



SUPERCONDUCTOR

Synthesis for Parallel Browsers

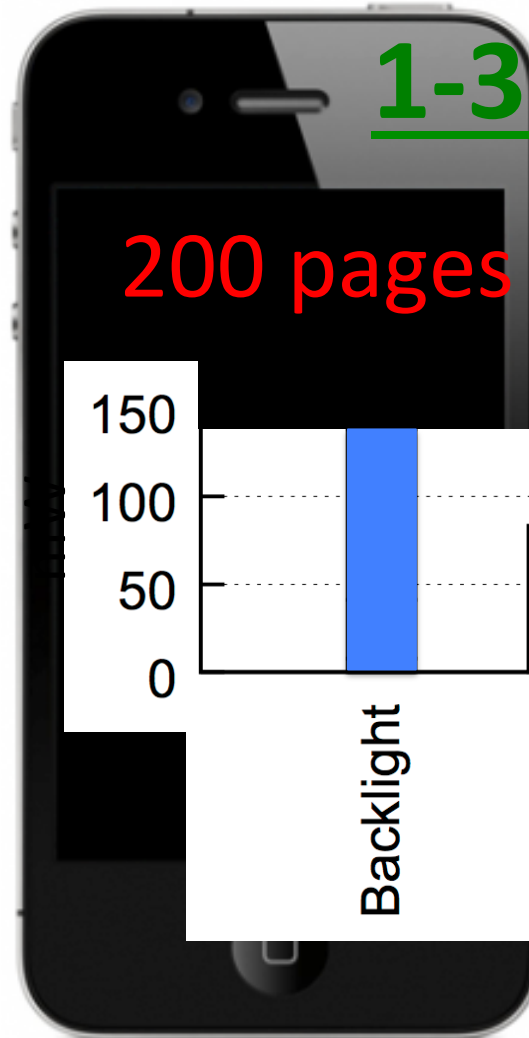
Leo Meyerovich, Matthew Torok,
Eric Atkinson, Rastislav Bodik
Parallelism Lab, UC Berkeley

