Compiler and Architecture Design Group
Institute for Computing Systems Architecture
University of Edinburgh, United Kingdom

ARM

Measuring QoE of Interactive Workloads and Characterising Frequency Governors on Mobile Devices

IISWC 2014

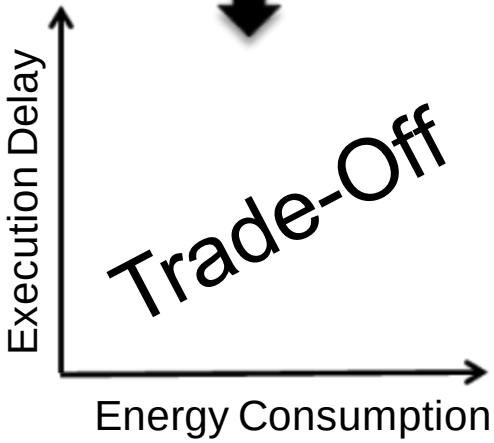
Volker Seeker
Pavlos Petoumenos, Hugh Leather, Björn Franke

Problem Statement

- heterogeneous workload
- interactive workload
- battery constraints
- ...

Responsive

Energy Efficient

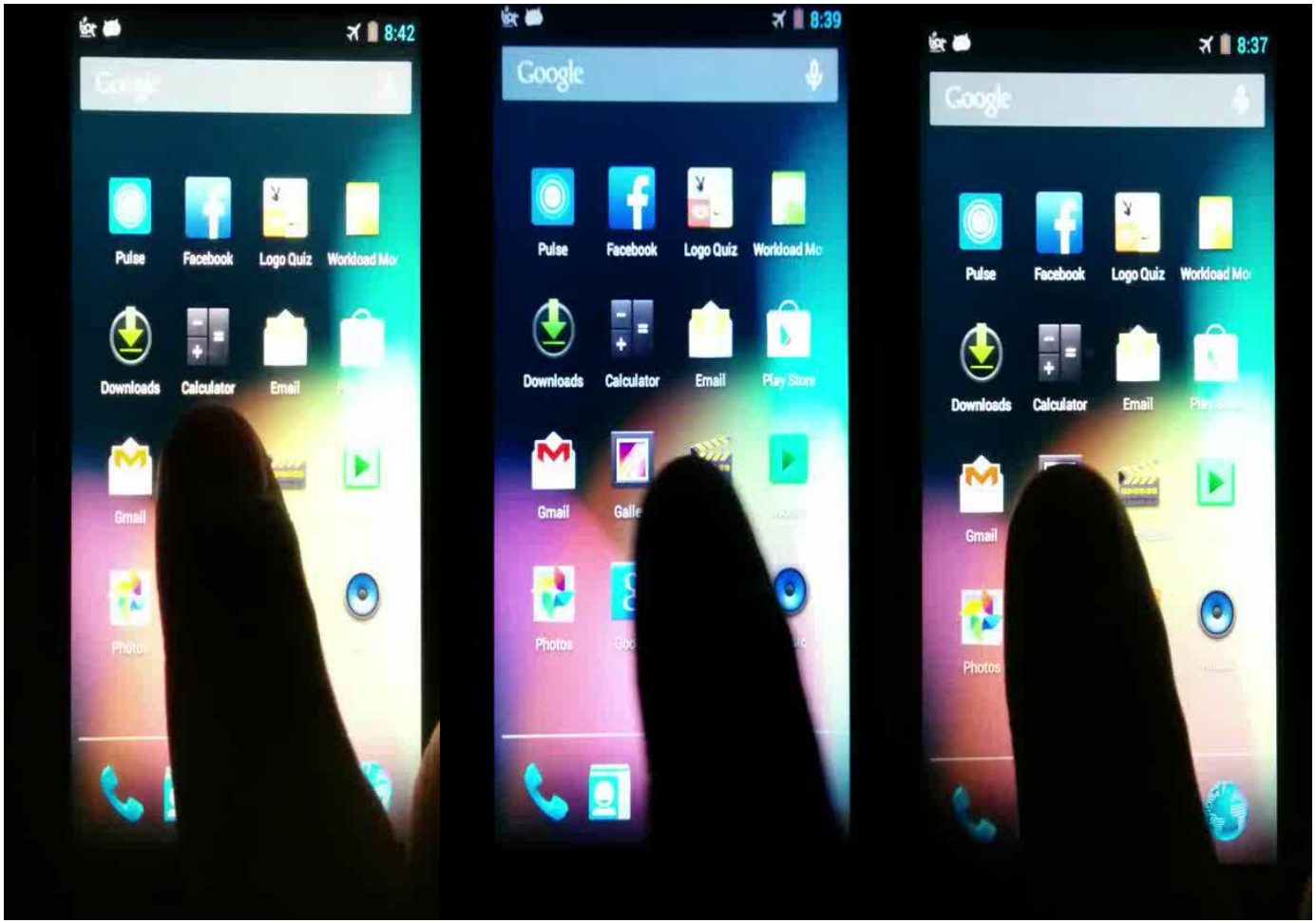


How can I measure satisfaction?

Satisfied User

Workload Scenario

Frequency Governor



0.3 GHz **1.2 GHz** **2.2 GHz**

Workload Scenario

too slow | **sweet spot** | **wasting energy**

0.3 GHz | **1.2 GHz** | **2.2 GHz**

Frequency Governor

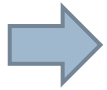
How do we find the sweet spot?

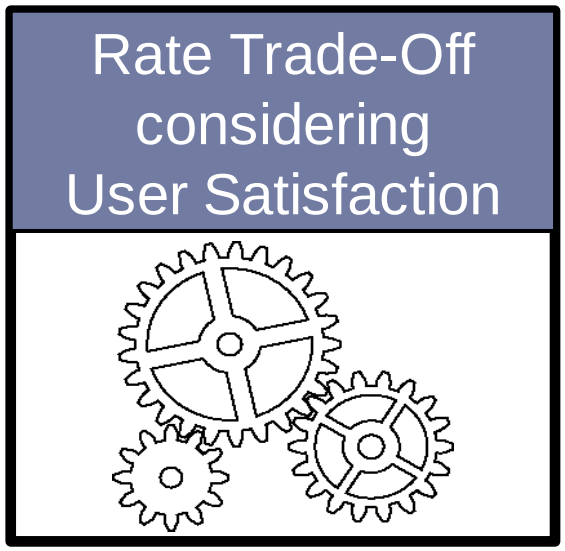
Consider the User's Perspective !

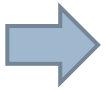
Questionaires are cost intensive

What we need!

Interactive Mobile Workload

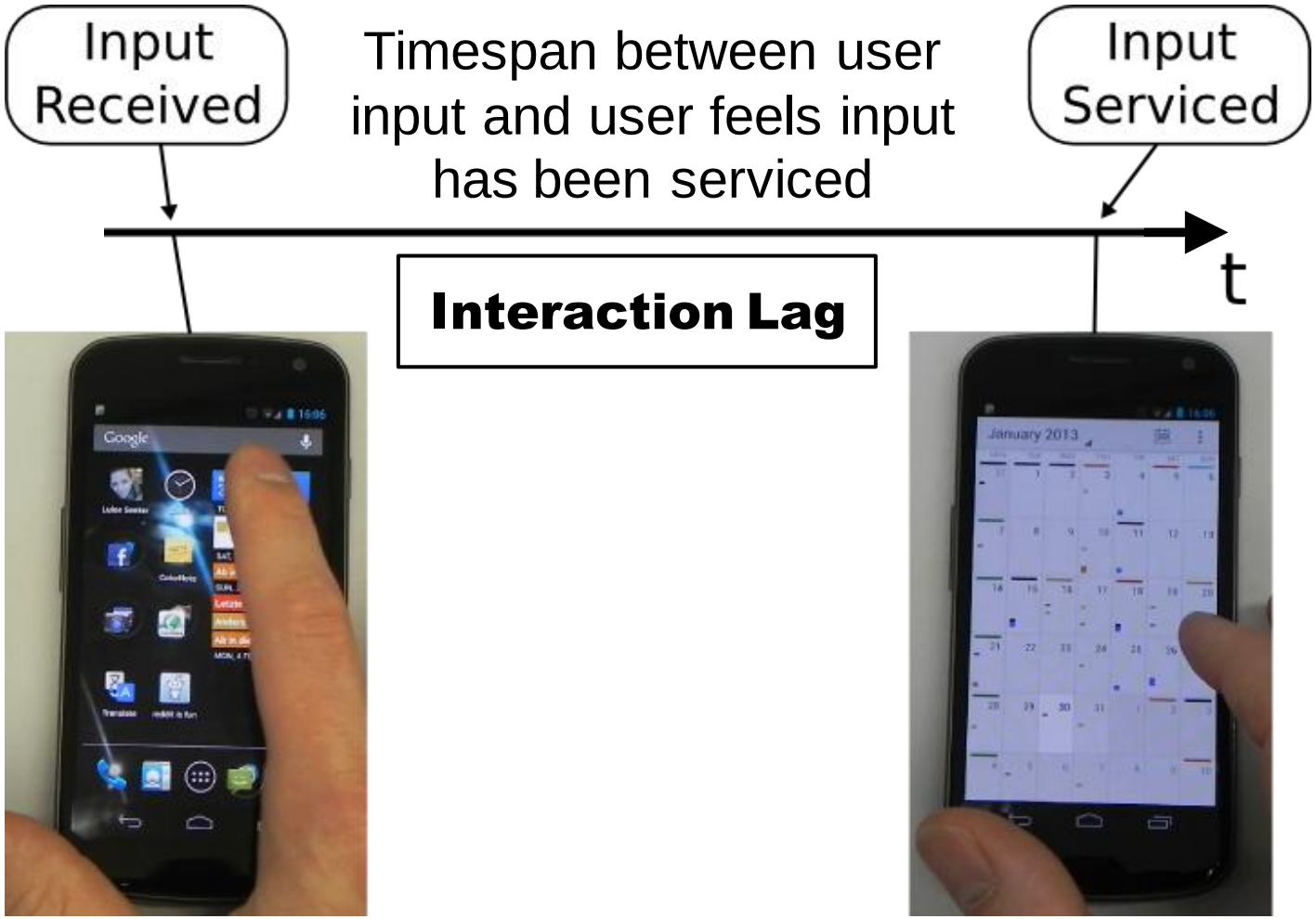






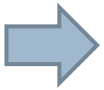
Distance to sweet spot

What Does The User Care About?



