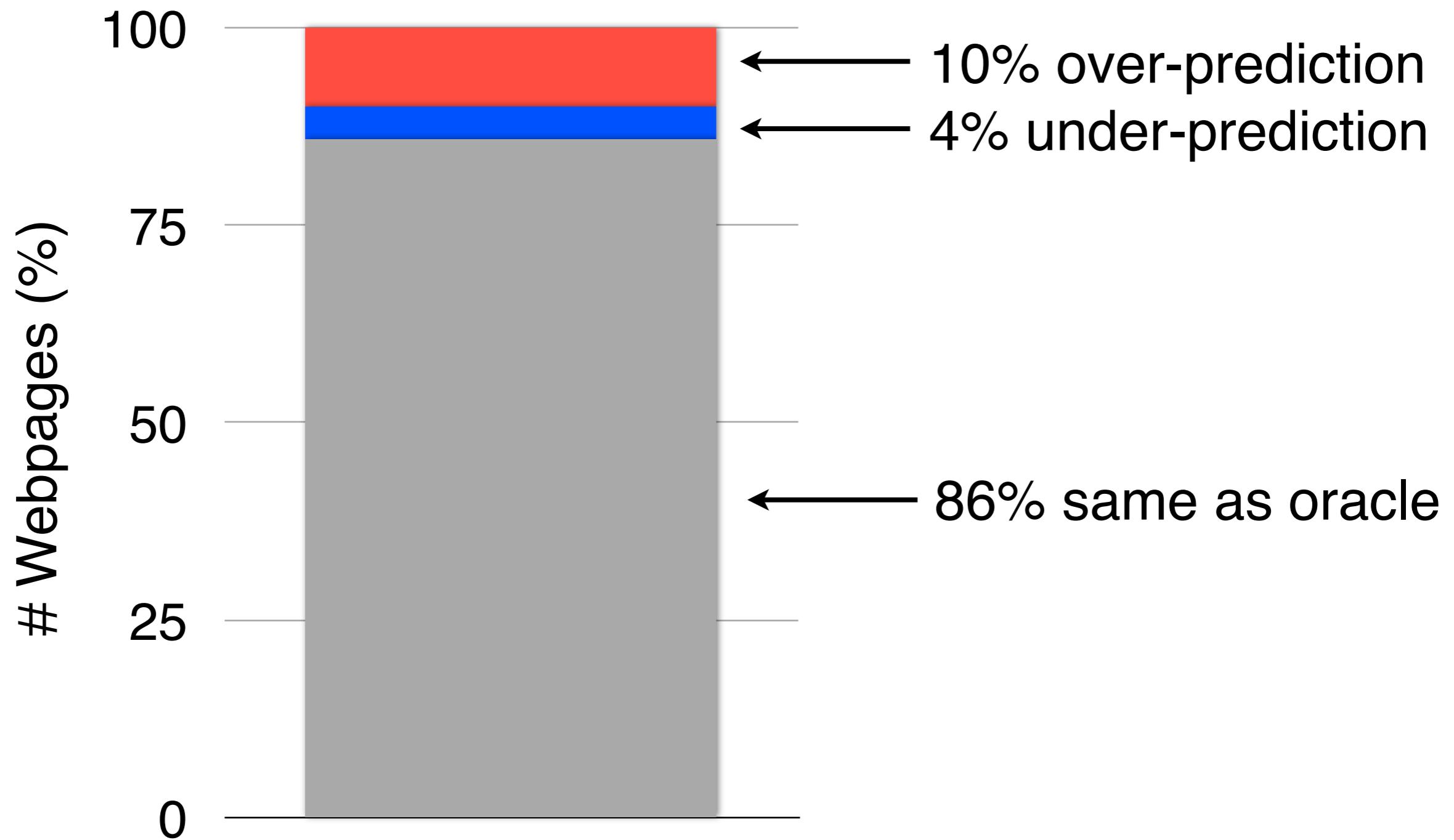
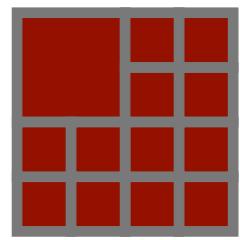
Oracle Analysis