IFIP WG on Language Design
Austin, Texas, 2012

Future
Browsers

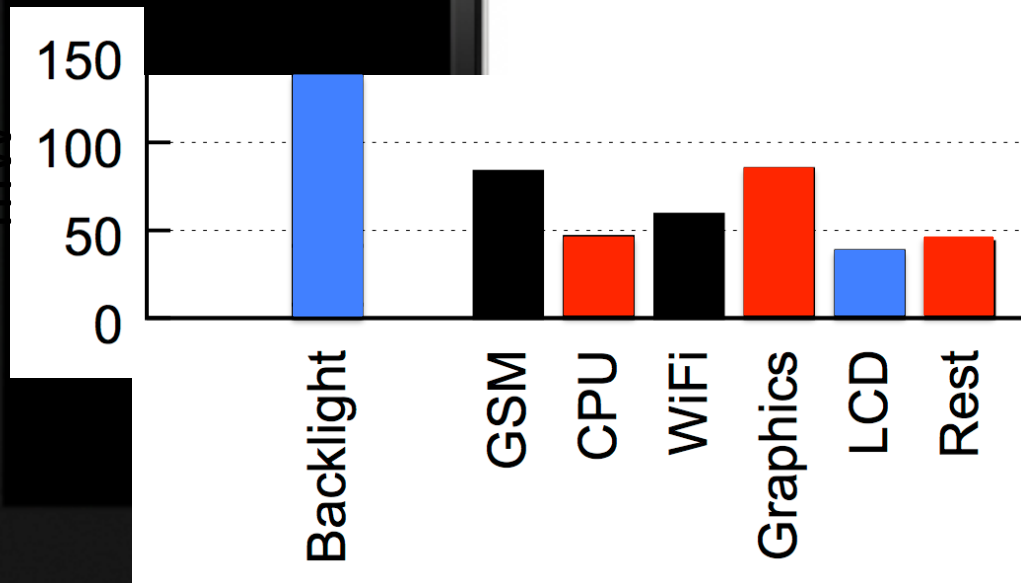
Synthesis for Layout &
Parallelism

Why Optimize the Browser: **Power**

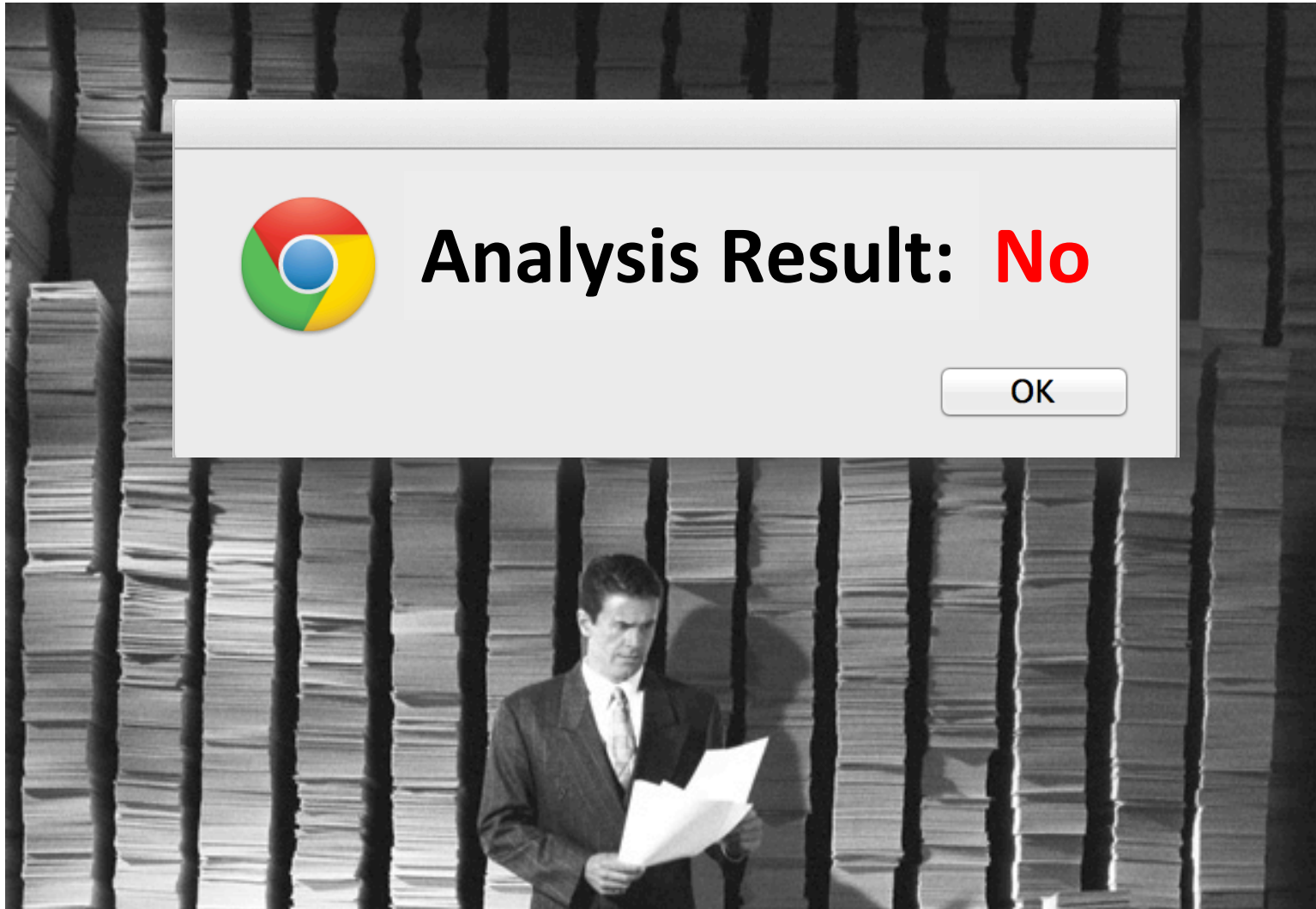


1-3 Watts

200 pages / battery cycle



Why Optimize the Browser: **Speed**

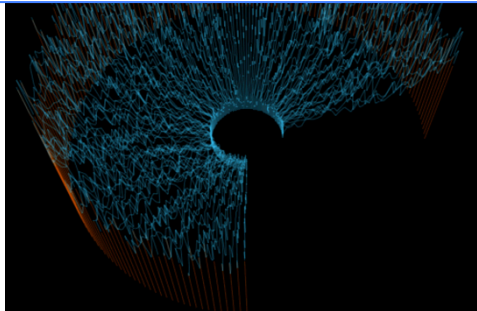
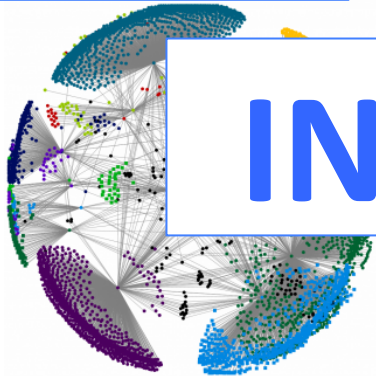
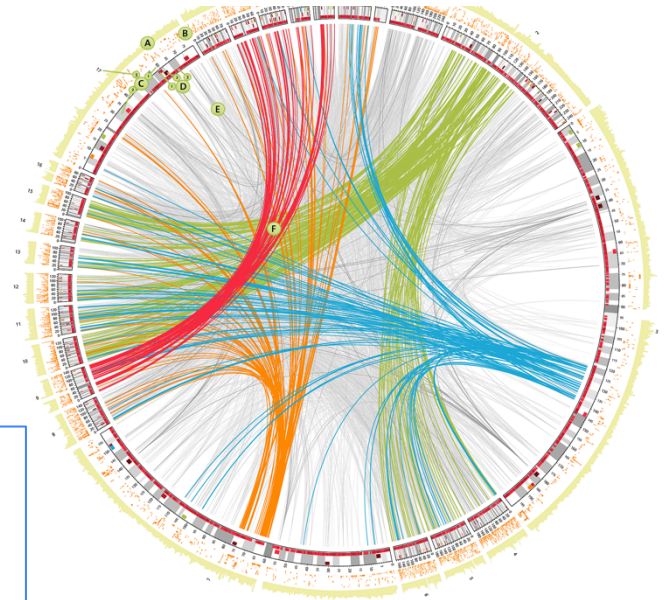
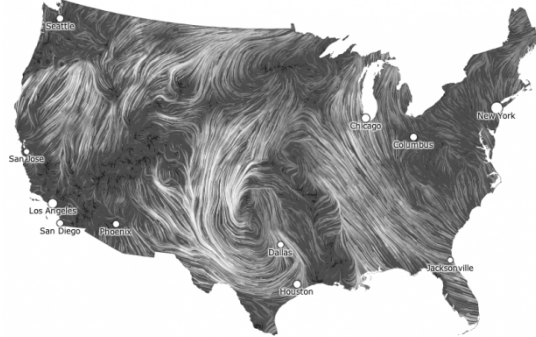
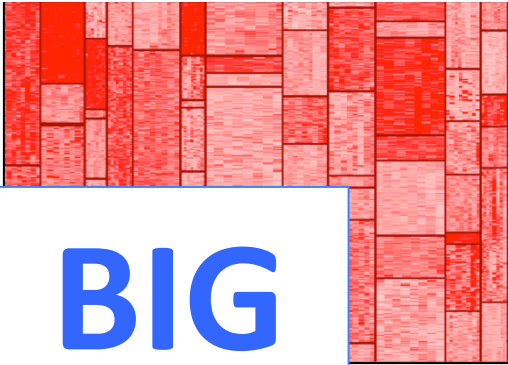


Why Optimize the Browser: **Speed**

BIG

INTERACTIVE

VISUALIZATIONS



Why Optimize the Browser: **Speed**



Visualization dead slow when using a large dataset?

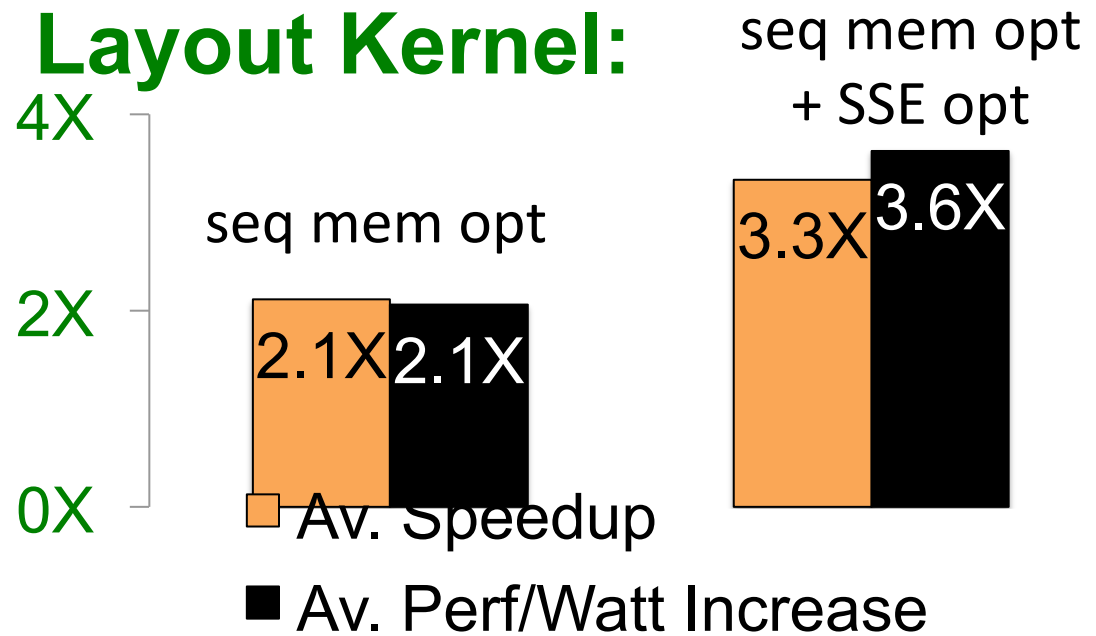
▲
2
▼ ... For anything < 5K, the [JavaScript]
D3 library works wonders ...

Choose: **big** *interactive*
 but **static** but **small**

Parallelism for Energy & Speed

- 4 CPUs cores x 128b SIMD
- 4-8 GPGPU cores
- “Race-to-halt”: time \approx energy

Layout Kernel:



What to Parallelize?

HotPar 09

HotPar 2011, WWW 2010, PPOPP 2013

Parsing

DOM

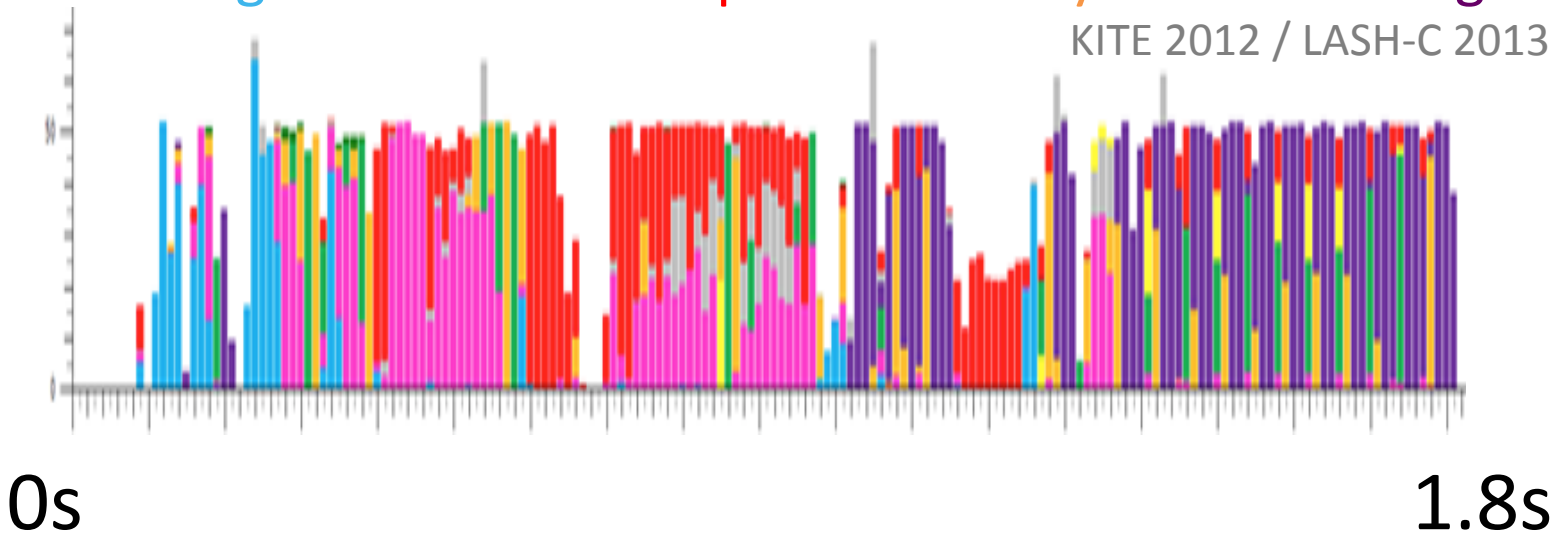
JavaScript

Layout

Rendering

KITE 2012 / LASH-C 2013

Browser
CPU
activity



Everything

Intermezzo: Russian Election Demo

SUPERCONDUCTOR

Big Data Visualization Demo: Treemap

A recent national election in an unnamed country had results which cemented the ruling party's power. However, international election monitors have called into question the validity of these results. Luckily for us, the country released raw data from its polling stations, allowing us to explore this data and decide for ourselves.