Research Goals

Interactive Mobile Workload



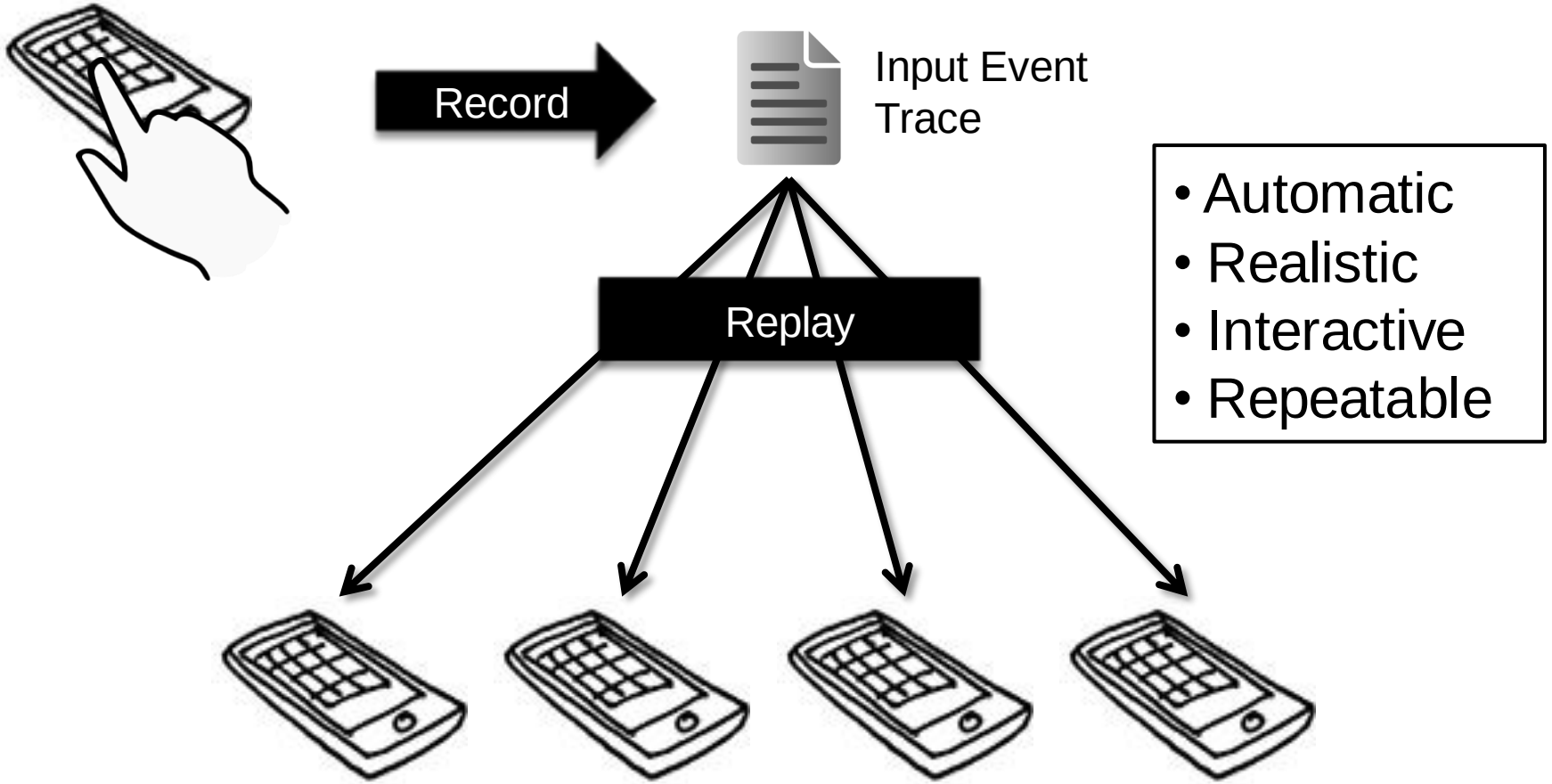
Distance to sweet spot

Methodology must ...

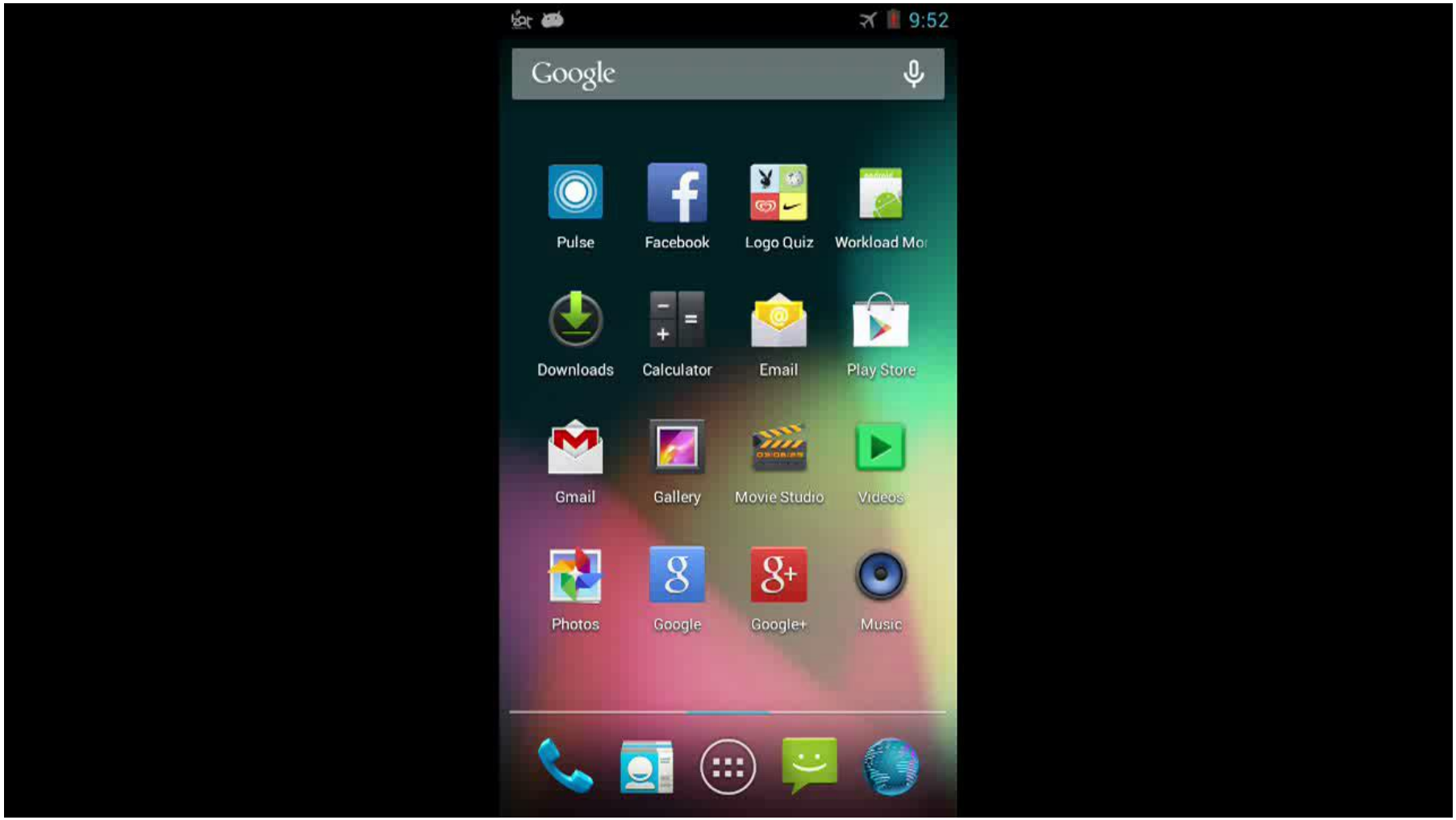
- ... deal with **interactive** workloads
- ... execute **repeatable** workloads
- ... execute workloads **automatically**
- ... identify **interaction lags**
- ... automatically rate **user satisfaction**

Not possible with current mobile benchmarks!

Executing Mobile Workloads



Automatic Workload Execution



Research Goals

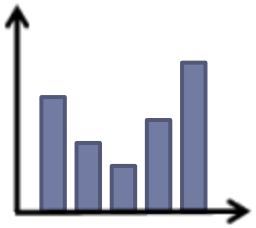
Methodology must ...

