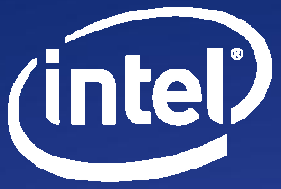


Towards Parallelizing the Layout Engine of Firefox*

HotPar'10

Carmen Badea, UC Irvine
Mohammad R. Haghighat, Intel Corporation
Alex Nicolau, UC Irvine
Alex Veidenbaum, UC Irvine

June 14th, 2010

*This work was supported in part by the National Science Foundation under grant NSF CCF-0811882

Outline

Motivation

CSS Rule Matching Implementation in Firefox

Profiling/Tracing - Strategy and Results

Parallel CSS Rule Matching

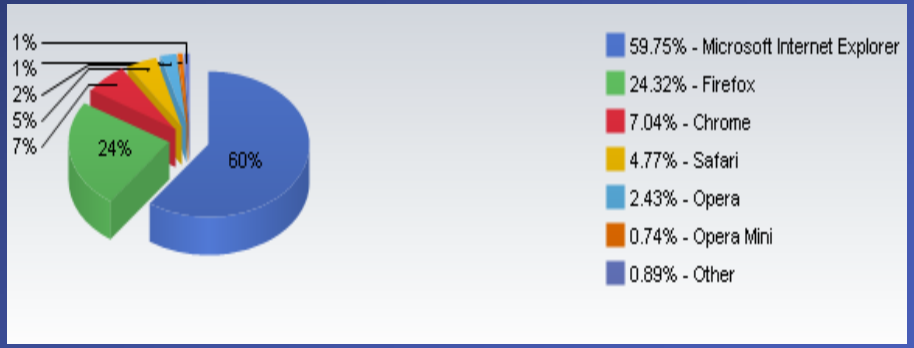
Performance Results

Summary

Parallelizing the Firefox Layout Engine - Motivation

Mozilla Firefox*

- ~24% market share of all the total web browsers
- Improved performance since the recent adoption of the tracing JIT JavaScript* engine (TraceMonkey*)
- Layout Engine
  - heavy performance hitter component
  - 40% of the total execution time
    - VTune* experiments on Intel platforms