With our visualization system, **Superconductor**, we can interactively explore the entirety of the data set, without resorting to simplifications, summaries, or offline calculation.

Presented is a treemap built from the data taken from every single polling station in the country — **94,601 in total**. Each tiny red rectangle represents one polling station; its size corresponds to the number of votes cast there; its color corresponds to the percentage of those votes cast for the ruling party (the more red, the higher the percentage.) Polling places are grouped by region, inside the dark red outlines.

You can manipulate the sliders along the top to filter shown polling places by turnout (the percentage of votes cast versus the number of registered voters assigned to a polling station.) The 'span' slider controls the range of polling stations shown, the 'average' slider controls the turnout the span extends around (e.g., a span of ± 5 and an average of 50% will show only polling places with 45-55% turnout.) A counter along the bottom provides a live count of the percentage of votes for the ruling party for the polling places currently shown.

Try lowering the span to small number, then increasing the average. Watch the size of the treemap (indicating votes cast at polling places within your range) and amount of redness on the map.

localhost:8000/examples/election-treemap/

Turnout Span: $\pm 50\%$ Average Turnout: 50%

Zoom

Ruling Party Share: 50%

Highlighting: none

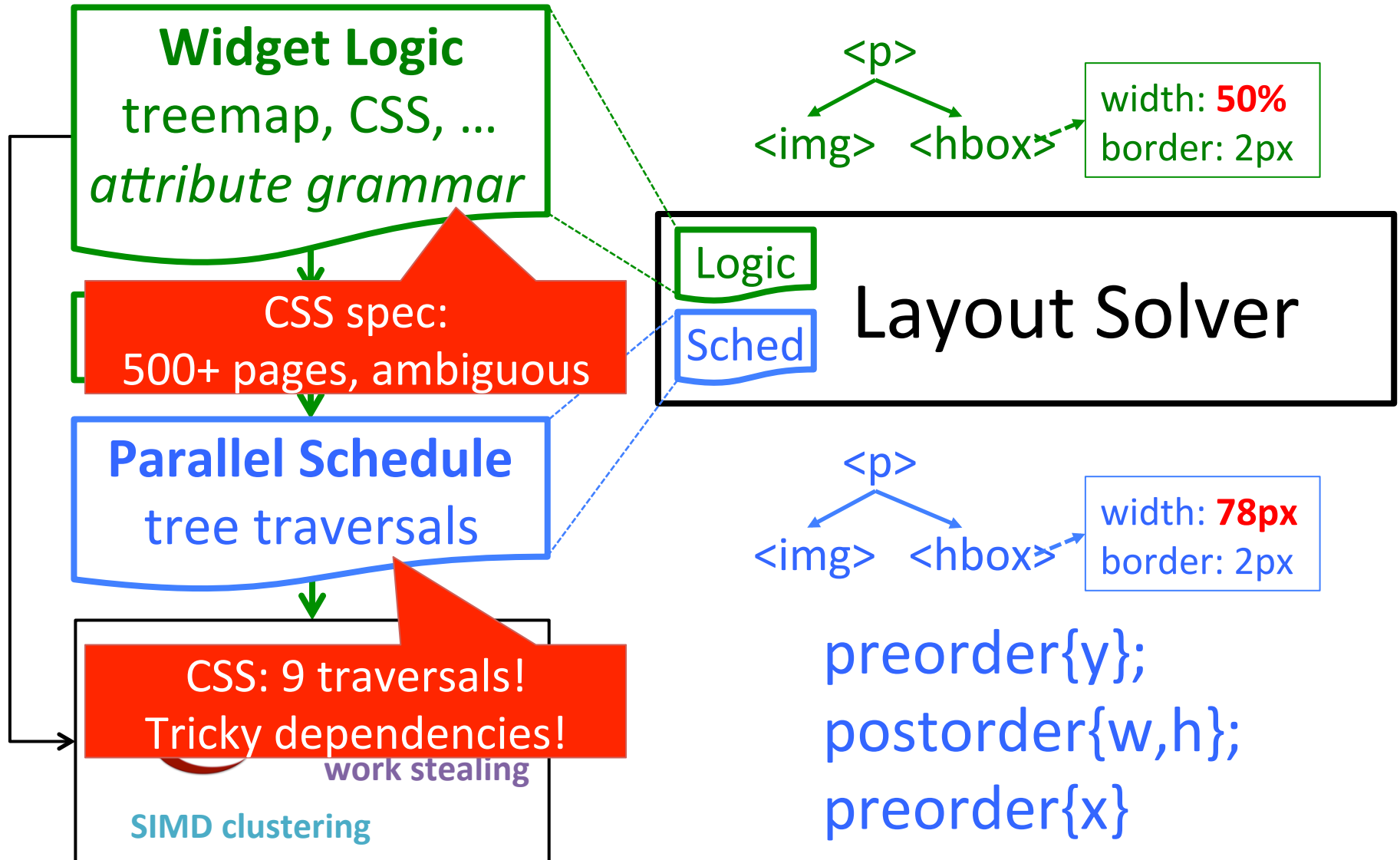
24 FPS (0-24)