- ... deal with **interactive** workloads ✓
- ... execute **repeatable** workloads ✓
- ... execute workloads **automatically** ✓
- ... identify **interaction lags**
- ... automatically rate **user satisfaction**

Considering the User's Perspective Concept

Execute mobile workload and capture a video

Review the video and mark interaction lags



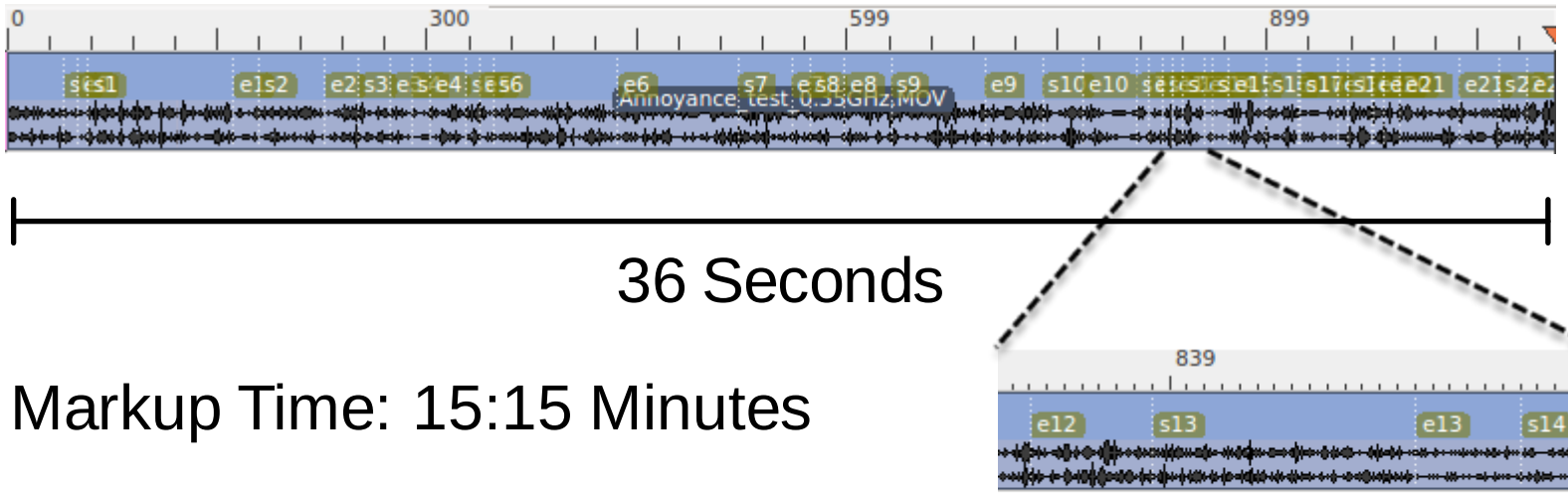
Compare lag lengths to different system configurations

Interaction Lag
Timespan between user input and user feels input has been serviced