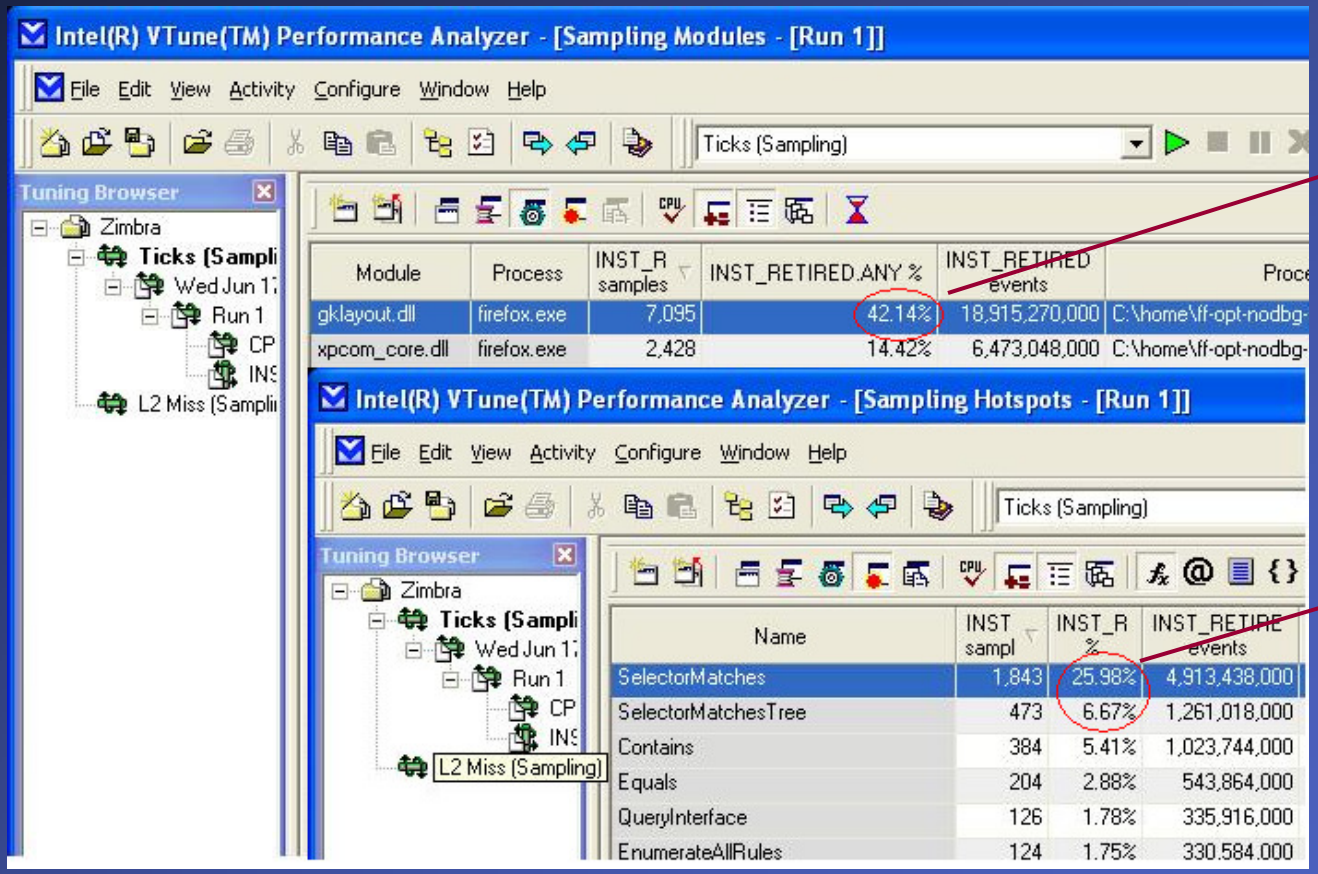


Source: Net Applications (Browser Market Share)
http://marketshare.hitslink.com/

Parallelizing the Firefox Layout Engine - Motivation

Firefox Layout Engine

- Cascading Style Sheets (CSS) rule matching
  - hot part of layout
  - amenable to parallelization