The Parallel Browser



Synthesis for **Layout &
Parallelism**

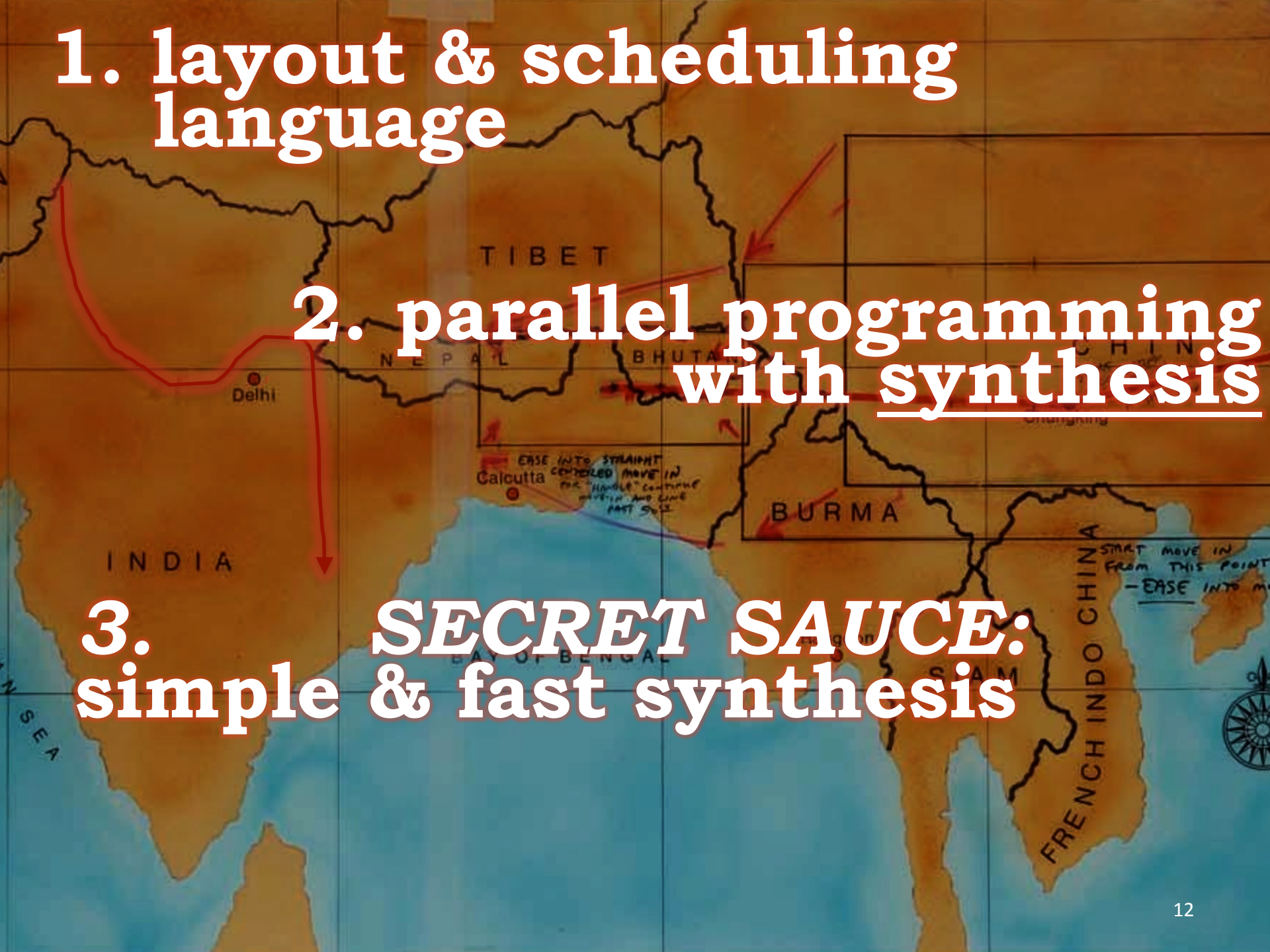
Writing Parallel Layout Widgets is Hard



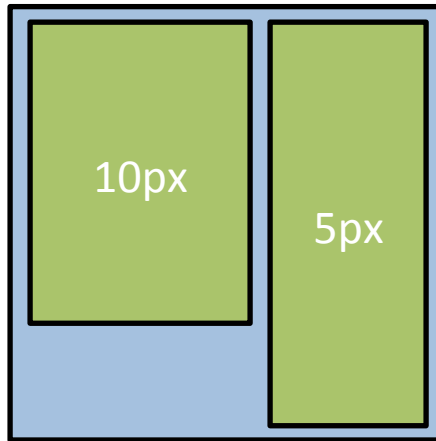
**1. layout & scheduling
language**

**2. parallel programming
with synthesis**

**3. *SECRET SAUCE:*
simple & fast synthesis**



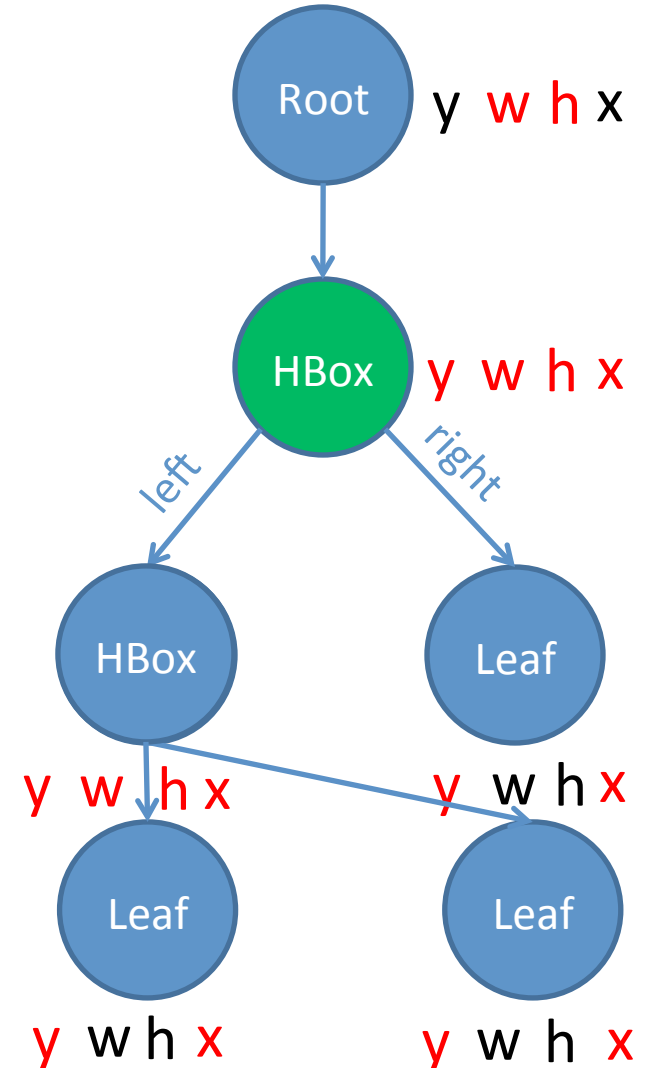
Functional Language: HBox Example



inputs

vars

```
class HBox(PaintRect) : Box
  child left, right : Box
  w := left.w + right.w
  h := max(left.h, right.h)
  left.x := x
  right.x := x + left.w
  left.y := y
  right.y := y
```



Efficient Schedule Example: HBox

postorder{w,h}; preorder{x,y}

pass1_postorder_hbox():

w := left.w + right.w

h := ...

pass2_preorder_hbox():

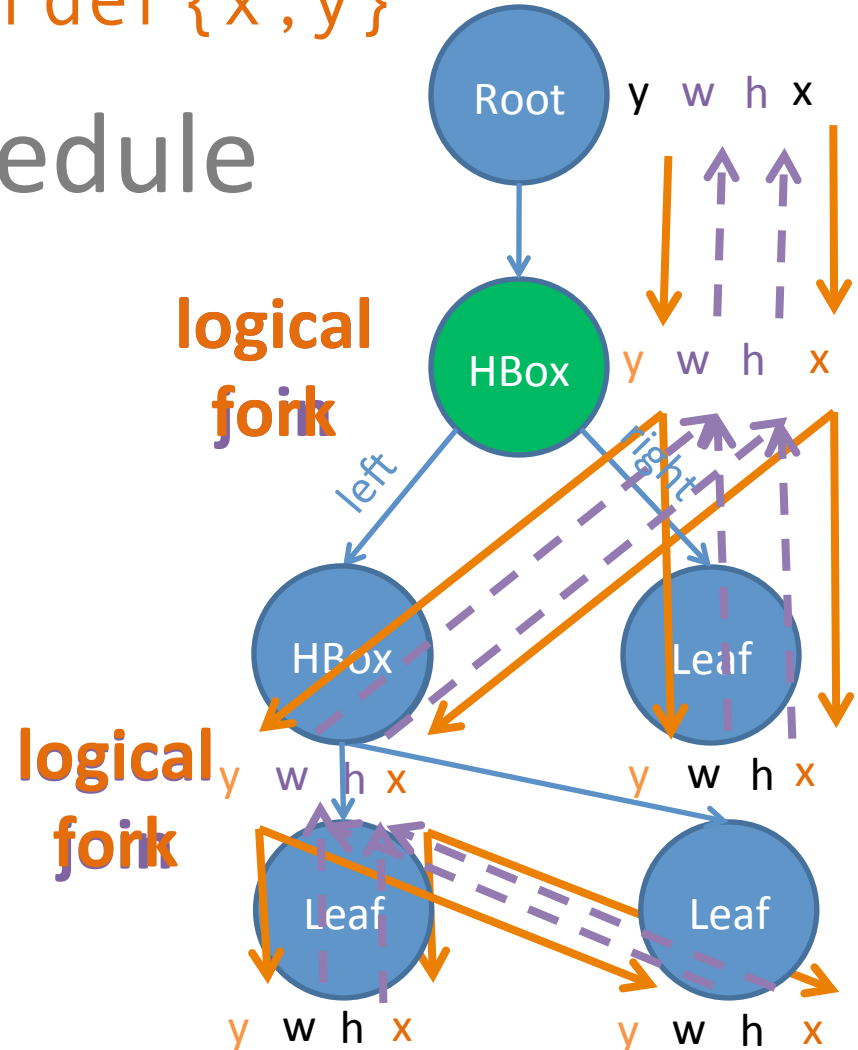
left.x := ...

right.x := ...

left.y := ...

right.y := ...

explicit input schedule



Efficient Schedule Example: HBox

postorder{w,h}; preorder{x,y}

pass1_postorder_hbox():

w := left.w + right.w

h := ...

pass2_preorder_hbox():

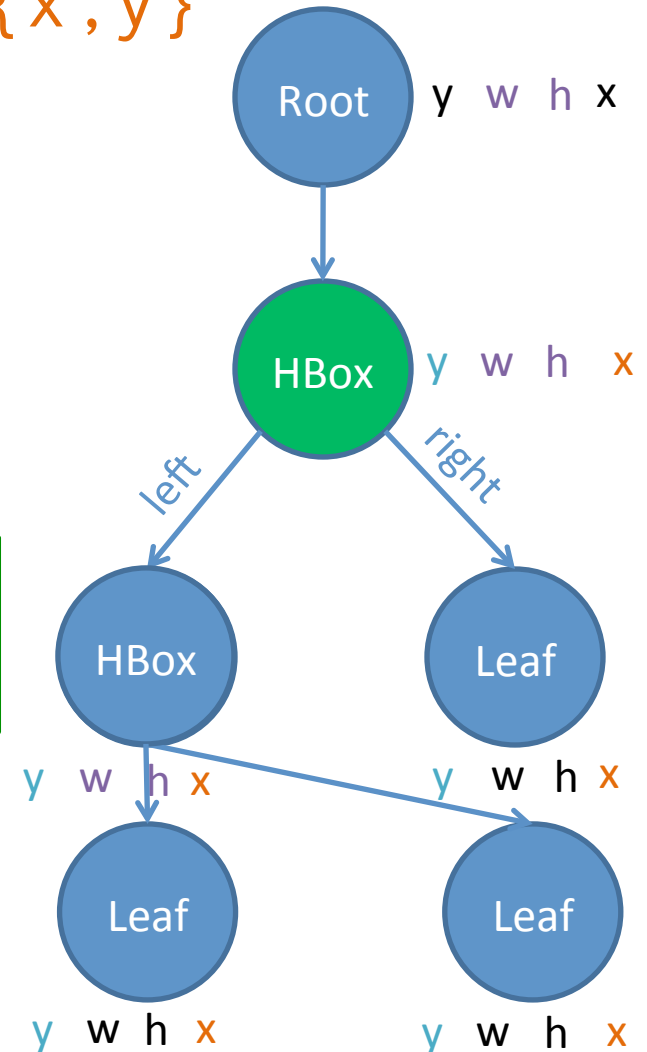
left.x := ...

right.x := ...

left.y := ...

right.y := ...