Interaction Lag Markup

1104
Frames

22 Lags

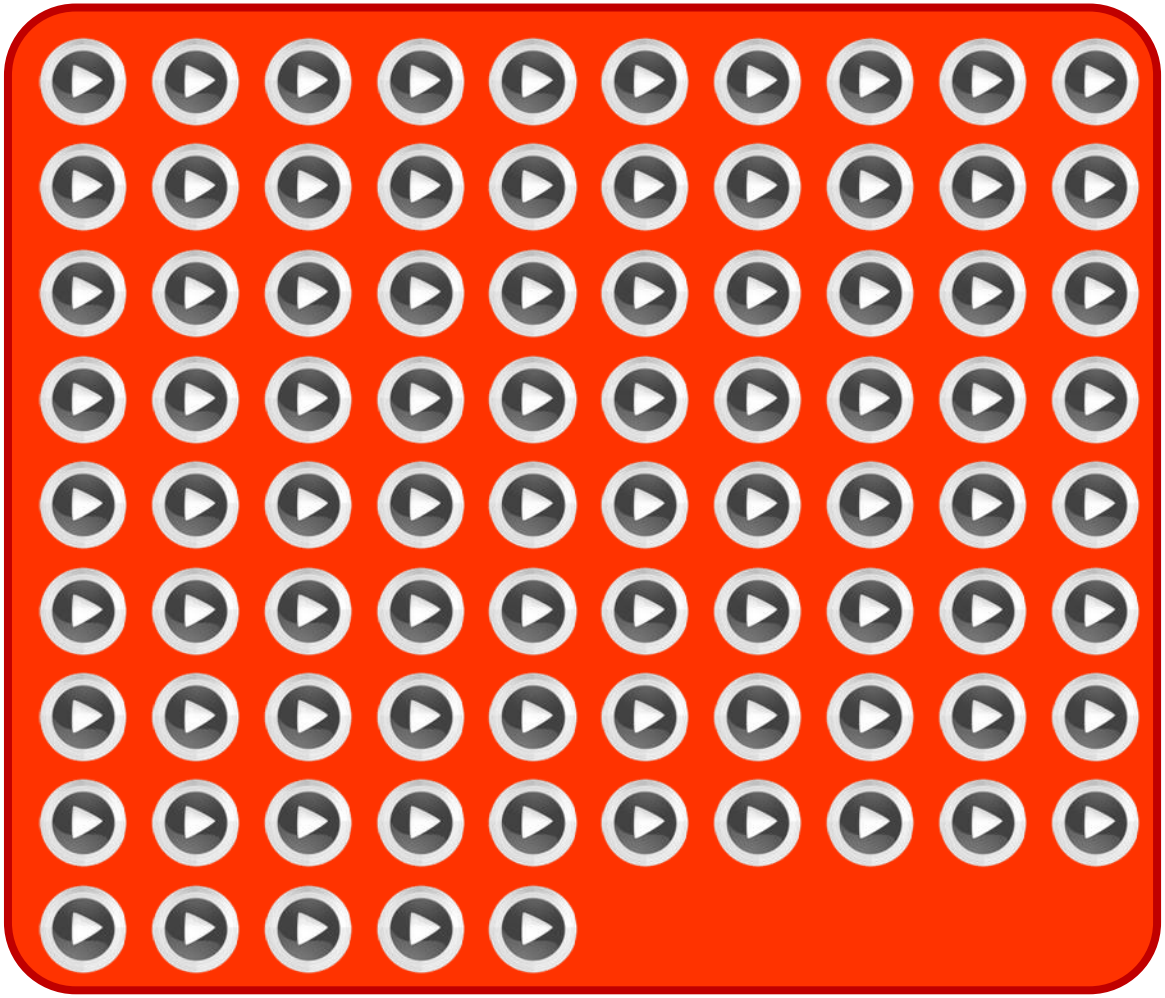


36 Seconds

Markup Time: 15:15 Minutes

Markup Costs

Markup lags in each video manually

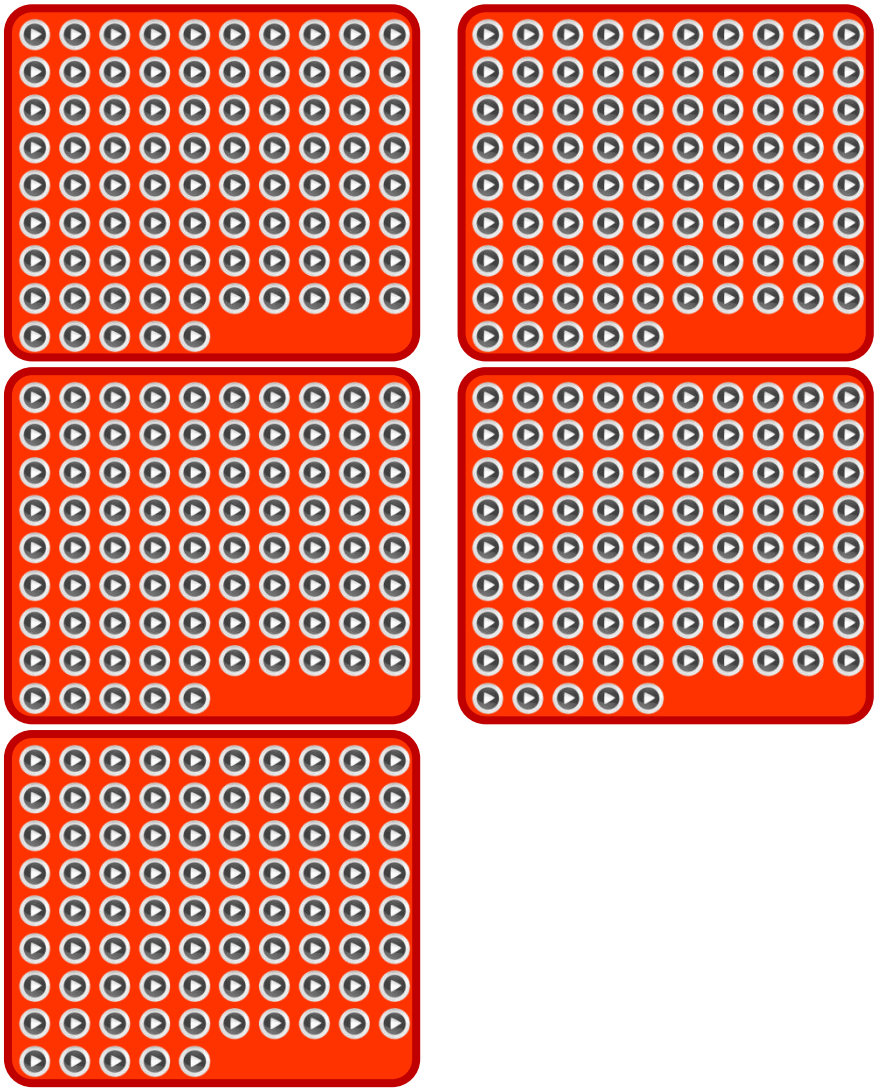


Markup Time:
360 hours
or 9 working weeks

Workload
10 Minutes Length
17 System Configurations
5 Iterations
➔ 85 Videos

Markup Costs

Markup lags in each video manually

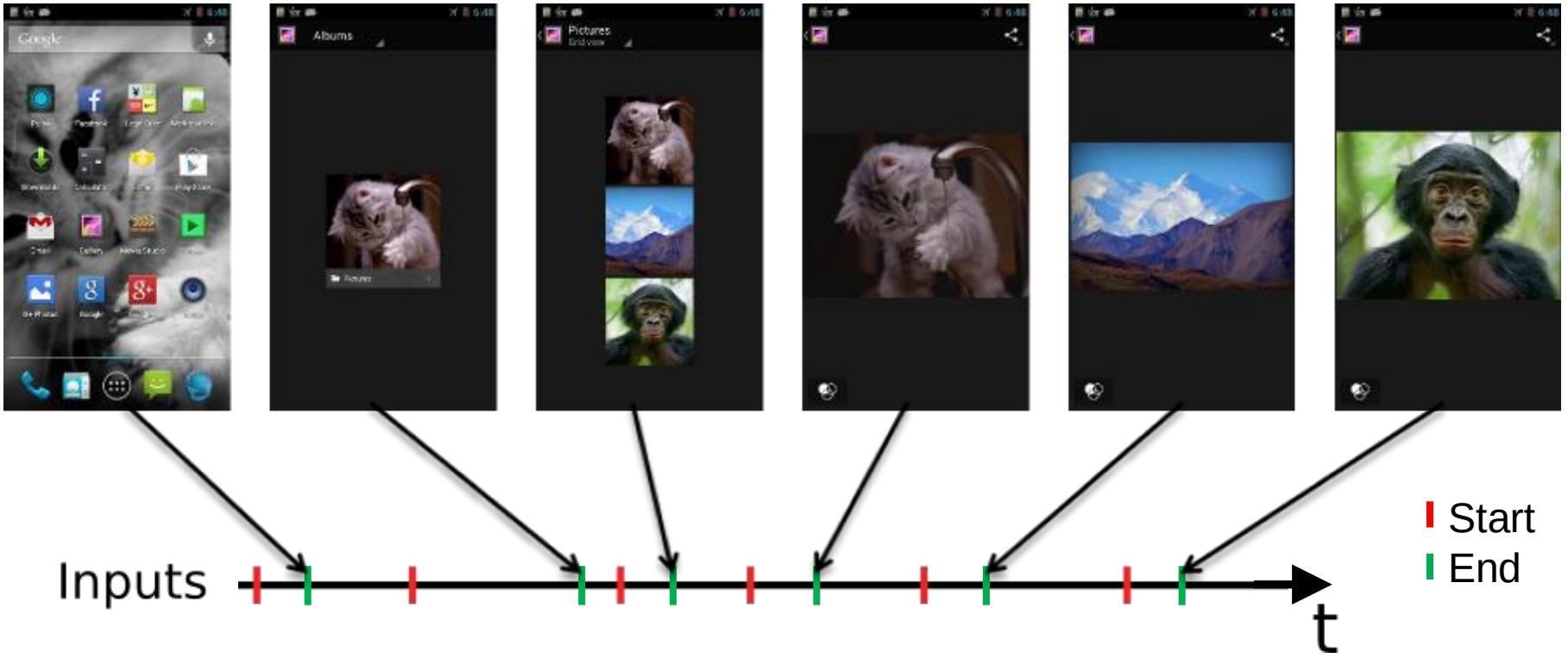


Markup Time:
1800 hours
or 1 working year

5 Workloads
10 Minutes each
17 System Configurations
5 Iterations
➔ 425 Videos

Reusing a Video Markup

Images of Lag Endings



Use an image of the lag ending to find it again in a different video

Dealing with Non-Determinism

Image 1

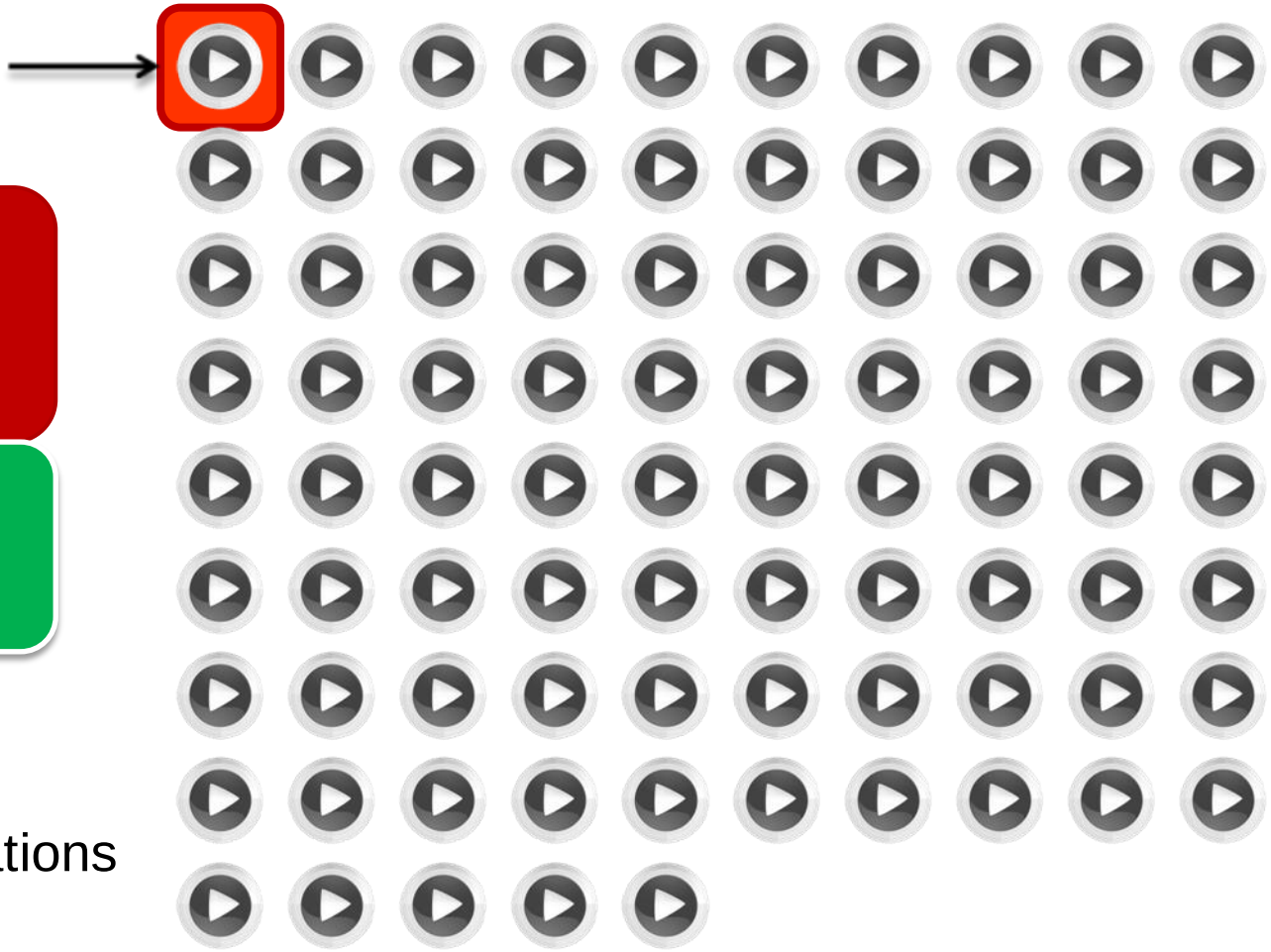
Image 2

Masked Image

Mask out non deterministic areas to compare images.

Markup Costs

Find Lag Endings in a single video of the recorded workload.



Still requires 5 hours of manual work

Speedup of 85x

Workload
10 Minute Workload
17 System Configurations
5 Iterations
➔ 85 Videos

Finding Potential Lag Endings

Previous Input

Current Input

Next Input

Potential End Frames

1

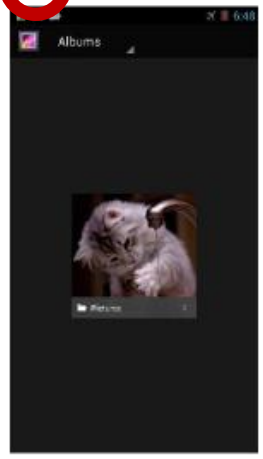
2

3

4

5

6



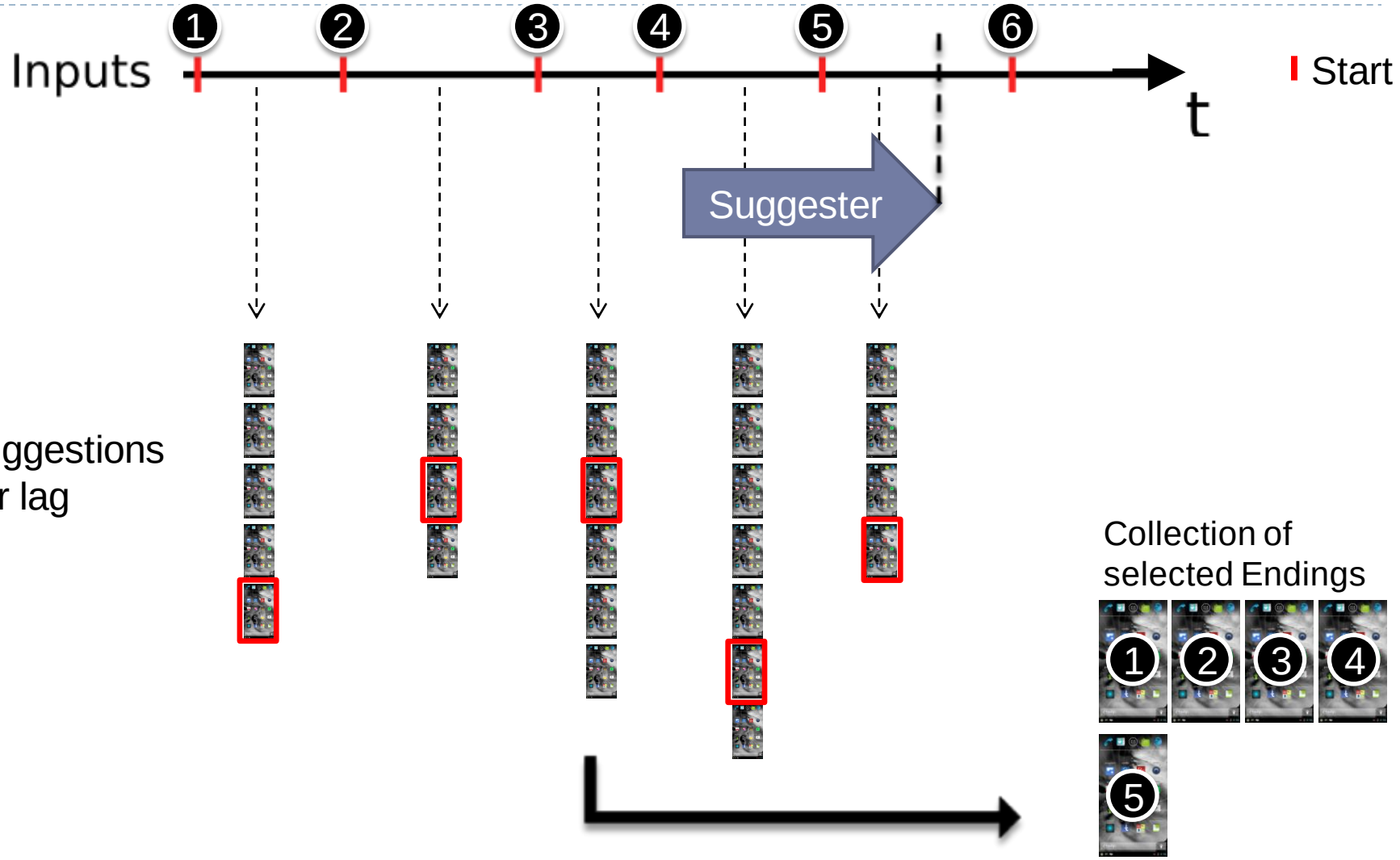
7

8

Pick a lag ending from a selection of potential ending frames rather than looking at every single one.