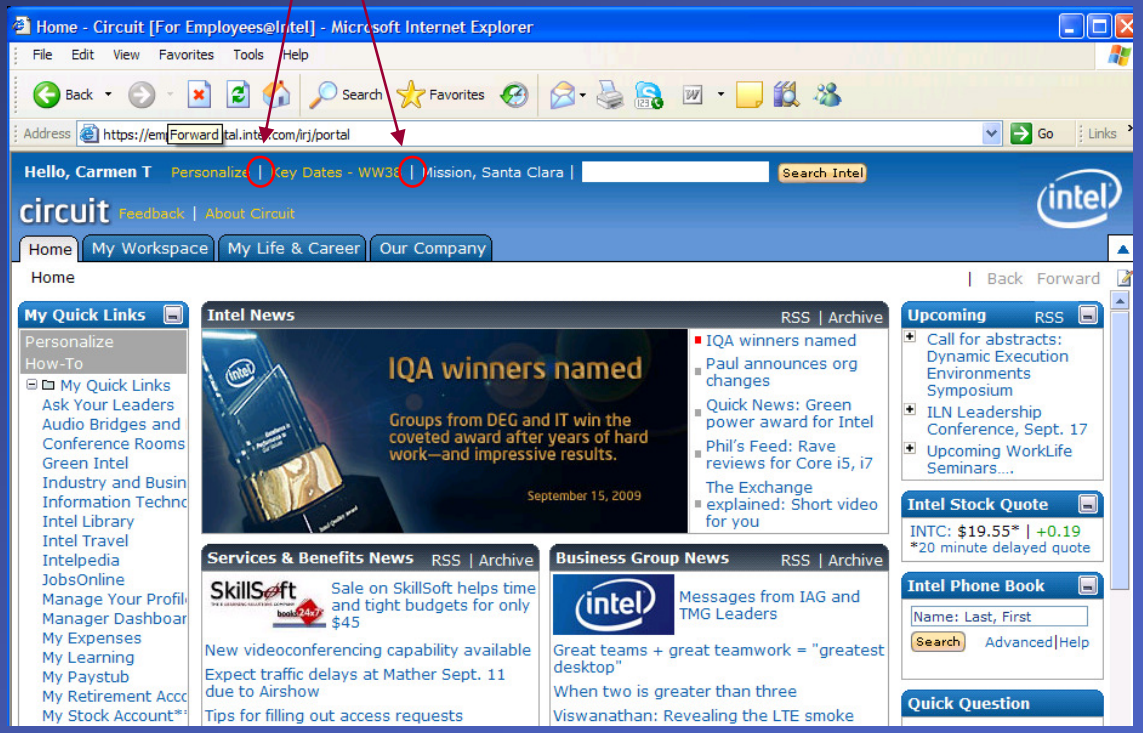


Cascading Style Sheets: A bit of background

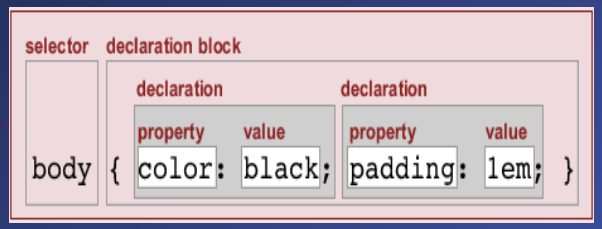
- CSS: designed to allow the separation between webpage content (HTML) and presentation (e.g. colors, fonts, etc.)
- Enables multiple pages to share formatting features and reduces the amount of repetition in specifying layout styles

```
<style>
    .intelSeperator{
        color:#fff;
        font-size:9pt;
    }
</style>

<a class="intelSeperator">|</a>
```



CSS rule structure



*

- Simple selector
- Combination of simple selectors

*** Image source: http://css.maxdesign.com.au/**

Outline

Motivation

CSS Rule Matching Implementation in Firefox

Profiling/Tracing - Strategy and Results

Parallel CSS Rule Matching

Performance Results

Summary

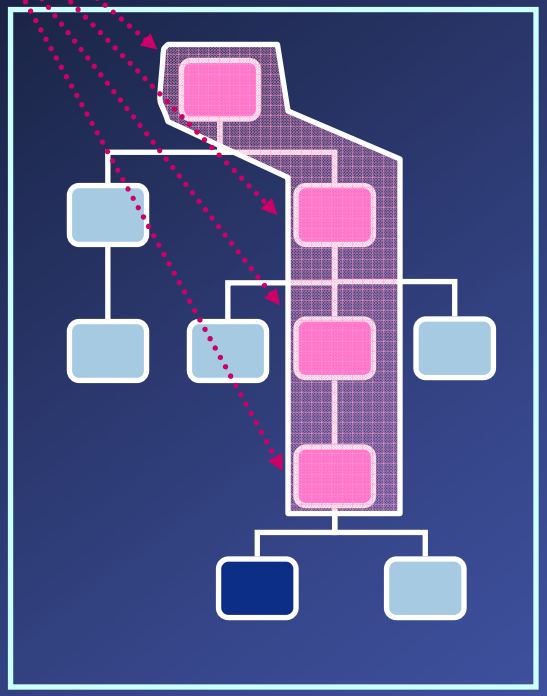
CSS Rule matching implementation in Firefox

CSS style rule examples

```
h1 + h2 {font-size: 9pt} <- adjacent sibling selector

ul em { color: blue}     <- descendant selector
```

*



HTML sample document tree

* Samples courtesy of http://css.maxdesign.com.au/

CSS Rule matching process in Firefox

- Does a style rule applies to a certain HTML element ?
- Series of iterations
  - each iteration executes a SelectorMatches call
  - One for each ancestor up the tree

Outline

Motivation

CSS Rule Matching Implementation in Firefox

Profiling/Tracing - Strategy and Results

Parallel CSS Rule Matching

Performance Results

Summary

Profiling/Tracing Strategy

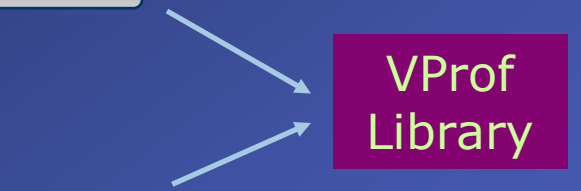
Firefox CSS Rule matching process

- Algorithm split into two main branches
  - Sibling selector
  - Descendant selector

Which is most executed?