Schedule sound
for \forall input trees



Efficient Schedule Example: HBox

postorder{w,h}; preorder{x,y}

pass1_postorder_hbox():

w := left.w + right.w

h := ...

pass2_preorder_hbox():

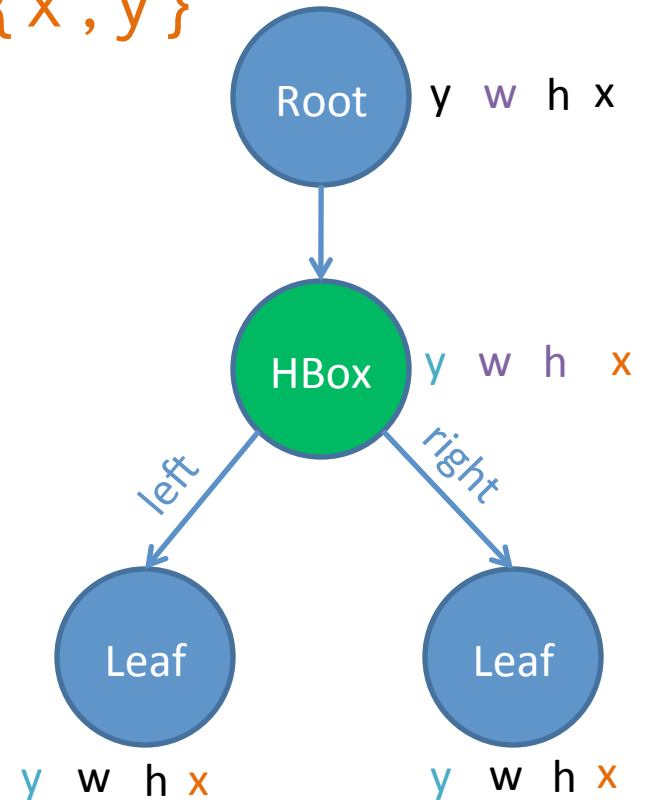
left.x := ...

right.x := ...

left.y := ...

right.y := ...

Schedule sound
for \forall input trees



Scheduling Tricky Features?

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

Mozilla Challenge Problem

#1

- Arts
- Biography
- Geography

Today's featured article



Nested (*rich*) text

Banksia cuneata is an endangered species of flowering plant in the Proteaceae family. Endemic to southwest Western Australia, it belongs to the subgenus *Leostylis*, which contains three closely related species with flower clusters that

Tables (DAGs), Floats (value speculation), ...

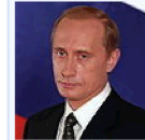
classified as endangered, surviving in fragments of remnant bushland in a region which has been 93% cleared for agriculture. As *Banksia cuneata* is killed by fire and regenerates from seed, it is highly sensitive to bushfire frequency; fires recurring within four years could wipe out populations of 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...

In the news



- [Vladimir Putin](#) (pictured) is elected president of Russia
- [A series of explosions](#) at an airport in Moscow

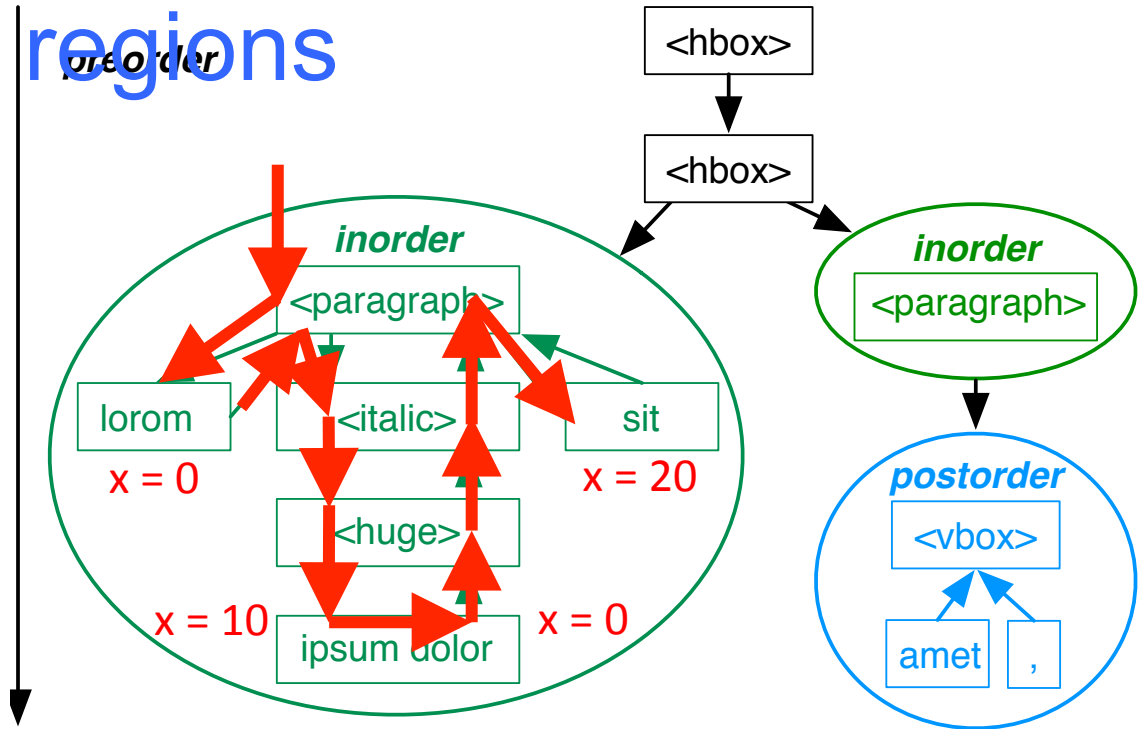
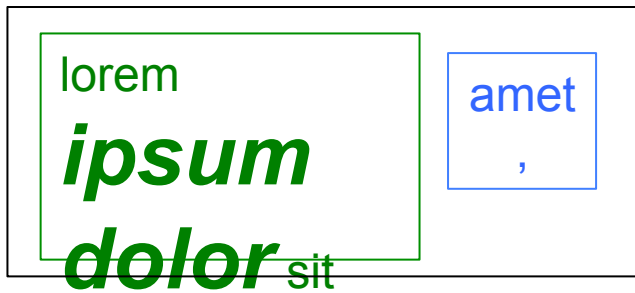
On this day...