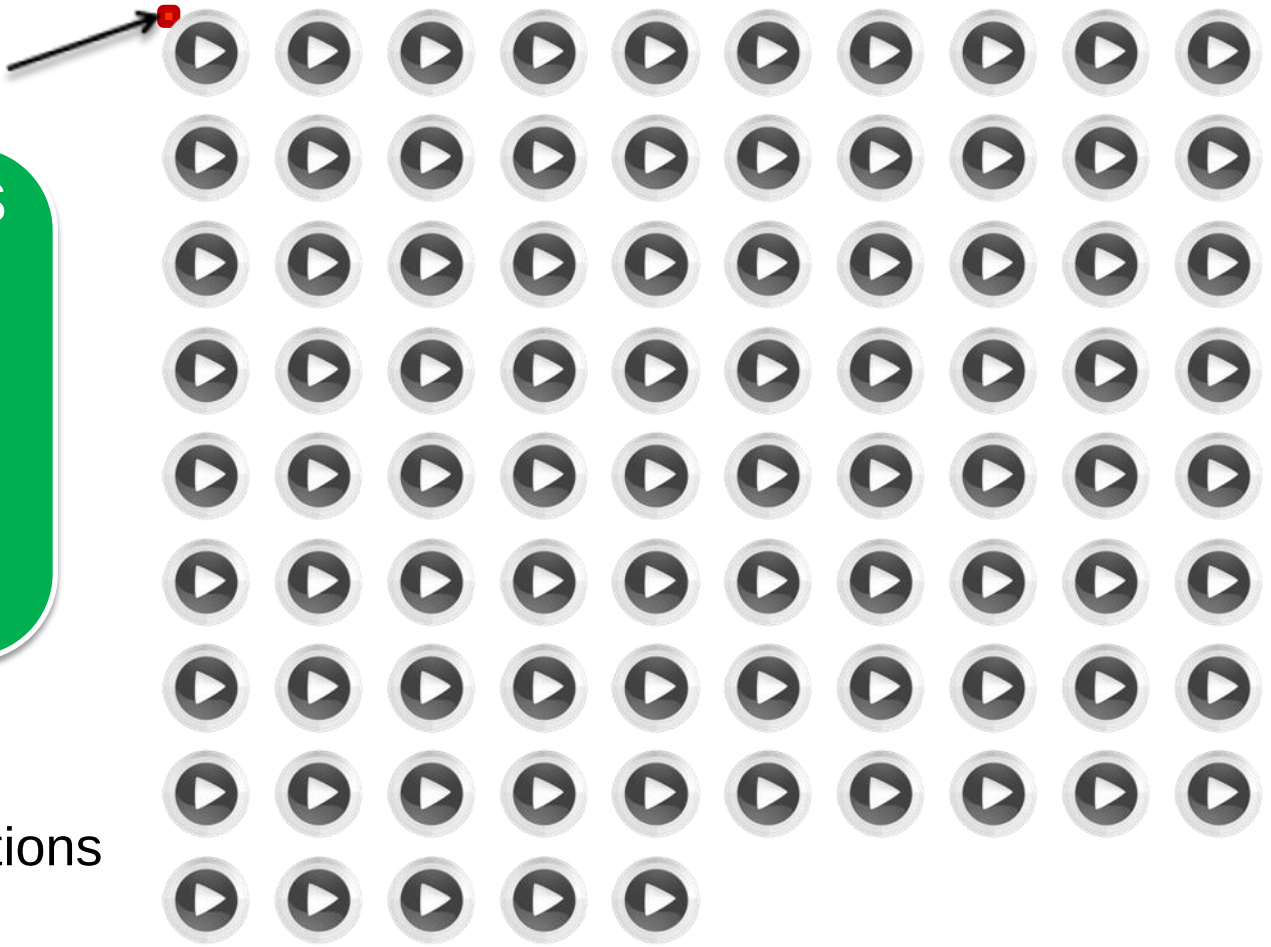
Looking at 8 rather than 191

Suggesting Potential Lag Endings



Markup Costs

Pick lag endings from suggested selection.



16:02 Minutes Manual Markup Work

Speedup of 1347x

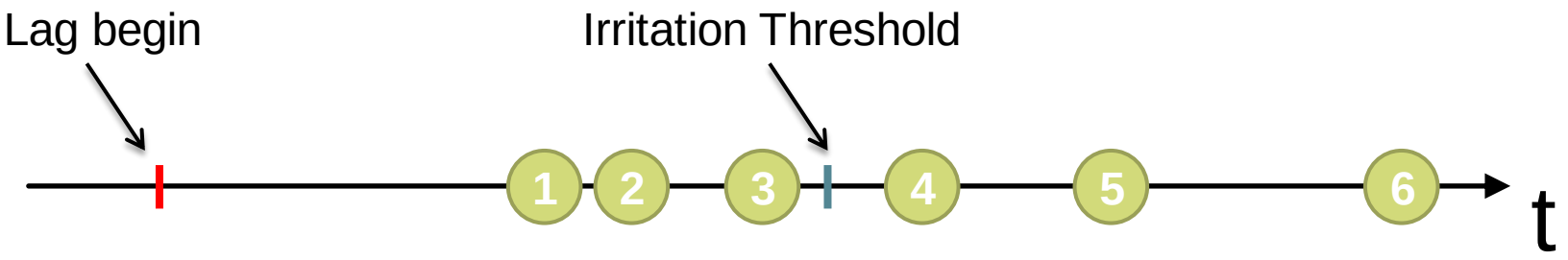
Workload
10 Minute Workload
17 System Configurations
5 Iterations
➔ 85 Videos

Research Goals

Methodology must ...

- ... deal with **interactive** workloads ✓
- ... execute **repeatable** workloads ✓
- ... execute workloads **automatically** ✓
- ... identify **interaction lags** ✓
- ... automatically rate **user satisfaction**

User Irritation Metric



x Lag ending of system configuration x

Calculate User Irritation

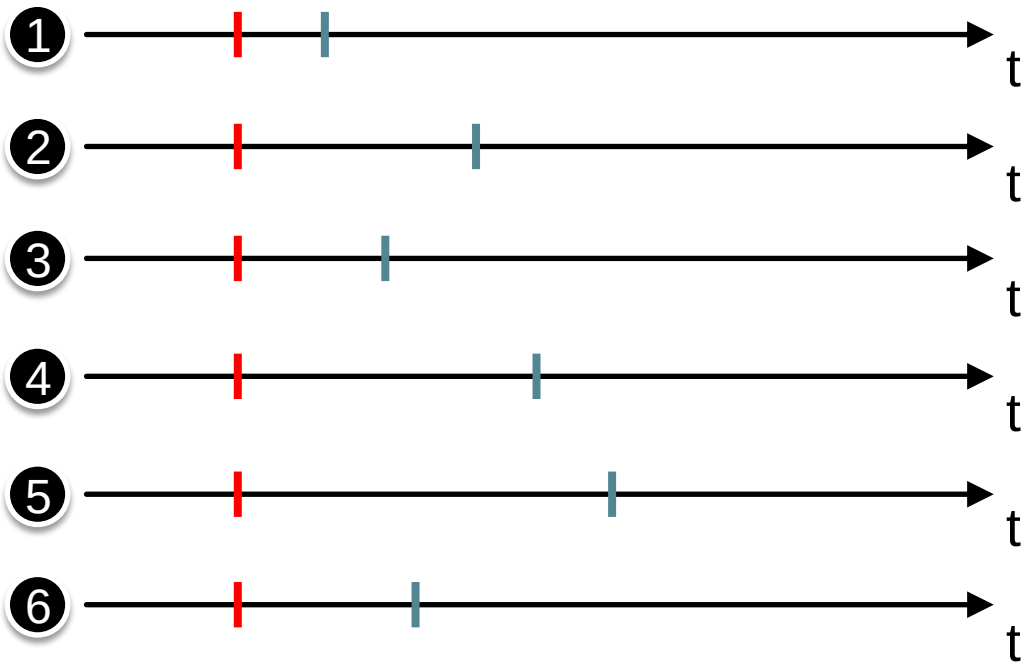
- Set a user irritation threshold for each lag
- If the length of a lag stays below the threshold, it counts as not irritating
- If the length of a lag exceeds the threshold, a penalty is applied

Compare different system configurations in terms of user irritation

Irritation Thresholds

Threshold Policy

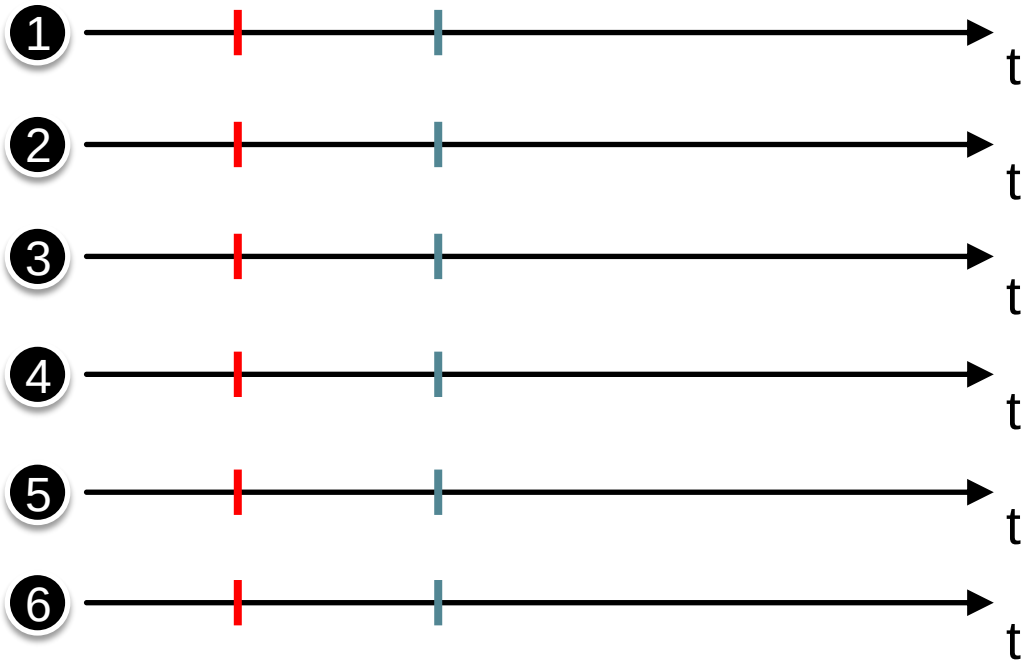
Setting Thresholds per lag in a workload



Irritation Thresholds

Threshold Policy

Setting Thresholds per lag in a workload



Research Goals

Methodology must ...