- Four exit points

Which is the most often taken?



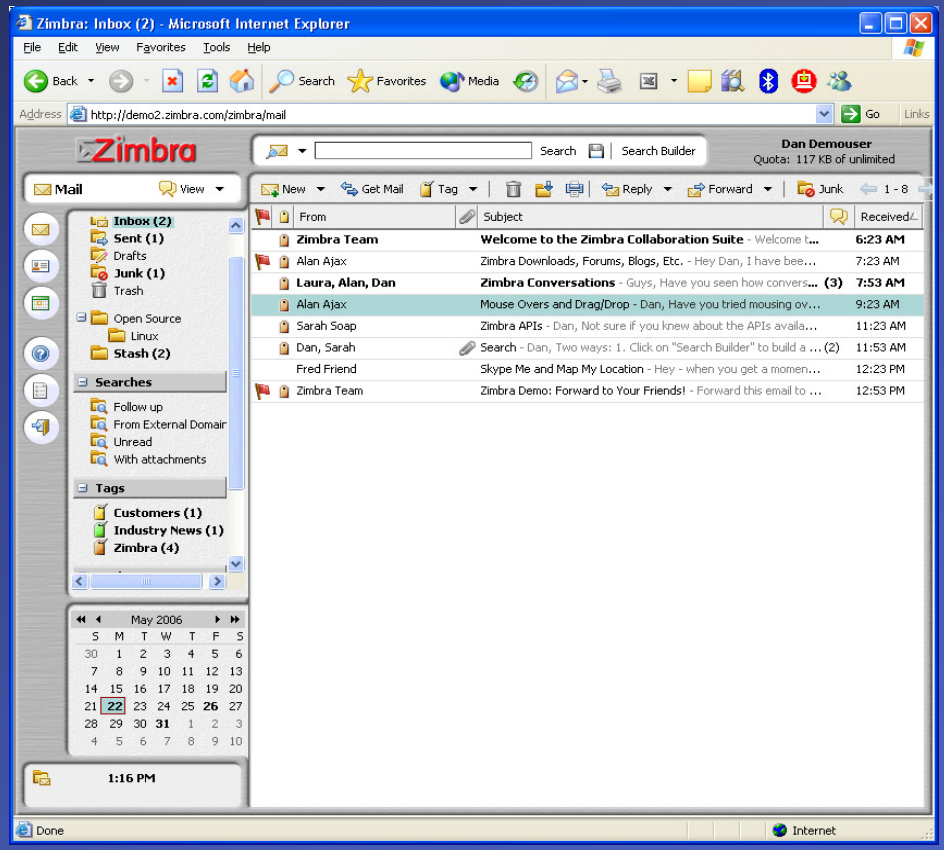
  - Implemented an extensive tracing algorithm
  - Execution patterns of the CSS rule matching process

Profiling/Tracing Results

Benchmarks

- Mozilla Firefox Page Load Tests
  - A collection of 390 web pages from all over the world
  - Test the page loading process in Firefox

- Zimbra* Collaboration Suite
  - AJAX-based rich-browser integrated suite of email, calendar, contacts, VoIP, etc.