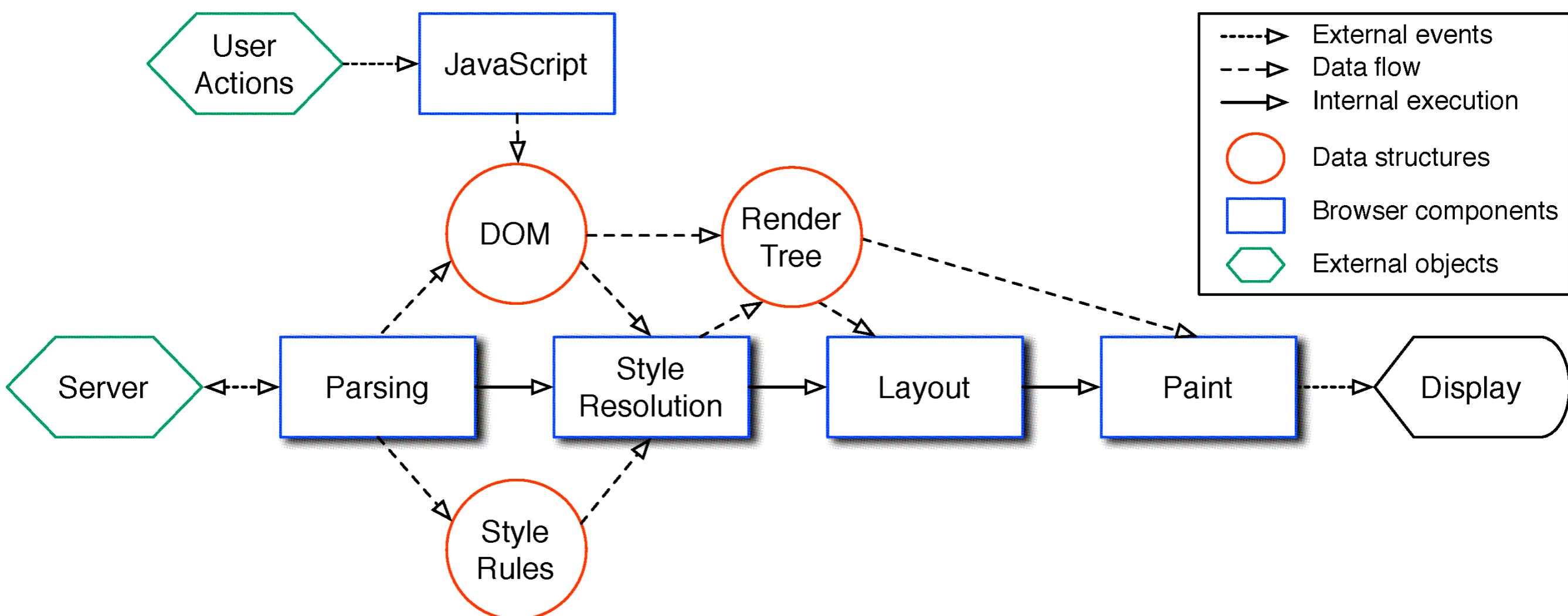
Oracle Scheduler

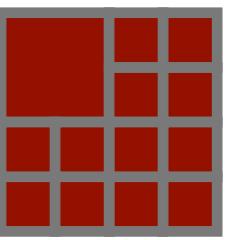
- ~80% average energy saving (over perf.-oriented mode)
- 3.5% cut-off violations