- ... deal with **interactive** workloads ✓
- ... execute **repeatable** workloads ✓
- ... execute workloads **automatically** ✓
- ... identify **interaction lags** ✓
- ... automatically rate **user satisfaction** ✓

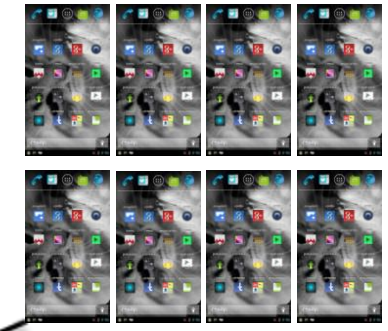
Final Methodology

Run once

Execute **prerecorded** mobile workload and capture a video

Pick lag endings from suggested selection

Ending Images

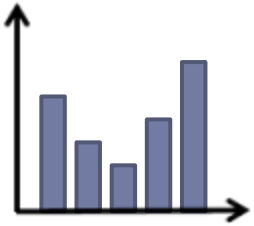


Run arbitrary number of times

Execute **prerecorded** mobile workload and capture a video

Detect lag endings using annotations

Compare lag lengths to different system configurations



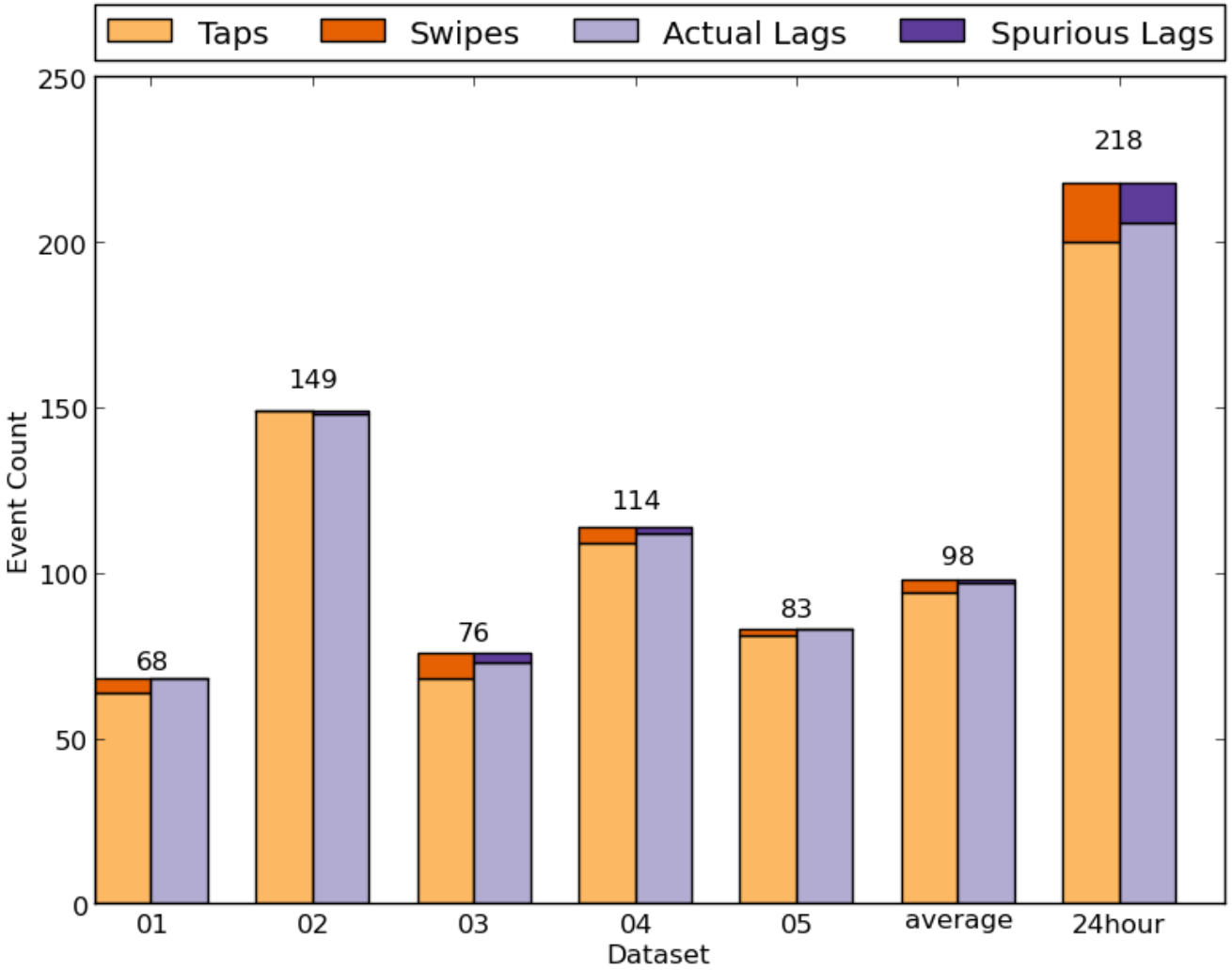
Frequency Governor Case Study

How close are Linux governors to the sweet spot?

Qualcomm Dragonboard 8074
- Snapdragon 800 Processor
- 4.3” qHD 540x960 LCD
- Android 4.3 Jelly Bean

Linux Governors
- Conservative
- Interactive
- Ondemand

Workload Input Classification

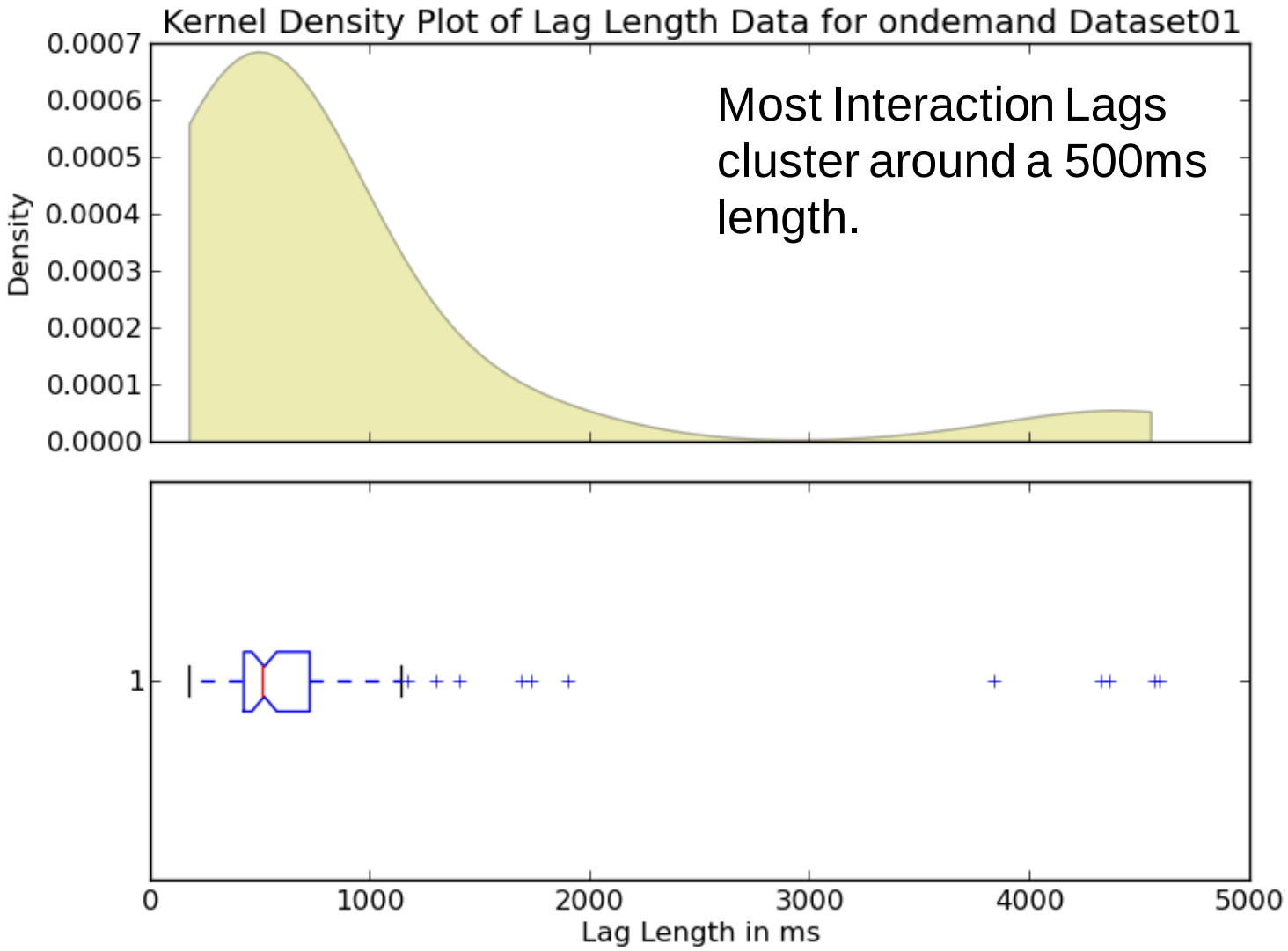


5 Workloads
recorded from 5 Users

Each Workload:
10 Minutes
14 Fixed Frequencies
3 Standard Governors
5 Iterations
➔ 85 Runs