Profiling/Tracing Results for Mozilla Page Load Tests

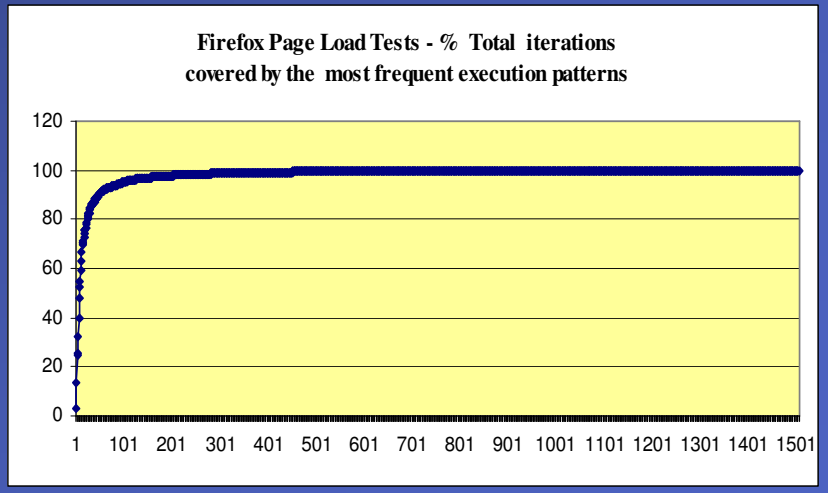
- Selector matching process heavily biased towards descendant selectors
  - 99% of the time for both benchmarks
- One exit point taken in the majority of cases
  - **Dominant pattern: rule non-match**
  - Zimbra Collaboration Suite : ~ 89%
  - Mozilla Page Load Tests : ~ 66%
- Mozilla Page Load Tests
  - top 25 most frequent execution patterns cover ~ 80% of all executed iterations

Microsoft Excel - FFBenchTracesFull

File Edit View Insert Format Tools Data Window Help

F2

	A	B	C	D
1	#Occurences	Pattern	#Iterations	%iterations covered cumulative
2	5561595	p3	1	2.93
3	1997994	ppppppppppn1	10	13.43
4	1977238	pppppppppppn1	11	24.88
5	1846126	pn3	1	25.85
6	1599769	pppppppppn1	8	32.58
7	1530957	ppppppppppn1	9	39.83
8	1293911	pppppppppppppn1	12	47.99
9	1187742	pppppppn1	7	52.37
10	840304	ppppppn1	6	55.02
11	619361	pppppppppppppn1	13	59.25
12	550482	ppppppppppppppn1	14	63.31
13	418940	pppppppppppppppn1	15	66.61
14	382272	ppppppppppppppppn1	16	69.83
15	310138	pppppn1	5	70.65



Outline

Motivation

CSS Rule Matching Implementation in Firefox

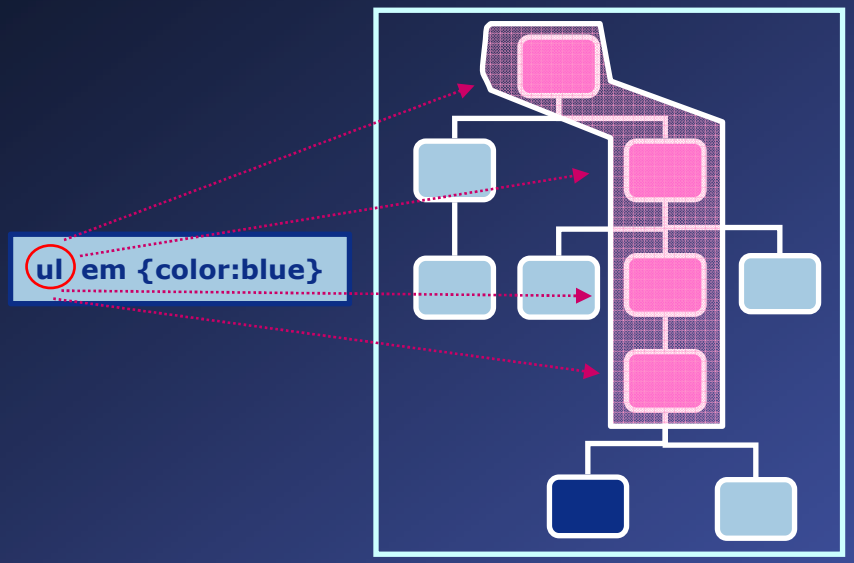
Profiling/Tracing - Strategy and Results