Oracle Analysis

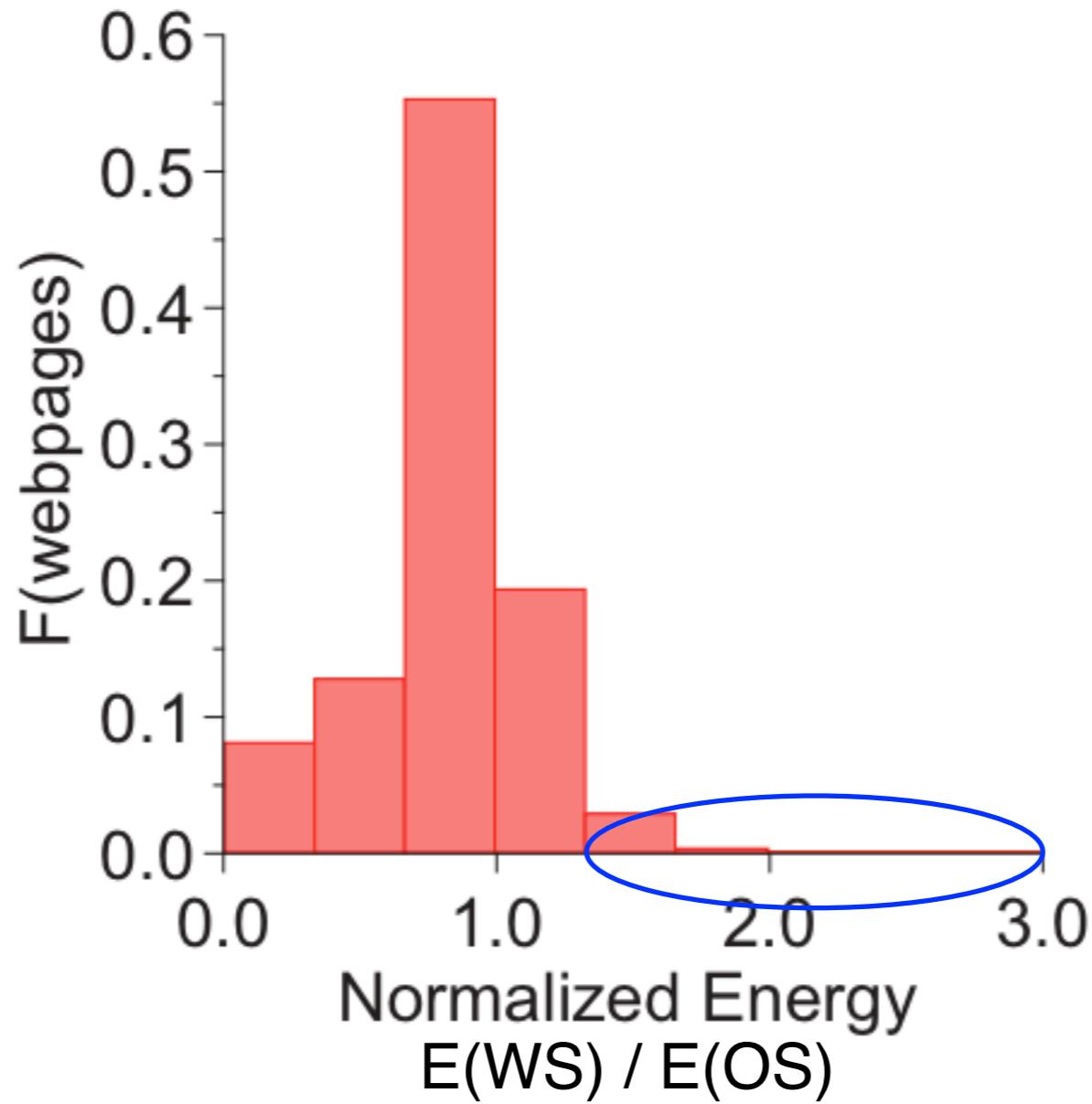


Web Browser Anatomy

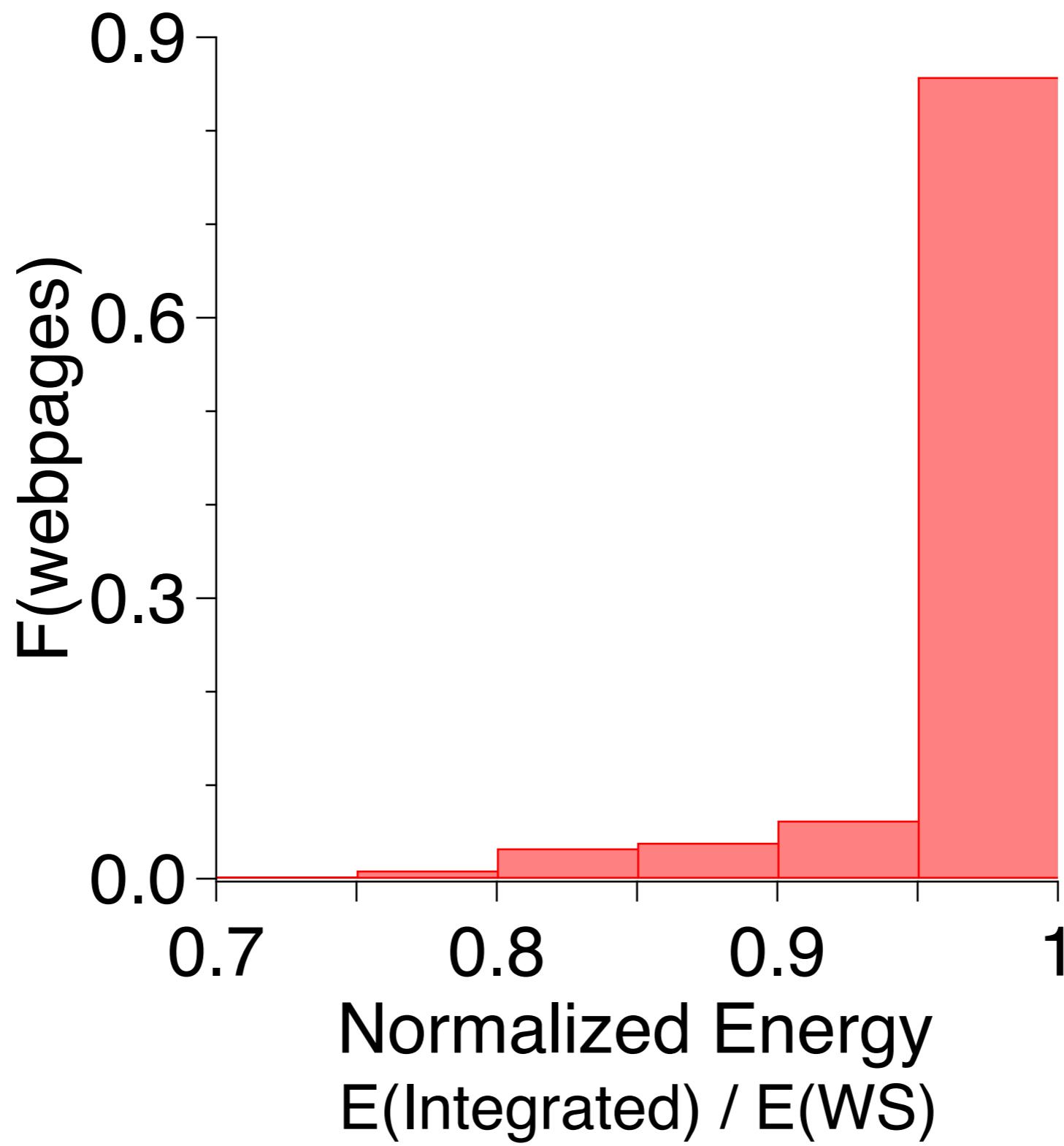




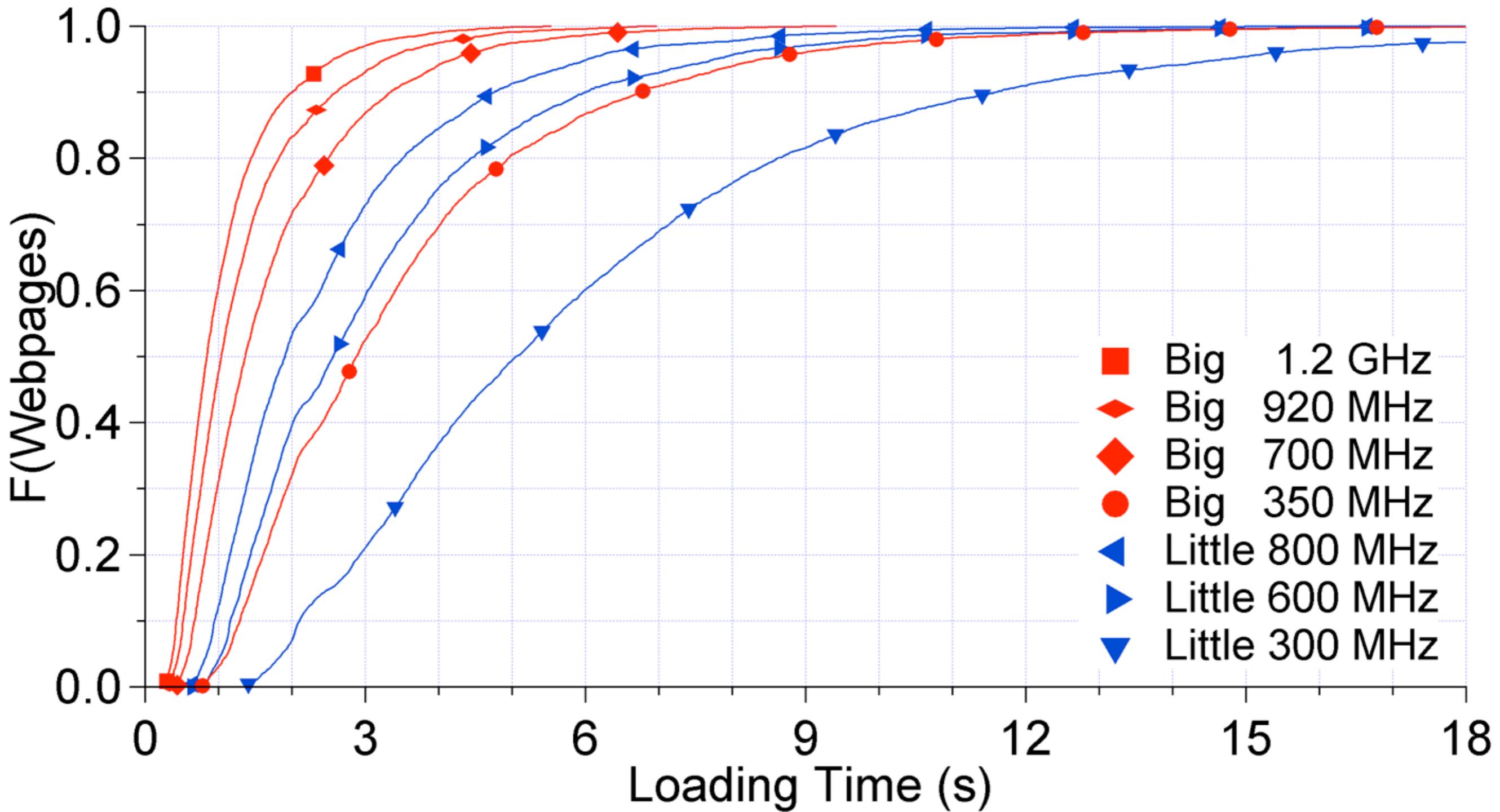
