Parallel Browser Layout

Eric H. Atkinson (ericatkinson@eecs.berkeley.edu)
Undergraduate, University of California, Berkeley, 565 Soda Hall, Berkeley, CA, 94720
ACM Student ID: 1948855

1. Introduction

Instead of simply displaying web pages, web browsers are increasingly being used to run full-scale applications like Gmail, Facebook, and Google Docs. Projects such as Firefox OS and Chrome OS even integrate the browser with the operating system and implement a userland entirely with web applications. However, browsers must be improved before web applications can run efficiently on smaller, more resource-constrained hardware.

One area for improvement is layout engines. Layout engines are responsible for computing size, position, and styling information for each element in a given document, then displaying them visually. In practice, these computations are done by traversing a tree representation of the document. Layout engines are currently a bottleneck for web browsers [2, 4], and our goal is to alleviate this by speeding up the tree traversals.

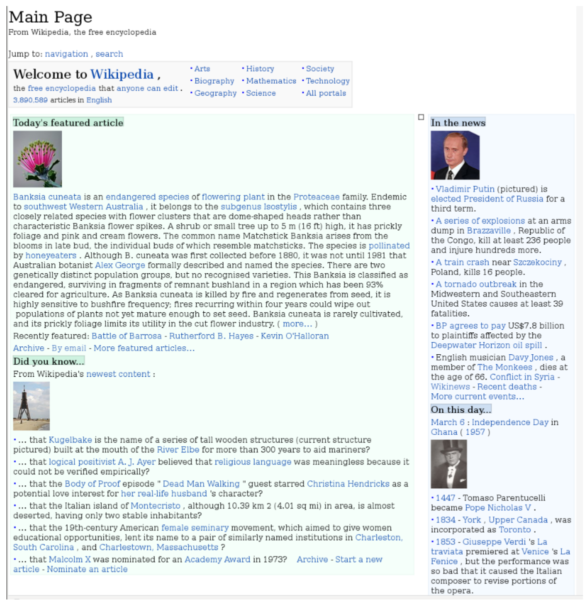
Specifically, we want to exploit parallelism while performing tree traversals over the document. Other components of the browser—like the lexer [3], parser [3], and CSS selector matching algorithm [4]—can already be parallelized, and tree traversals might seem like a comparatively well-understood problem. However, current layout engines are unable to use parallel traversal algorithms.

2. Approach

2.1 Problem Statement

Layout engines are written in C++ or other relatively low-level languages. They contain many hand-coded optimizations and specialized data structures that are designed to maximize their incrementality. The complexity of this code means they are notoriously difficult to implement in their current form and have resisted parallelization.

Our approach has been to specify layout semantics declaratively and use this to generate a parallel implementation.


Main Page
From Wikipedia, the free encyclopedia

Jump to: navigation , search

Welcome to Wikipedia ,
the free encyclopedia that anyone can edit .
3,890,589 articles in English

• Arts • History • Society
• Biography • Mathematics • Technology
• Geography • Science • All portals

Today's featured article

Banksia cuneata is an endangered species of flowering plant in the Proteaceae family. Endemic to southwest Western Australia , it belongs to the subgenus Isostylis , which contains three closely related species with flower clusters that are dome-shaped heads rather than characteristic Banksia flower spikes. A shrub or small tree up to 5 m (16 ft) high, it has prickly foliage and pink and cream flowers. The common name Matchstick Banksia arises from the blooms in late bud, the individual buds of which resemble matchsticks. The species is pollinated by honeyeaters . Although B. cuneata was first collected before 1880, it was not until 1981 that Australian botanist Alex George formally described and named the species. There are two genetically distinct population groups, but no recognised varieties. This Banksia is classified as endangered, surviving in fragments of remnant bushland in a region which has been 93% cleared for agriculture. As Banksia cuneata is killed by fire and regenerates from seed, it is highly sensitive to bushfire frequency; fires recurring within four years could wipe out populations of plants not yet mature enough to set seed. Banksia cuneata is rarely cultivated, and its prickly foliage limits its utility in the cut flower industry. (more...)

Recently featured: Battle of Barrosa - Rutherford B. Hayes - Kevin O'Halloran

Archive - By email - More featured articles...

Did you know...

From Wikipedia's newest content :

- ... that Kugelbake is the name of a series of tall wooden structures (current structure pictured) built at the mouth of the River Elbe for more than 300 years to aid mariners?
- ... that logical positivist A. J. Ayer believed that religious language was meaningless because it could not be verified empirically?
- ... that the Body of Proof episode " Dead Man Walking " guest starred Christina Hendricks as a potential love interest for her real-life husband 's character?
- ... that the Italian island of Montecristo , although 10.39 km 2 (4.01 sq mi) in area, is almost deserted, having only two stable inhabitants?
- ... that the 19th-century American female seminary movement, which aimed to give women educational opportunities, lent its name to a pair of similarly named institutions in Charleston, South Carolina , and Charlestown, Massachusetts ?
- ... that Malcolm X was nominated for an Academy Award in 1973? Archive - Start a new article - Nominate an article

In the news

- Vladimir Putin (pictured) is elected President of Russia for a third term.
- A series of explosions at an arms dump in Brazzaville , Republic of the Congo, kill at least 236 people and injure hundreds more.
- A train crash near Szczekociny , Poland, kills 16 people.
- A tornado outbreak in the Midwestern and Southeastern United States causes at least 39 fatalities.
- BP agrees to pay US$7.8 billion to plaintiffs affected by the Deepwater Horizon oil spill .
- English musician Davy Jones , a member of The Monkees , dies at the age of 66. Conflict in Syria - Wikinews - Recent deaths - More current events...