Parallel CSS Rule Matching

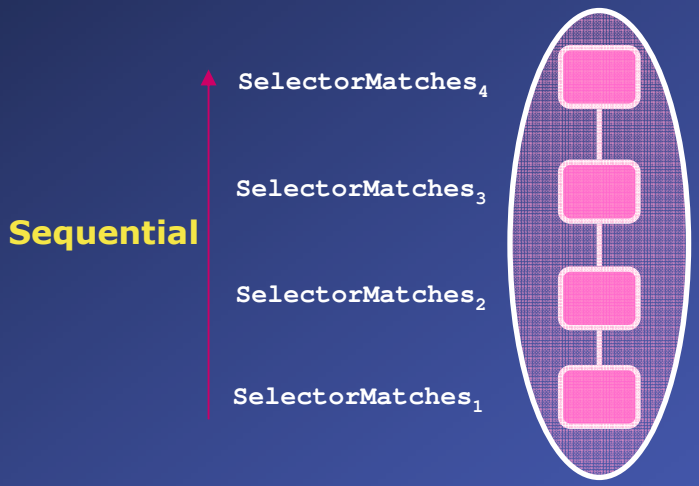
Performance Results

Summary

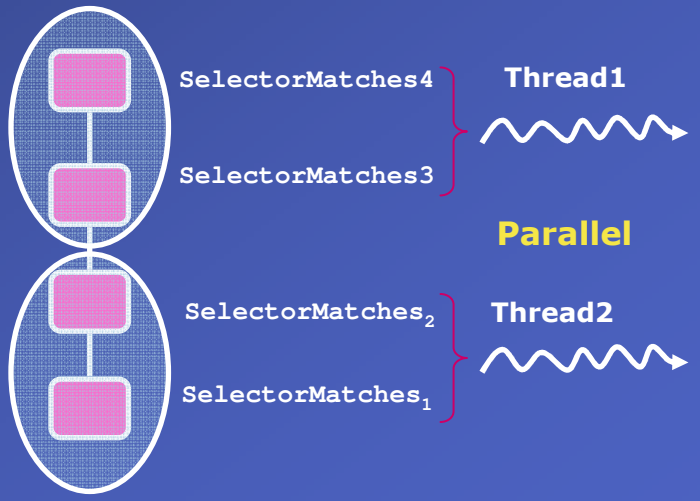
CSS Rule Matching – Parallel Implementation