Energy Saving



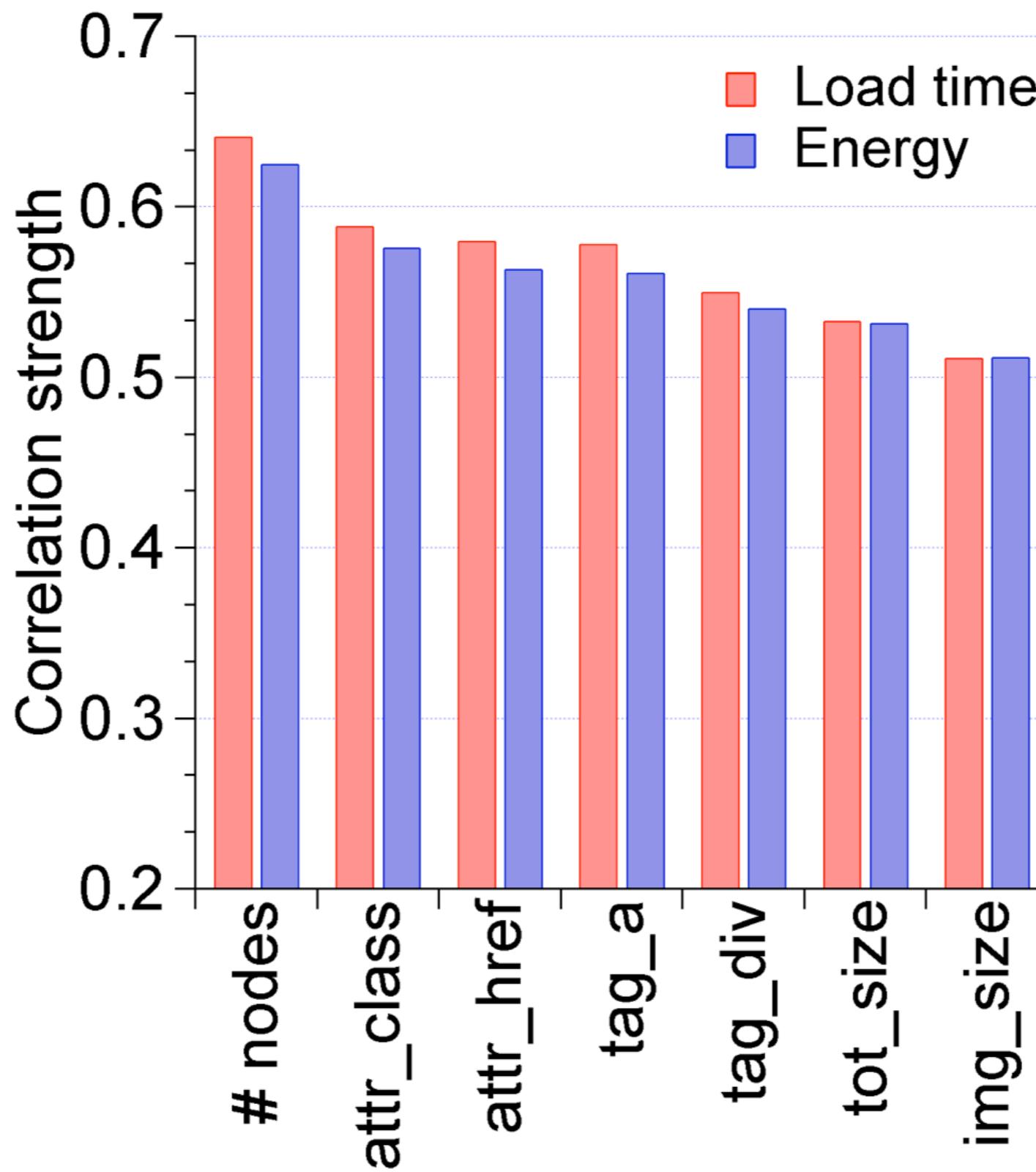
Integrated Scheduler



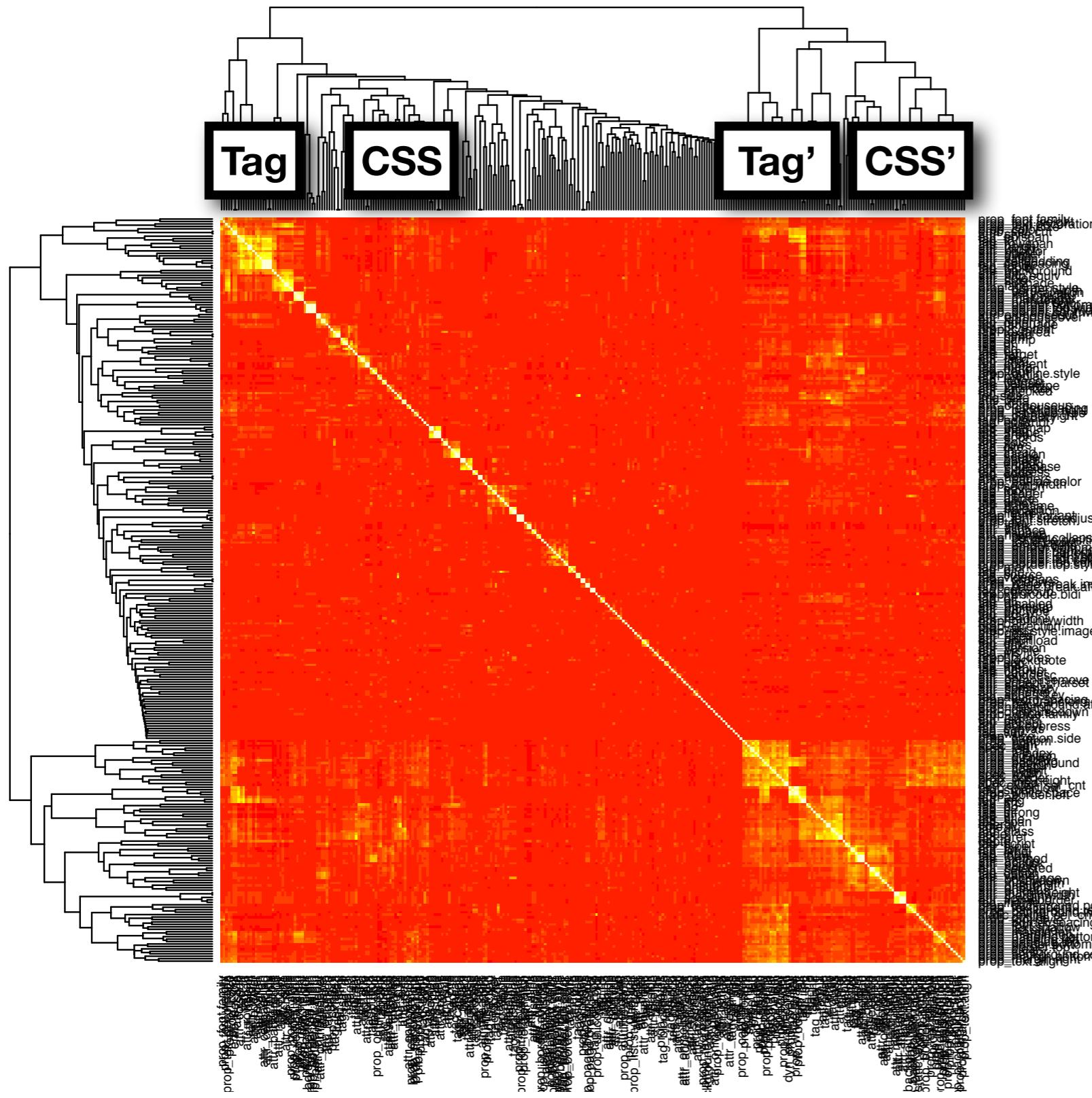
Cut-off Distributions