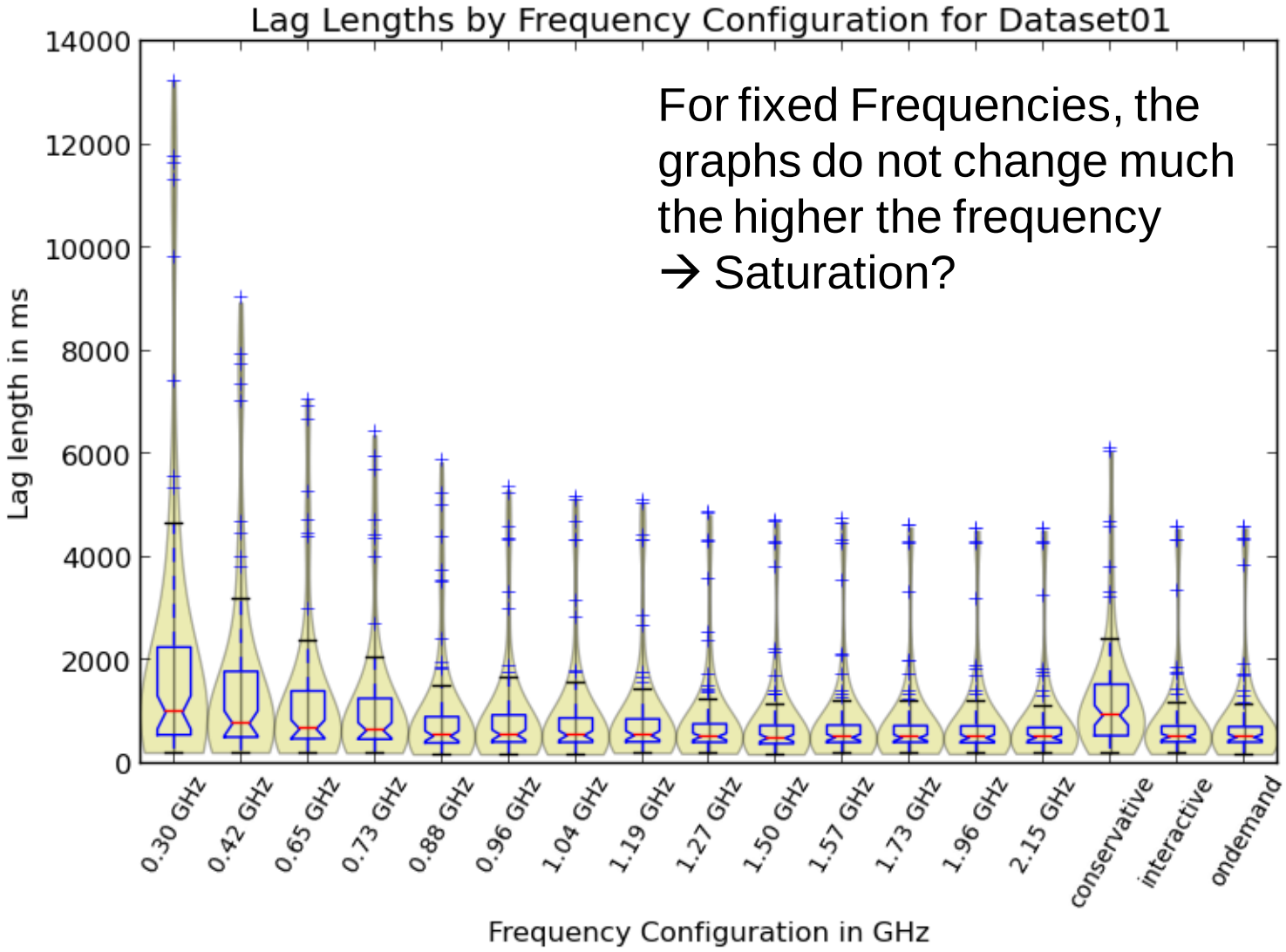
Lag Length of each Lag for Ondemand

Workload 01

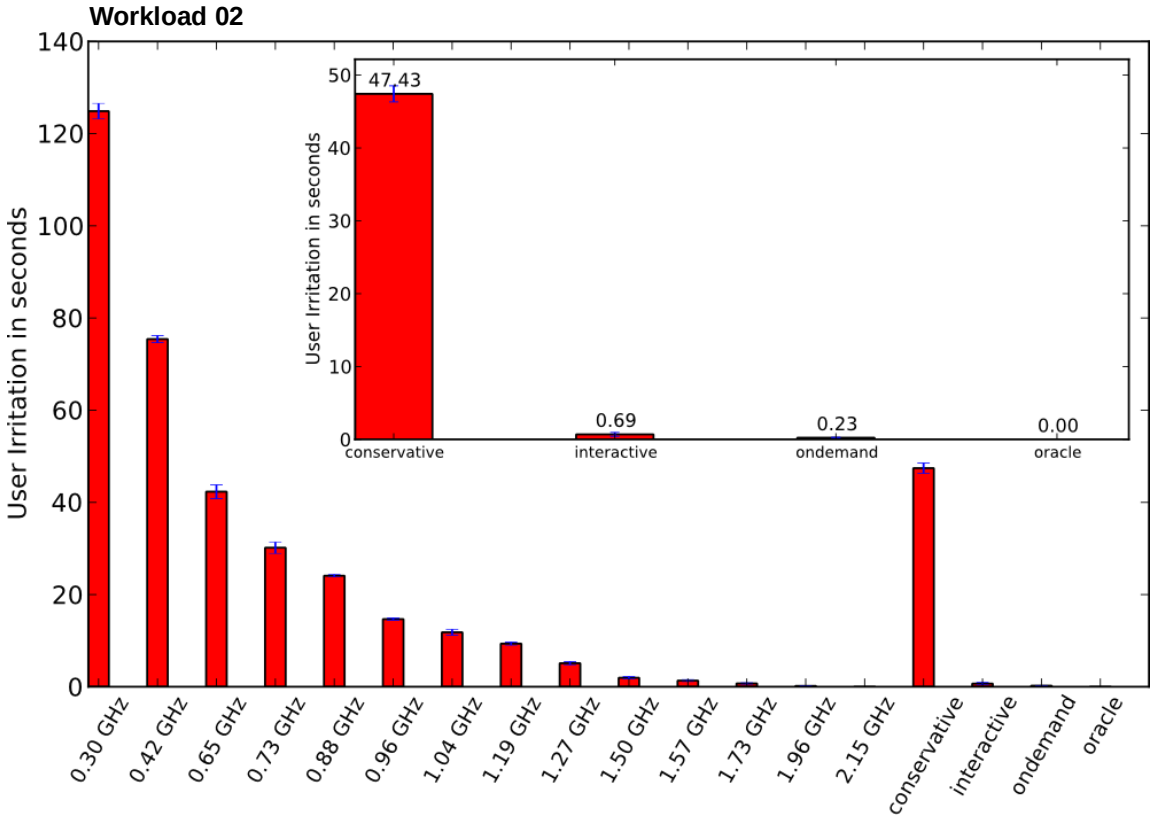


Lag Length of each Lag for all Frequency Configurations and Governors

Workload 01



User Irritation

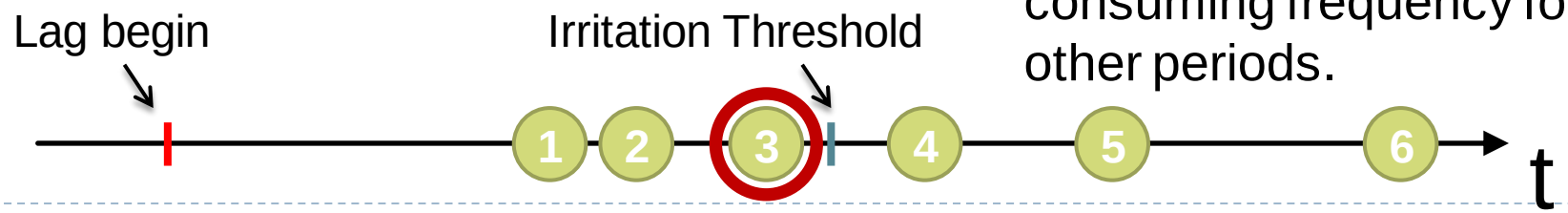


Irritation Threshold
110% of the lag length of the fastest frequency

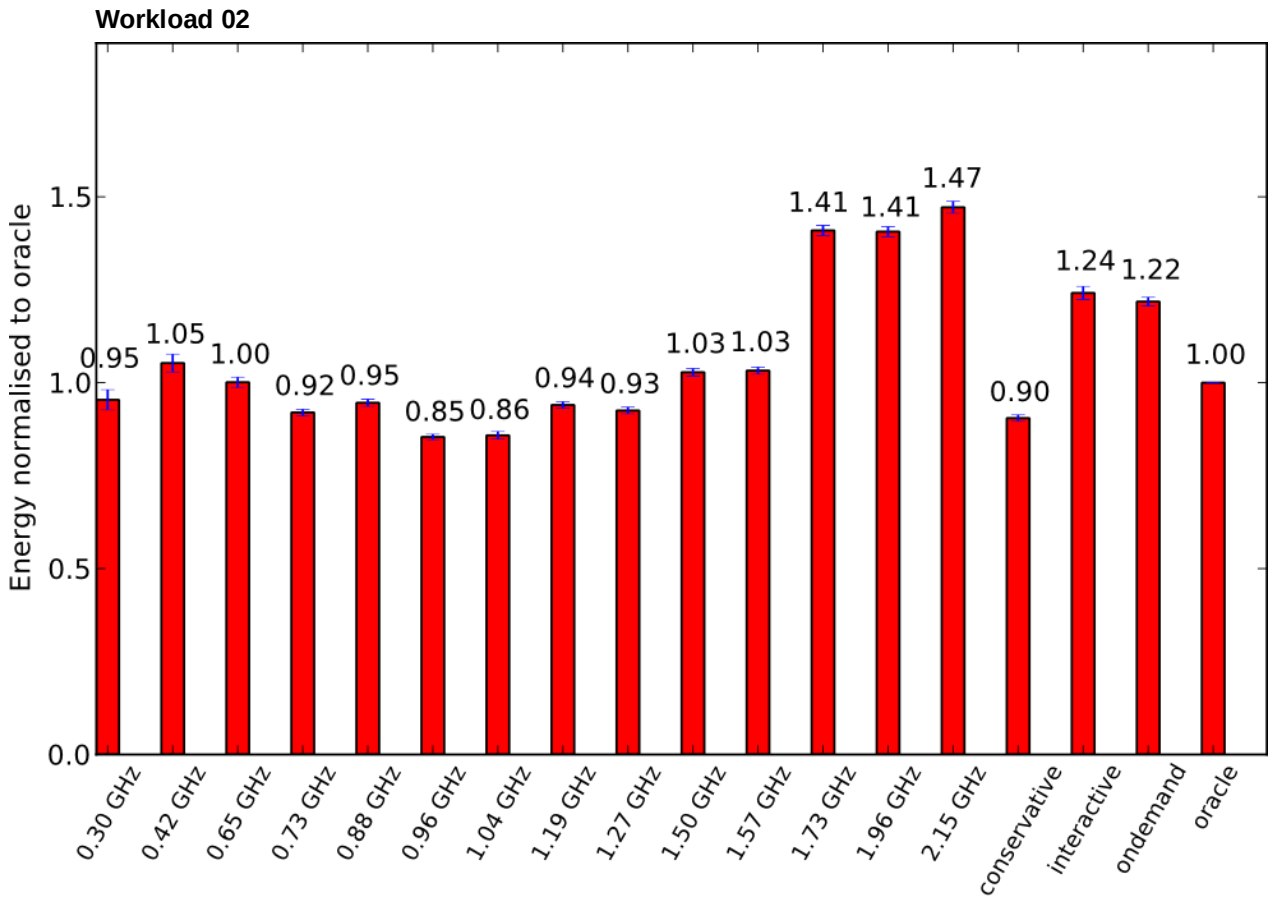
→ Everything below this threshold does not count as irritating

Oracle Governor
Assume for each lag the lowest frequency that is still below the irritation threshold

Assume the least energy consuming frequency for all other periods.



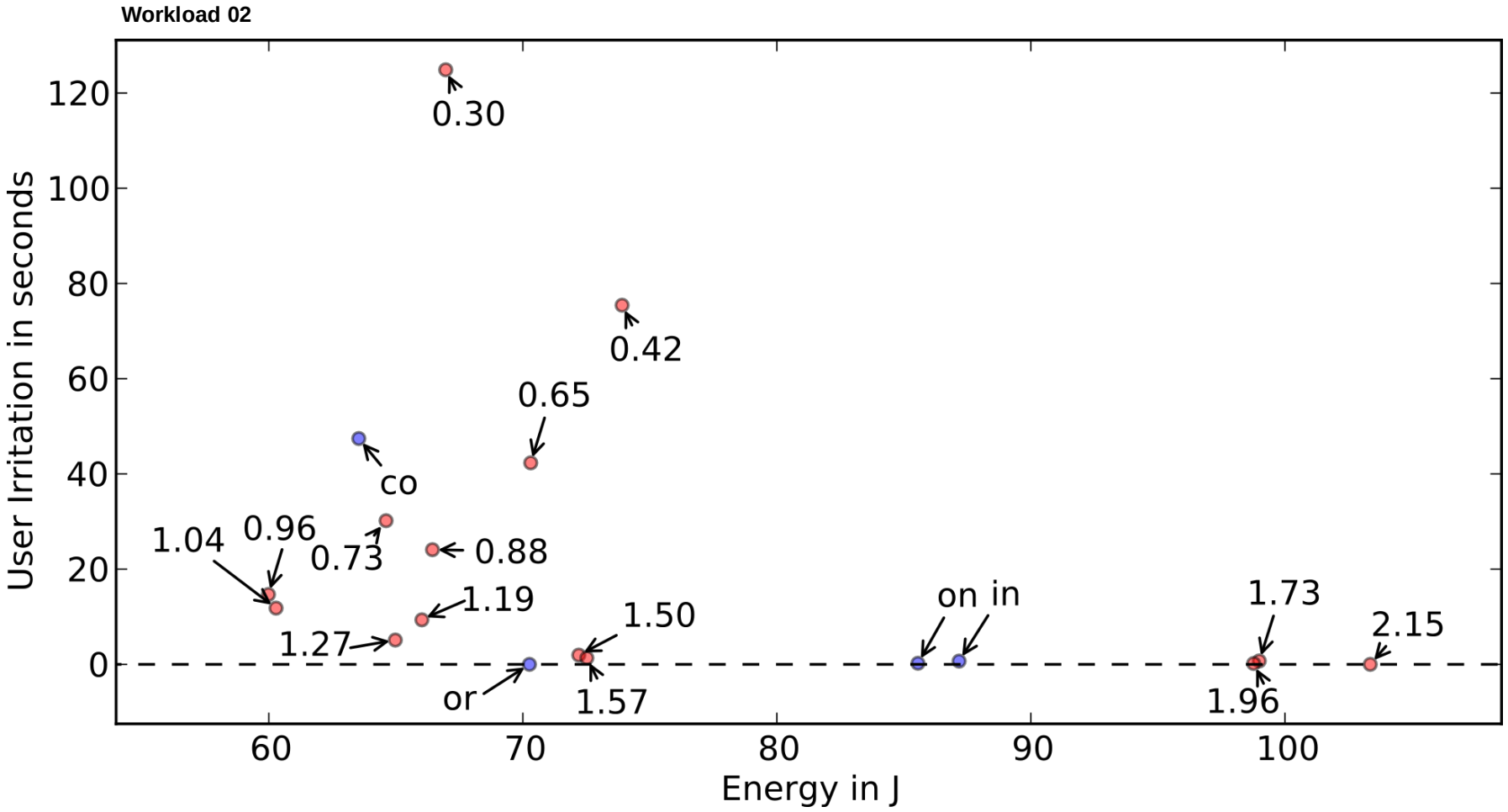