Original



Parallelized



Outline

Motivation

CSS Rule Matching Implementation in Firefox

Profiling/Tracing - Strategy and Results

Parallel CSS Rule Matching

Performance Results

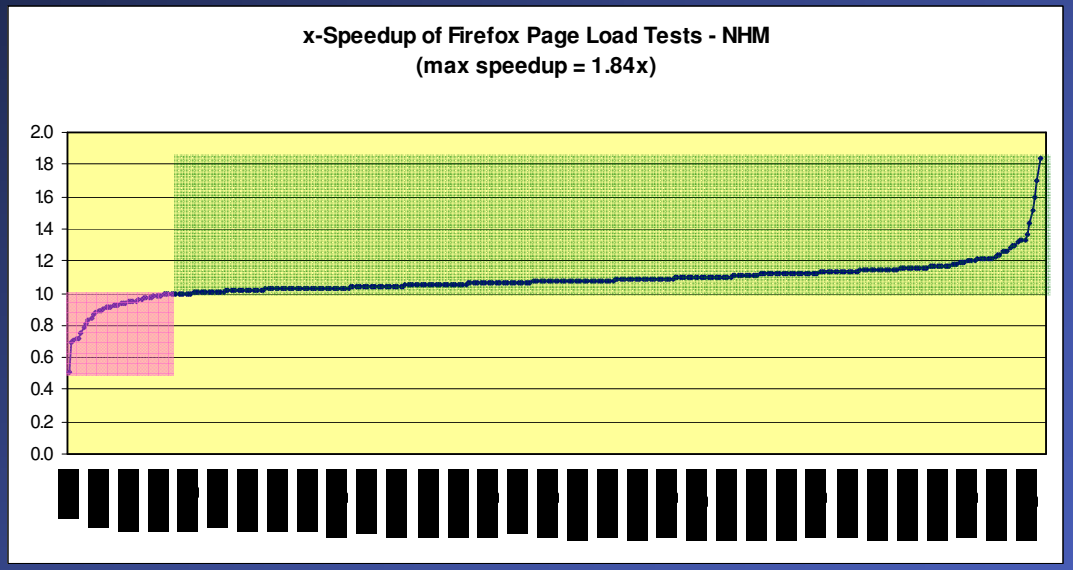
Summary

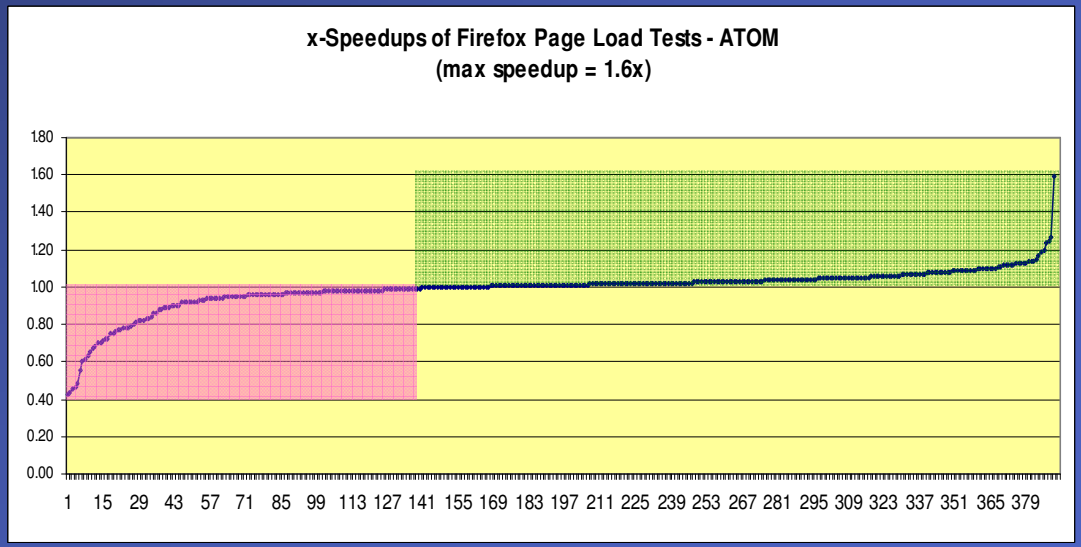
Parallelized CSS Rule Match in Firefox - Performance Results

- Flexible and configurable
- Parallel configuration parameters
  - Default iteration chunk size
  - # of worker threads available
  - Threshold on the # of ancestors necessary to enable parallel CSS matching
- We have tested
  - 7 configurations for the Zimbra Collaboration Suite
  - 12 configurations for the Firefox Page Load Tests
  - Kept first two parameters fixed and varied the last one
    - 2 threads w/ default 4-iterations chunk