Predictor Strength

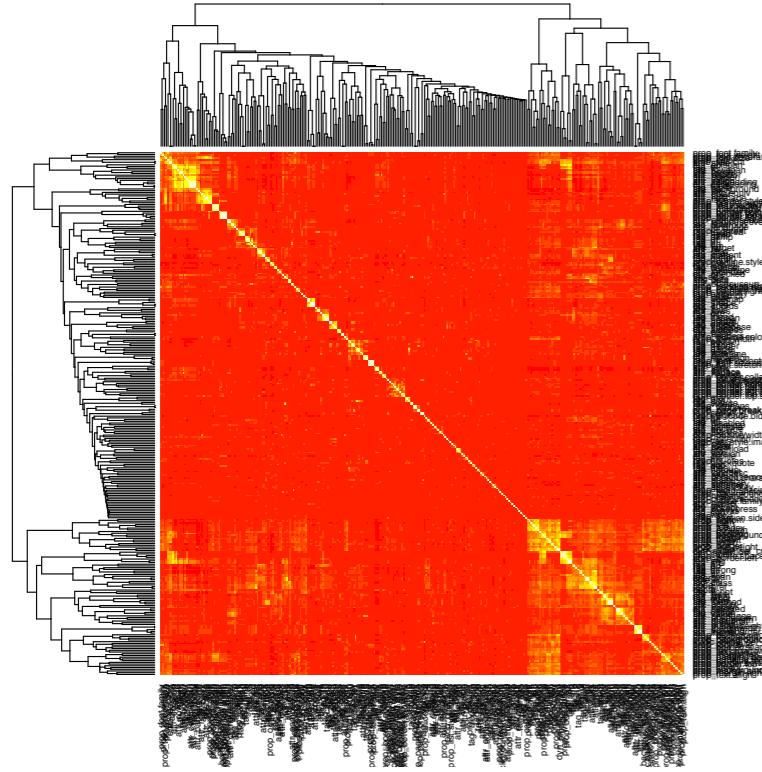


Feature Pruning

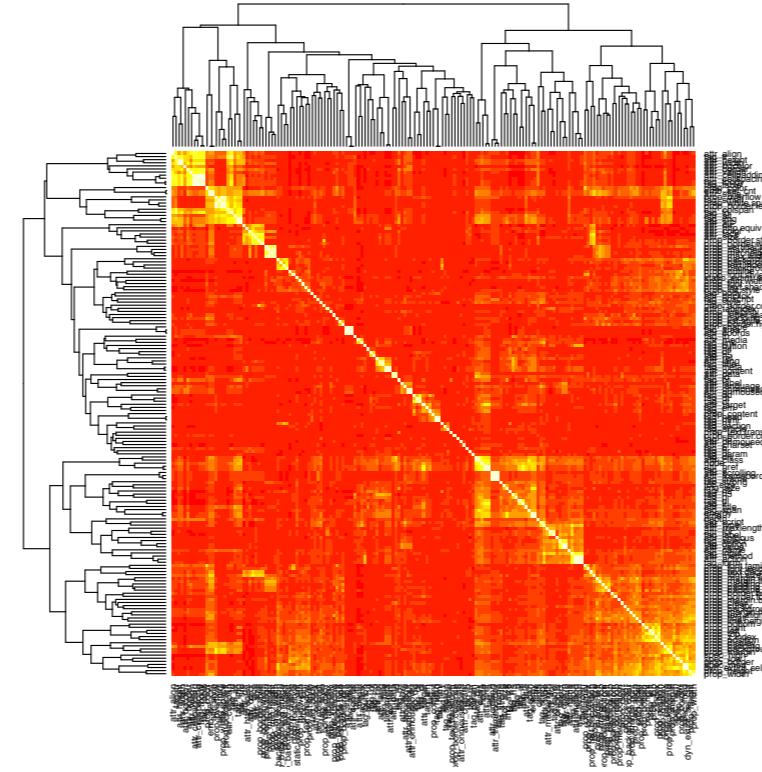