March 6 : Independence Day in the United States



Isolate Dependencies into Subgrammars

parallel across
regions



sequential

$child[i].x = f(child[i-1].x)$

parallel
region

Isolate Dependencies into Subgrammars

Static Partitioning

preorder
<hbox>, ...

inorder

<paragraph>

<italic>

<huge>

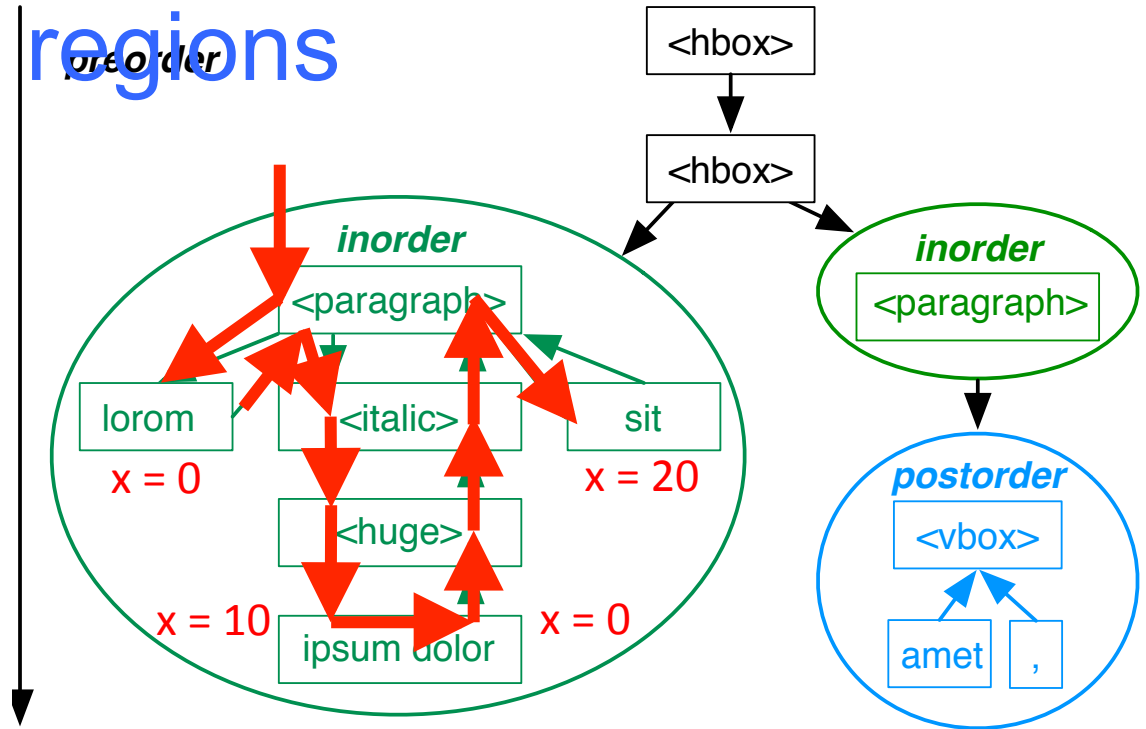
<word>

postorder

<vbox>

<word>

parallel across regions



sequential

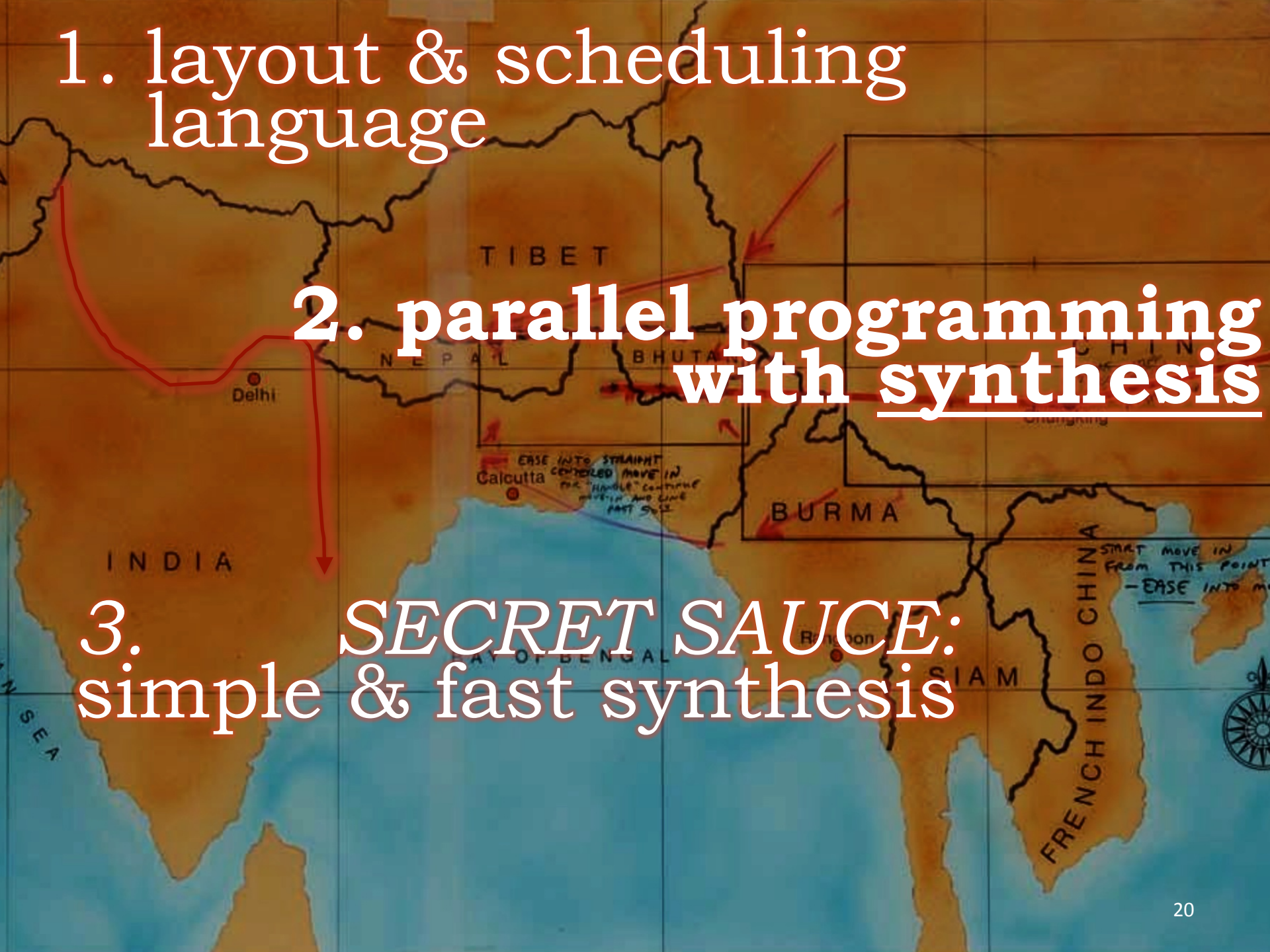
parallel region

* scheduling is still global

1. layout & scheduling
language

2. parallel programming
with synthesis

3. *SECRET SAUCE*:
simple & fast synthesis



Sketching Parallel Programs

Lightweight: `postorder{w,h};` `??{??}`

explicit parallelization (gets checked) →

automatic parallelization →

+ Prolog EDSL for *semantic* constraints: $\in, \cap, p(x), \dots$

1. Maintenance: “Don’t break my parallelization!”

2. Exploratory: “Partitioning for nested text?”

parallel Algorithm = *AG* Logic + Control *sketch*

Robert Kowalski, CACM, 1979

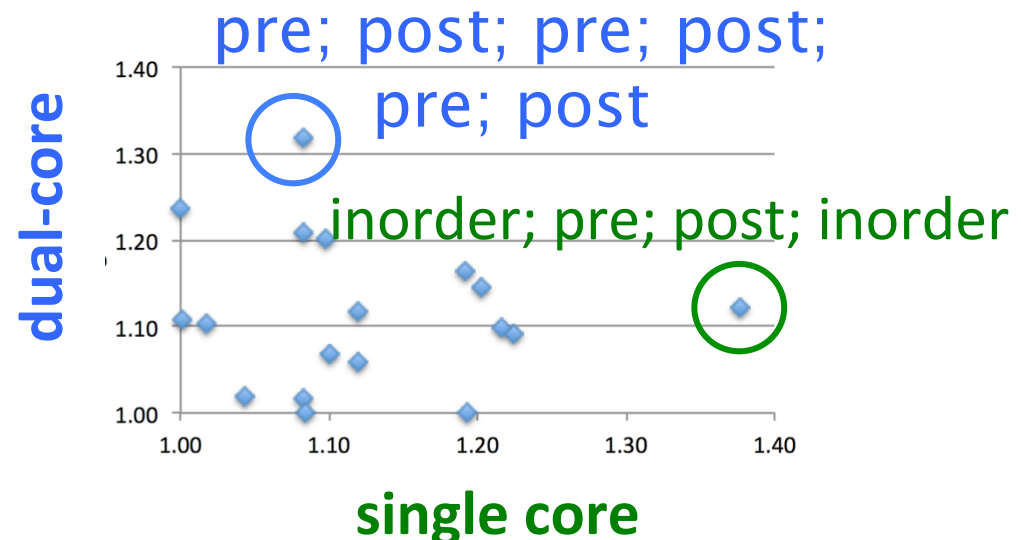
Third Use: Autotuning Schedules

Multiple concretizations of “??” are sound!