Energy Consumption



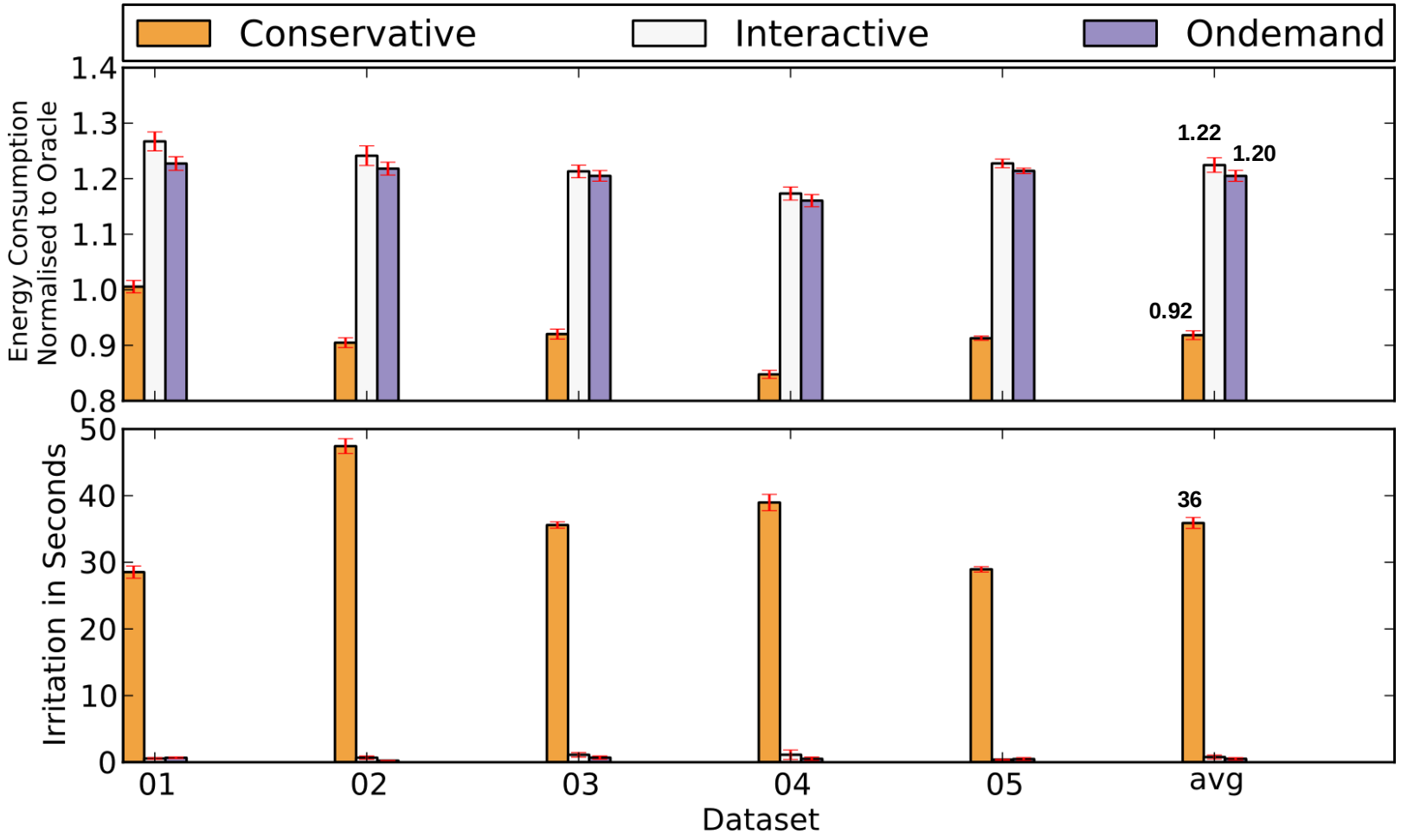
Power Model
Run a CPU intensive artificial micro benchmark with each available frequency fixed.

Calculate average power for each frequency and subtract idle state power.

Energy and User Irritation



Energy and User Irritation



Summary and Future Work

- Automation of proposed method to a high degree **(1347x speedup)**
- Demonstration of method feasibility for standard frequency governors compared to an oracle **(up to 22% less energy)**

Future Work

- Apply methodology to big.LITTLE type heterogeneous processors
- Integrate methodology into OS to make live decisions