1. We find several correlations between the different features
 2. Some correlations are stronger than others

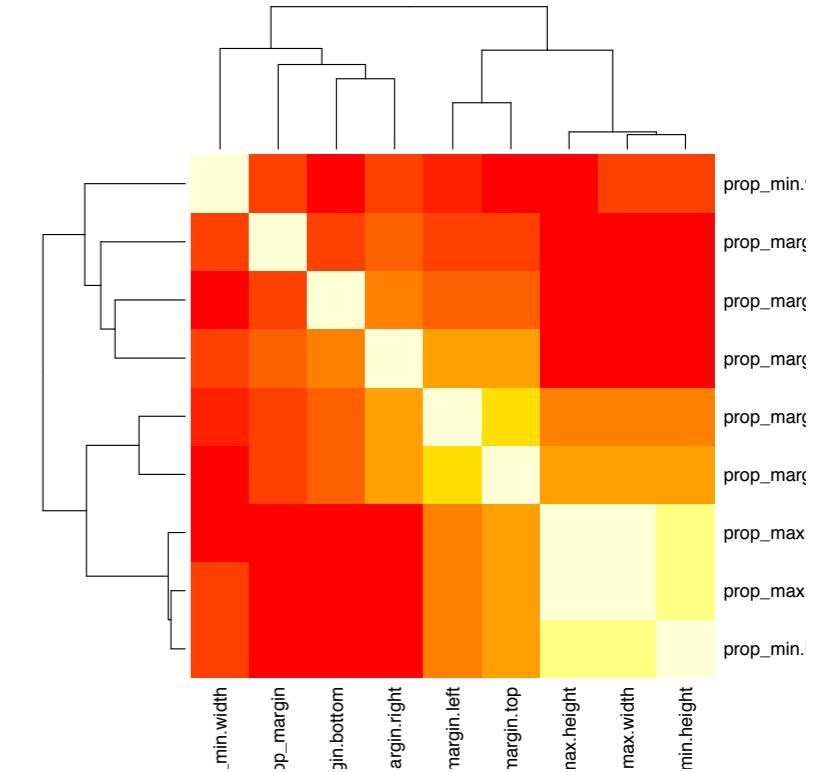
Feature Pruning