1. `postorder{w,h}; preorder{x,y}`
2. `postorder{w} || postorder{h}; ??{x,y}`

Autotune for best on real hardware

HBox++ Speedup
each axis normalized to
worst sched on its hw



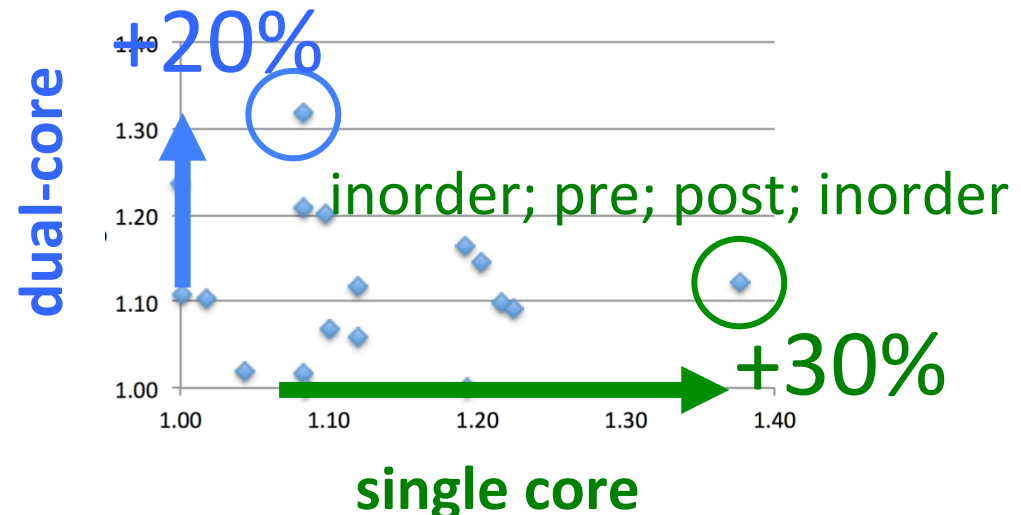
Third Use: Autotuning Schedules

Multiple concretizations of “??” are sound!

1. `postorder{w,h}; preorder{x,y}`
2. `postorder{w} || postorder{h}; ??{x,y}`

Autotune for best on real hardware

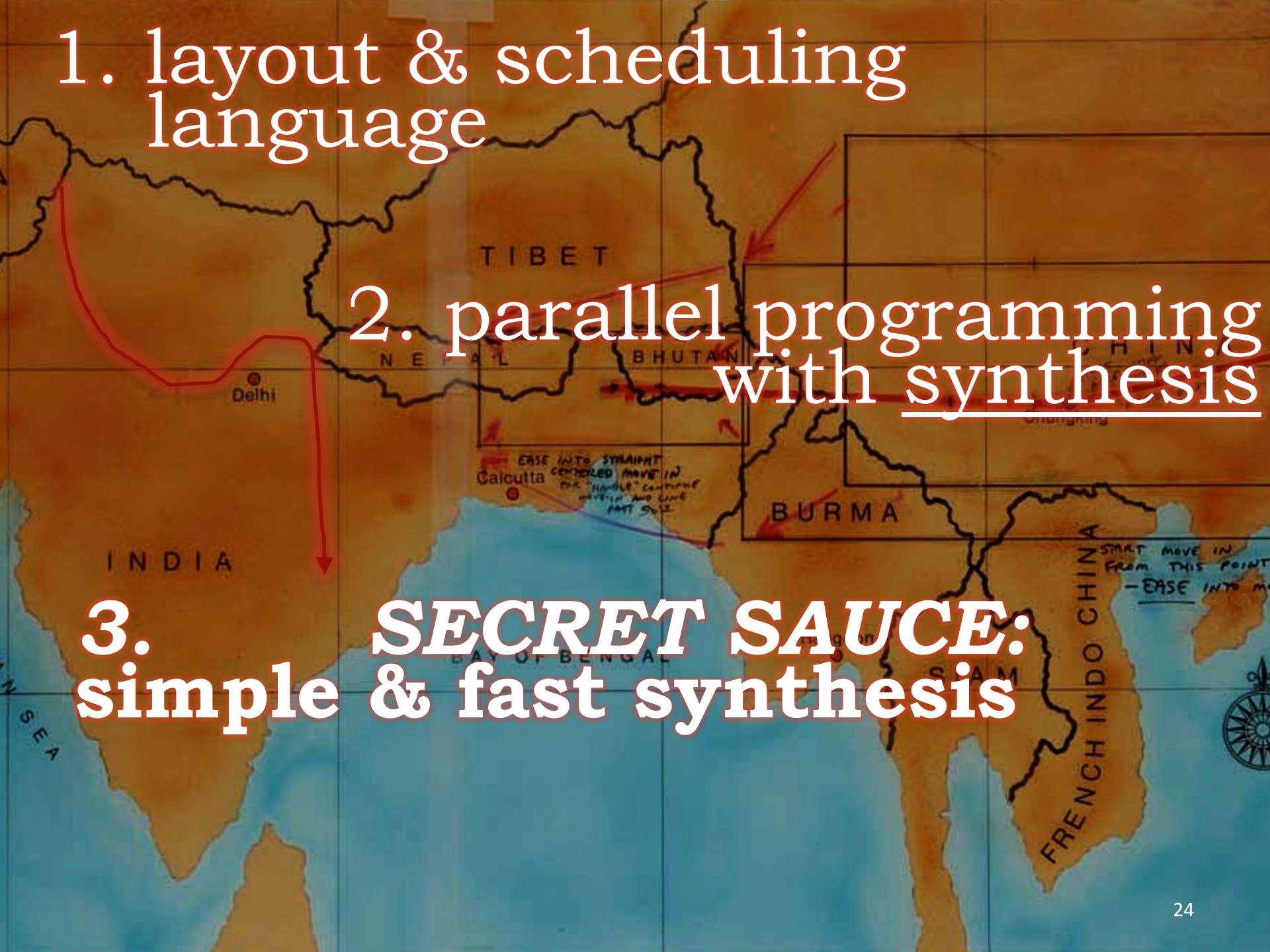
HBox++ Speedup
each axis normalized to
worst sched on its hw