On this day...

March 6 : Independence Day in Ghana (1957)

- 1447 - Tomaso Parentucelli became Pope Nicholas V .
- 1834 - York , Upper Canada , was incorporated as Toronto .
- 1853 - Giuseppe Verdi 's La traviata premiered at Venice 's La Fenice , but the performance was so bad that it caused the Italian composer to revise portions of the opera.

Figure 1. Wikipedia as displayed by our FTL-generated layout engine

Our previous work on the Fast Tree Layout (FTL) synthesizer has shown how to find a traversal sequence given a declarative specification and implement it efficiently [5]. The goal here is to apply FTL to browser layout. In particular, we want to implement the CSS 2.1 Specification, a standard which defines the core functionality of modern layout engines. This involves two major components:

- Identifying extensions to FTL. I helped determine what kinds of extensions were necessary to support CSS—in particular, quadratic-time computations and subtree parallelism
- Reformulating CSS in terms of FTL. I expressed a significant subset of CSS in FTL, including cases where FTL and the standard have incompatible abstractions.

2.2 FTL Extensions

FTL’s input is a declarative specification which is based on attribute grammars. Attributes in a browser context are properties such as height that belong to each element in the document, and the specification states how to compute an attribute from those nearby it. This is a natural way to express many of the computations a layout engine must do.

However, ordinary attribute grammars are not powerful enough to express all layout operations. Some elements, like tables, must be represented as DAGs instead of trees. Also, attribute grammars perform simple linear-time loops over their children, making quadratic-time algorithms a challenge. For DAG support, we have constructs that allow a child to sometimes have more than one parent, such as a table cell which is a child of its row and column. To express quadratic-time algorithms, we allow attributes to have list values. We can then construct a list with one element for every node, and perform some action on the list at every node. These extensions integrate well with attribute grammars, showing that grammars as a formalism are extensible.

There are some layout operations that FTL can only perform sequentially. Certain web elements (specifically, floats and inlines) have dependencies that require a sequential inorder traversal to resolve. To prevent this traversal from becoming a bottleneck, we have added the ability to localize these dependencies to a subtree of the document and keep the overall traversal parallel. Ther is While inorder subtrees may be inherently sequential, we can run them in parallel with each other and with the rest of the document to still achieve scaling.

2.3 Implementing CSS in FTL

We are able to express a large amount of layout features using FTL's attribute-grammar-like language. Many aspects of CSS have a natural representation in attribute grammars. For example, to say that a node's child is left-aligned with the node itself, one might say `child.left := left`. We have used FTL to describe most of CSS, including text, images, block and line layout, floats, tables, and the CSS box model. We do not think there are fundamental issues preventing us from implementing other features.

There are cases where CSS is hard to express with FTL's language. For instance, CSS has the notion of a *line box* which contains a line of text. However, the rules for constructing a line box are not fully defined within CSS. By contrast, our grammar must define the full, explicit semantics of how to lay out a line of text, but has no notion of what a line box is.

3. Results

An FTL formulation of CSS allows us to generate a layout engine that is fully parallelizable. FTL produces a 9-pass sequence of traversals, and 8 of these are pre- or postorder traversals which can readily be parallelized. The other traversal is able to use subtree parallelism to . When laying out Wikipedia and the xkcd blog each, without optimization we achieved over a 3X speedup moving from 1 to 8 cores. We believe that this can be made to scale much better with further optimization.

In just over 1000 lines of code, we have implemented a layout engine that implements a large fraction of CSS. Our engine is complete enough to lay out real web pages, as shown in Figure 1. While imperfect, this rendering of the Wikipedia Home Page includes the page's core features such as tables, CSS boxes, lists, and text.

4. Conclusions and Future Work

We have shown that a rich layout engine that is highly parallel can be generated from an attribute-grammar-like specification. In the future I will be working with Mozilla to incorporate our ideas into the Servo project, whose purpose is to prototype future parallel browser technology. This will involve making our tree traversals work well with other components of the browser and adding incremental optimizations.

References

[1] Firefox parallel parsing.

[2] C. Cascaval, S. Fowler, P. Montesinos-Ortego, W. Piekarski, M. Reshadi, B. Robatmili, M. Weber, and V. Bhavsar. Zoomm: a parallel web browser engine for multicore mobile devices. In *Proceedings of the 18th ACM SIGPLAN symposium on Principles and practice of parallel programming*, PPoPP '13, pages 271–280, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1922-5. doi: 10.1145/2442516.2442543. URL `http://doi.acm.org/10.1145/2442516.2442543`.

[3] C. G. Jones, R. Liu, L. Meyerovich, K. Asanović, and R. Bodík. Parallelizing the web browser. In *Proceedings of the First USENIX conference on Hot topics in parallelism*, HotPar'09, pages 7–7, Berkeley, CA, USA, 2009. USENIX Association. URL `http://dl.acm.org/citation.cfm?id=1855591.1855598`.

[4] L. A. Meyerovich and R. Bodik. Fast and parallel webpage layout. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 711–720, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772763. URL `http://doi.acm.org/10.1145/1772690.1772763`.

[5] L. A. Meyerovich, M. E. Torok, E. Atkinson, and R. Bodik. Parallel schedule synthesis for attribute grammars. In *Proceedings of the 18th ACM SIGPLAN symposium on Principles and practice of parallel programming*, PPoPP '13, pages 187–196, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1922-5. doi: 10.1145/2442516.2442535. URL `http://doi.acm.org/10.1145/2442516.2442535`.