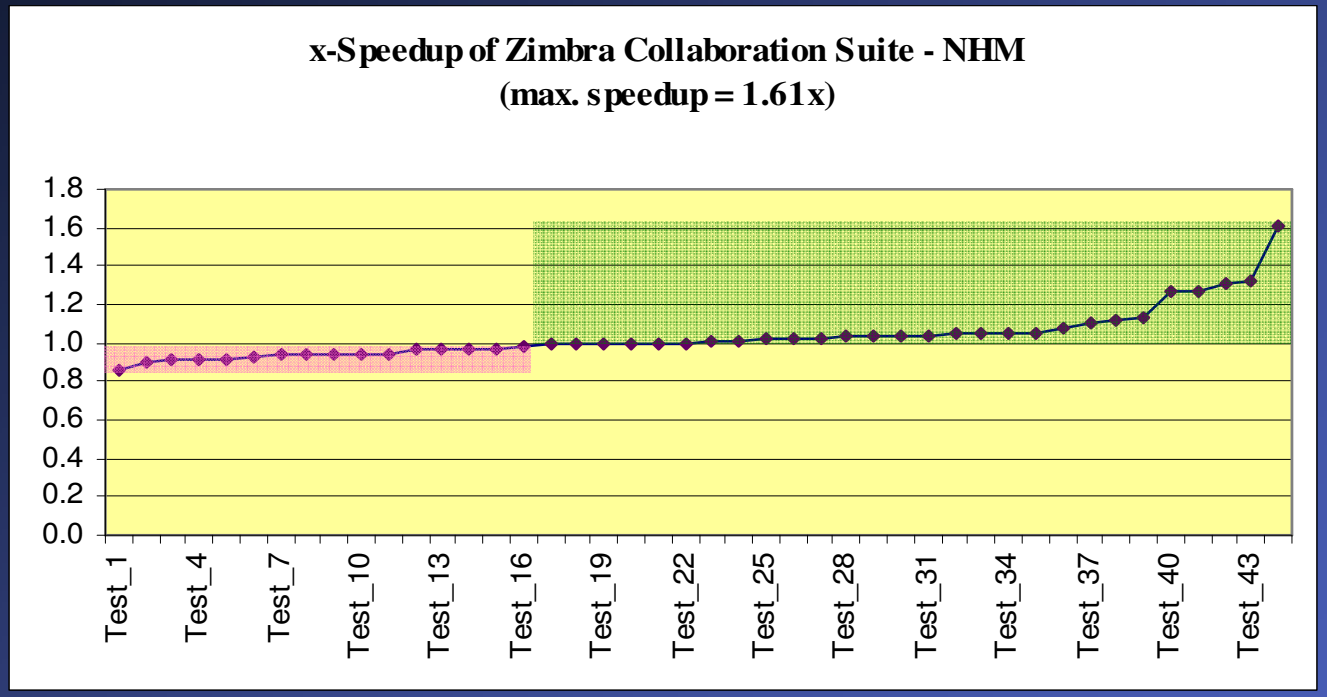
Parallelized CSS Rule Matching: Mozilla Tests Performance Results

- Mozilla Firefox Page Load Tests
  - Comprise ~ 400 distinct web page load tests
- Experiments carried out using 2 worker threads
- Mozilla Page Load Tests performance results on Nehalem
  - ~ 87% benchmarks show speedups
  - ~ 1.8x maximum speedup observed
- Mozilla Page Load Tests performance results on Atom
  - ~ 60% benchmarks show speedups
  - ~ 1.6x maximum speedup observed
  - Hyper-threaded





Parallelized CSS Rule Matching: Zimbra Suite Performance Results



The experiments were carried out on a Nehalem machine, with 2 worker threads

Zimbra performance results on Nehalem

- ~ 50% benchmarks show speedups
- 14% of the benchmarks are not affected performance-wise
- ~1.6x maximum speedup observed

Summary

- Implemented a fully-functional, **parallelized** CSS rule matching component in Firefox
  - Mozilla Page Load Tests
    - ~ 1.8x maximum speedup on NHM
    - ~ 1.6x maximum speedup on Atom
  - Zimbra Collaboration Suite: ~ 1.6x maximum speedup on NHM
- Submitted the parallel CSS rule matching code to Mozilla
  - Excellent feedback from Firefox layout engine component owners
  - Incorporated their suggestions into the existing parallelized code
  - Awaiting adoption in a future release

References

- Mozilla Project [*http://www.mozilla.com/en-US/]*
- Zimbra Collaboration Suite [*http://www.zimbra.com/]*
- Cascading Style Sheets (CSS) [*http://www.w3.org/Style/CSS/]*
- Intel VTune Performance Analyzer [*http://software.intel.com/en-us/intel-vtune/]*
- VProf Value Profiling Library [*Mozilla Source Code]*
- Leo Meyerovich and Ras Bodik, *Fast and Parallel Webpage Layout*, WWW 2010: 19th International World Wide Web Conference (Raleigh, NC, USA), 2010.
- Chris Grier, Shuo Tang, and Samuel T. King, *Secure Web Browsing with the OP Web Browser*, SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy (Washington, DC, USA), IEEE Computer Society, 2008, pp. 402–416.
- Jungwoo Ha, Mohammad R. Haghighat, Shengnan Cong, and Kathryn S. McKinley, *A Concurrent Trace-based Just-In-Time Compiler for Single-threaded JavaScript*, 2009 Workshop on Parallel Execution of Sequential Programs on Multicore Architectures (PESPMA 2009), in conjunction with ISCA.

Thank you!