**THE PROBLEM & SOLUTION**

- What is wrong with today’s web browsers?
  - None exploit multi-core parallelism benefits.
  - None were designed to handle today’s highly complex web pages.
  - Slow page-load times are a growing concern, especially on mobile platforms.
- What is a solution?
  - A web browser engine that parallelizes its web rendering tasks.
  - Need to avoid parallelism when its overhead outweighs its benefits.
- What is a practical solution?
  - Smart parallelism!
  - Construct a model to predict how many threads to spawn for rendering tasks.
  - Train and test model using supervised learning methods.

**WEB PAGE CHARACTERIZATION**

- The workload of a browser is dependent on the web page it is rendering.
- Document Object Model (DOM) tree of a web page describes its structure.
- DOM tree characteristics intuitively predictive of available parallelism:
  - DOM-size: total number of nodes in the DOM tree
  - attribute-count: total number of attributes in the HTML tags
  - web-page-size: size of the web pages’ HTML in bytes
  - number-of-leaves: number of leaves in the DOM tree
  - avg-tree-width: average number of nodes at each level of the tree
  - max-tree-width: largest number of nodes at a level of the tree
  - avg-work-per-level: average work per level of the tree
- Why characterize a web page?
  - These DOM tree characteristics are the input features of the predictive model.

**AUTOMATED LABELING**

- Why automate labeling?
  - Eliminates need for domain expert to manually label training data.
- Three cost models
  - Performance Cost Model
  - Energy Cost Model
  - Performance & Energy Cost Model
- Labeling flow
  1. Collect performance & energy data for each web page with 1, 2, 4, 8, … threads.
  2. For each web page, compute speedups and greenups for each thread.
  3. Using a tuned cost model, label the web page.
- How to label using the Performance/Energy Cost Model?
  - Using thread that achieves optimal performance/energy.
  - How to label using the Performance & Energy Cost Model?
  - Using Performance-Energy Tuple (PET) buckets.
  - Choose thread from highest PET bucket that satisfies greenup limit.

**CONCLUSION & VISION**

- Evident correlation between DOM-tree features and parallel work.
- Guided, work-load aware scheduling
- Use model to guide scheduling and maintain load-balancing.
- Big LITTLE for parallel web browsers
- Exploit and model BIG-LITTLE parallelism for web rendering tasks.
- App-clutter-free smartphones
- Demonstrate greener and higher performance than native apps.
- Revive the universal Web platform
- Shift focus of app development to the Web, accessible by all device-types.