369 features

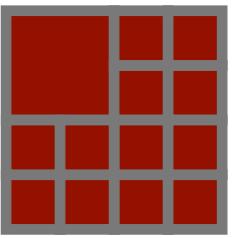


167 features

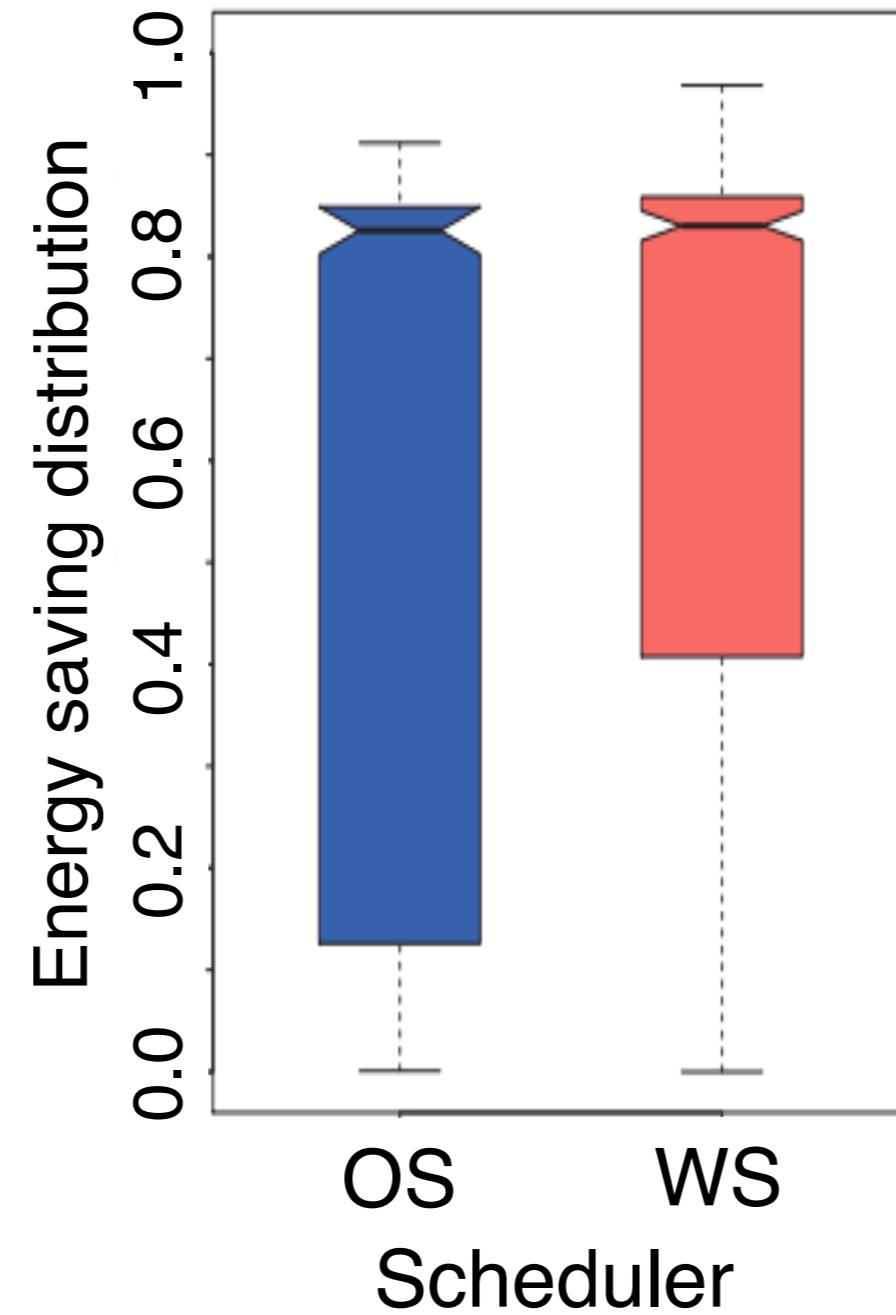
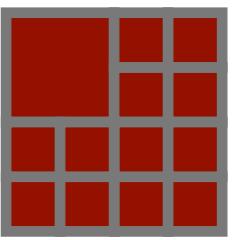


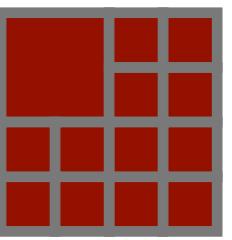
9 features

Energy Saving

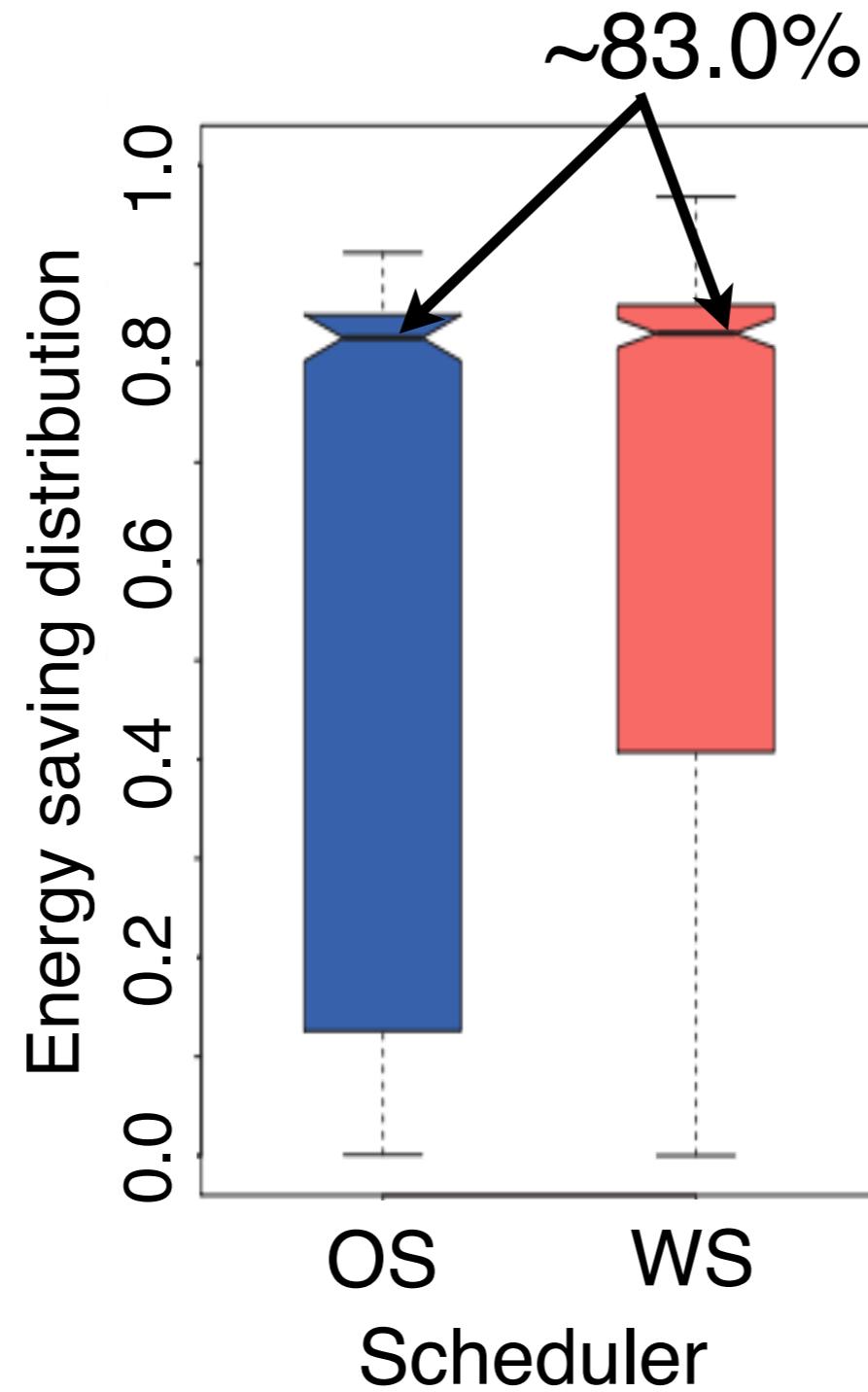


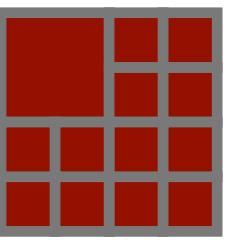
Energy Saving



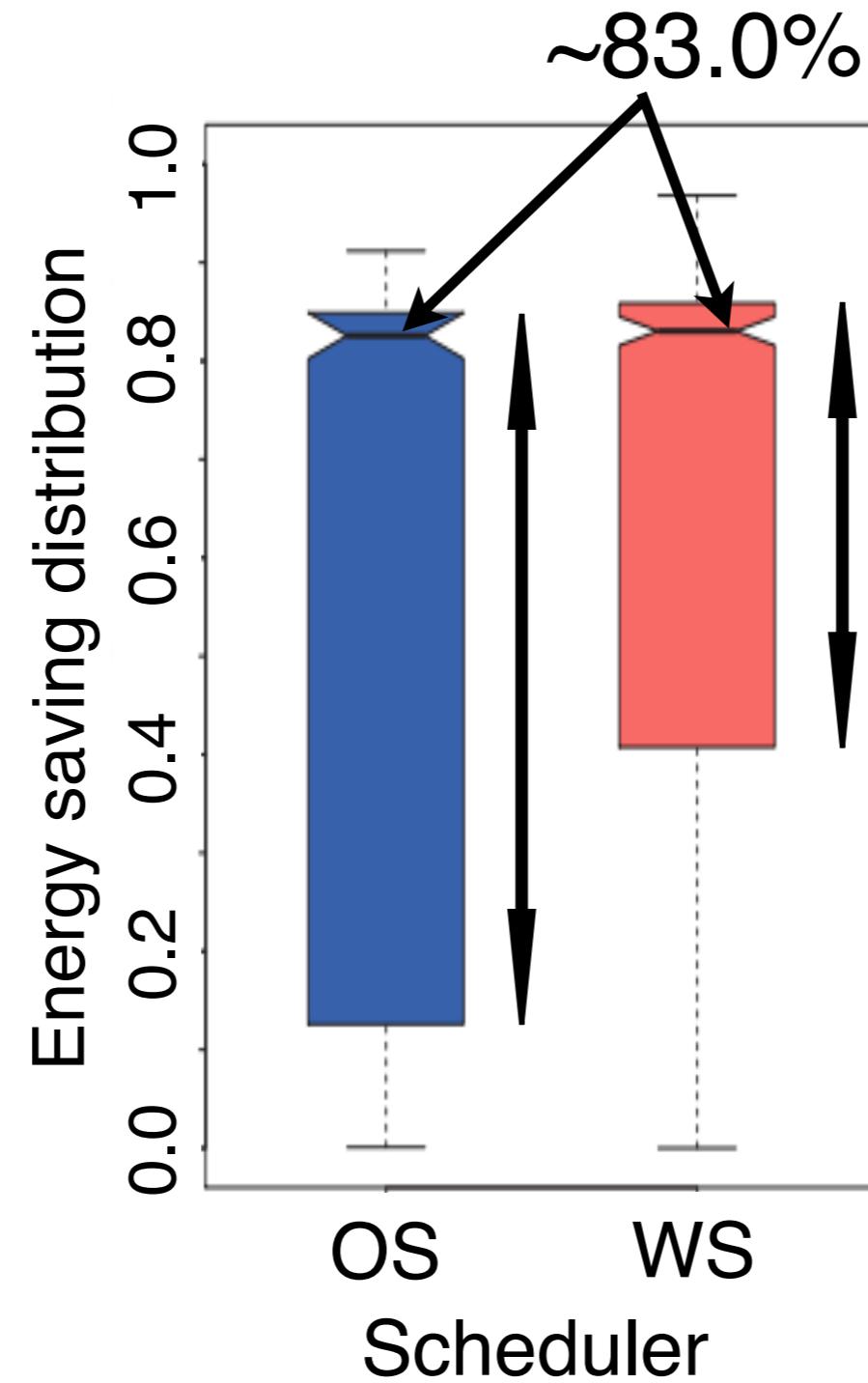


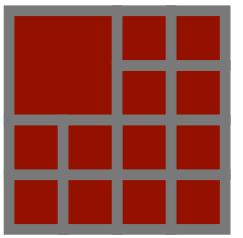
Energy Saving





Energy Saving





ARM's Big/Little