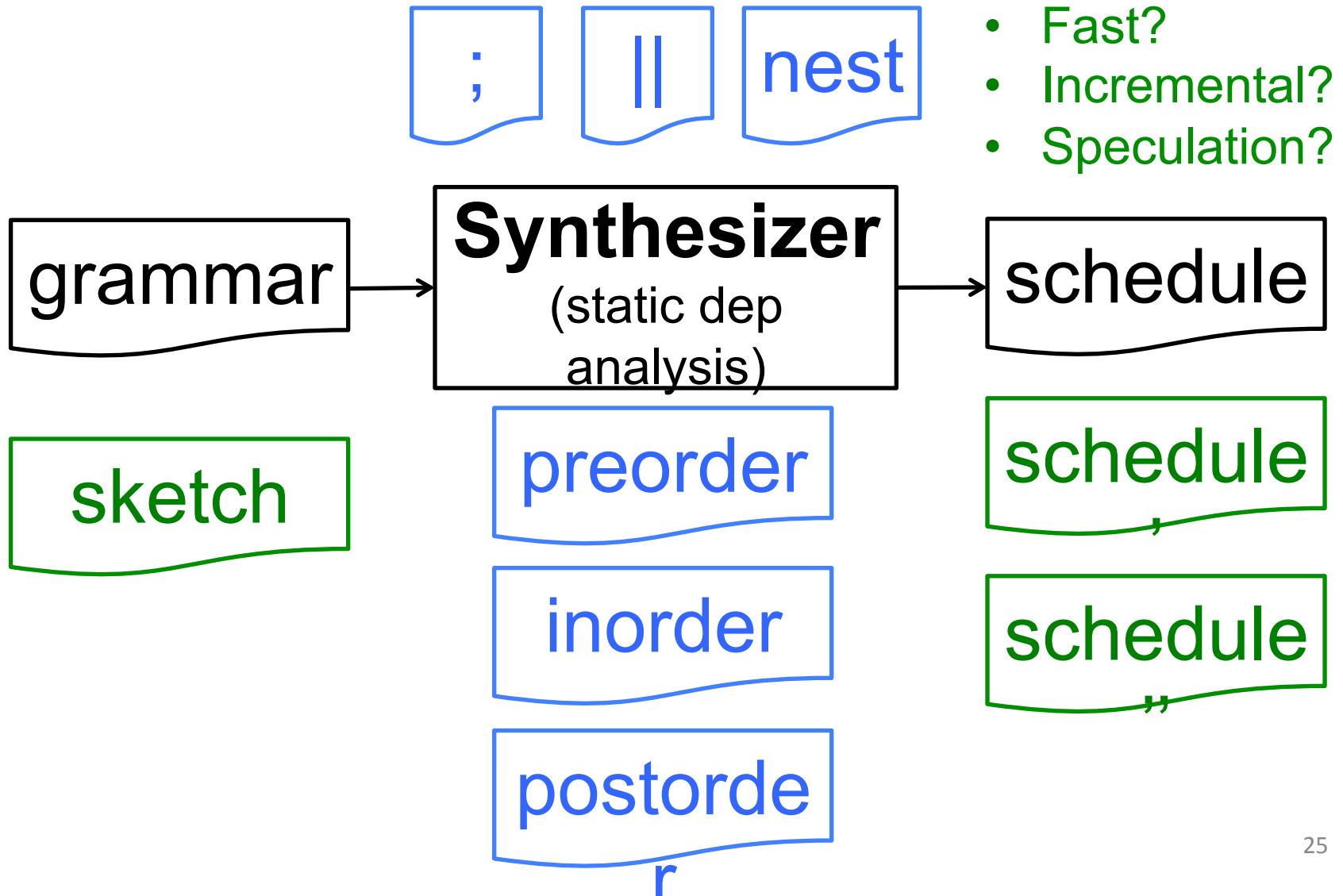
1. layout & scheduling
language

2. parallel programming
with synthesis

3. **SECRET SAUCE:**
simple & fast synthesis



An Extensible Synthesizer



Synthesis: Modular & $O(A^3)$

Enumerate Sched Prefixes

syntactic

`postorder{w, h, x, y}`

`preorder{w, h, x, y}`

`preorder{w, h}`

`... ; postorder{x, y}`

$O(A^2)$ search framework
(sketches, incremental,
iterative refinement, ...)

Check

pattern dep. check

X

X

✓: continue

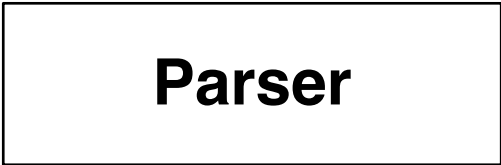
✓: all done.

**Pattern writer provides
modular $O(A)$ checkers**

Parallel Algorithms: in a JS framework!

[Speculation: HotPar 2009]

multicore



workers

[Memory: WWW 2010]

multicore



workers

[Scheduling: HotPar 2011, PPOPP 2013]

SIMD GPU multicore



webCL

New! Pipelining

GPU multicore



webGL

New Directions

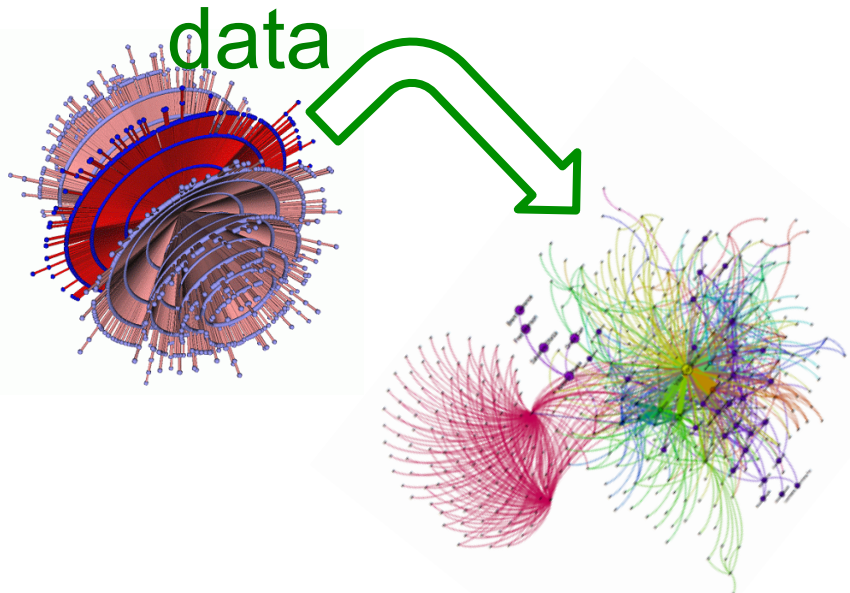
Smart Tools

- programming by demonstration

Big Data Viz

- graph algorithms
- views of server data

★ Chris Jones , Rose Liu , Leo Meyerovich , Krstje Asanovic , Rastko Parallelizing the Web Browser. HotPar '09 [show abstract], pdf
★ Leo A. Meyerovich and Ariel S. Rabkin . Socio-PLT: Sociological Principles for Programming Lan Draft. [show abstract], pdf
Leo A. Meyerovich , Arjun Guha , Jacob Baskin , Gregory H. C Shrinam Krishnamurthi . Flapjax: A Programming Language for Ajax Application OOPSLA 2009 [show abstract], <i>best student paper</i> , pdf ,



Takeaways

Parallel browser for energy & speed

Architecture

- new || algorithms + code gens
- native + “browser-in-a-browser” lib

Synthesis

- parallelization sketches
- CSS 2.1 spec
- **future:** graphs, clusters, smart tools

preorder{??}

Schedule Synthesis is $O(A^3)$

sketch = preorder{??} ; ??

sketching

postorder{w, h, x, y}

X: preorder

preorder{w, h, x, y}

X: !sat={x, y}

preorder{w, h}

✓: rem={x, y}

... ; postorder{x, y}

✓: rem={}

incrementalization

refinement

$O(A)$ passes * $O(A)$ refinements * $O(A)$ check = $O(A^3)$

(topological sort of static dependency graph)

Schedule Language Example: HBox

preorder{y}; postorder{w,h}; preorder{x}

pass0_preorder_hbox():

child1.y := ...

child2.y := ...

pass1_postorder_hbox():

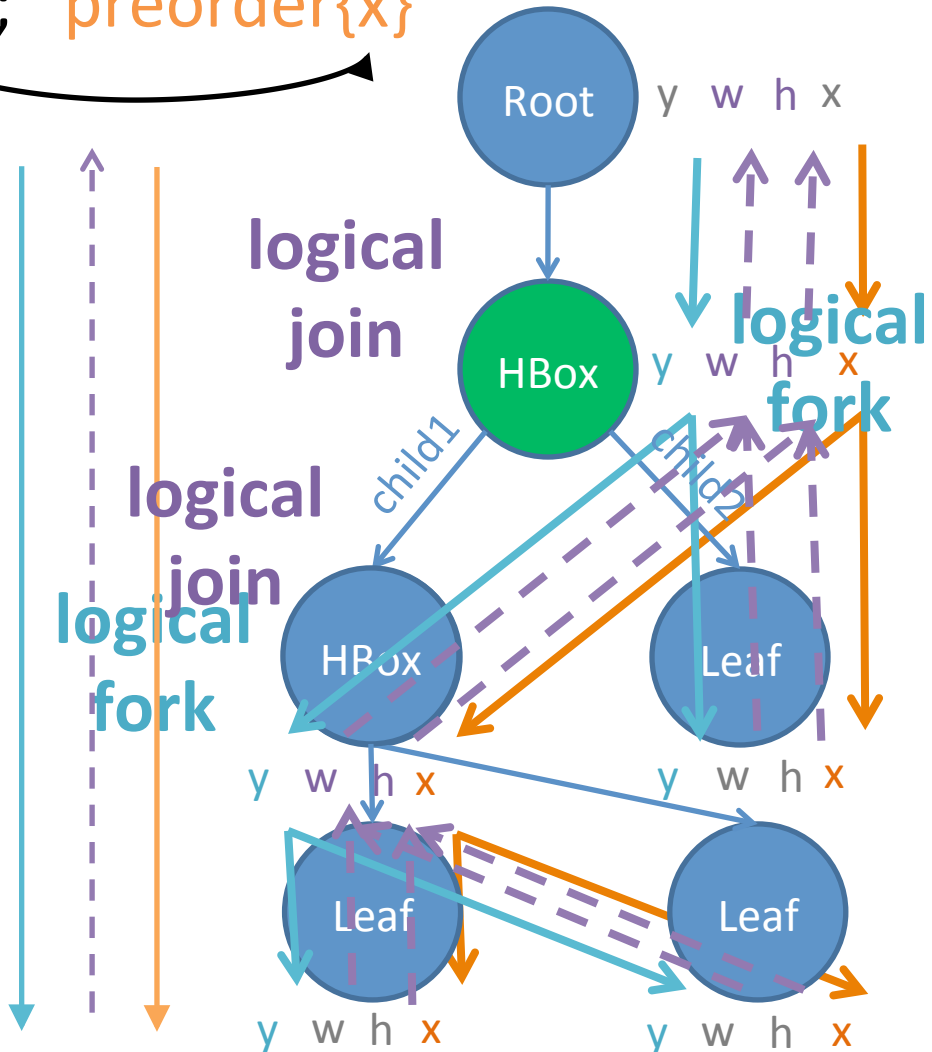
w := ...

h := ...

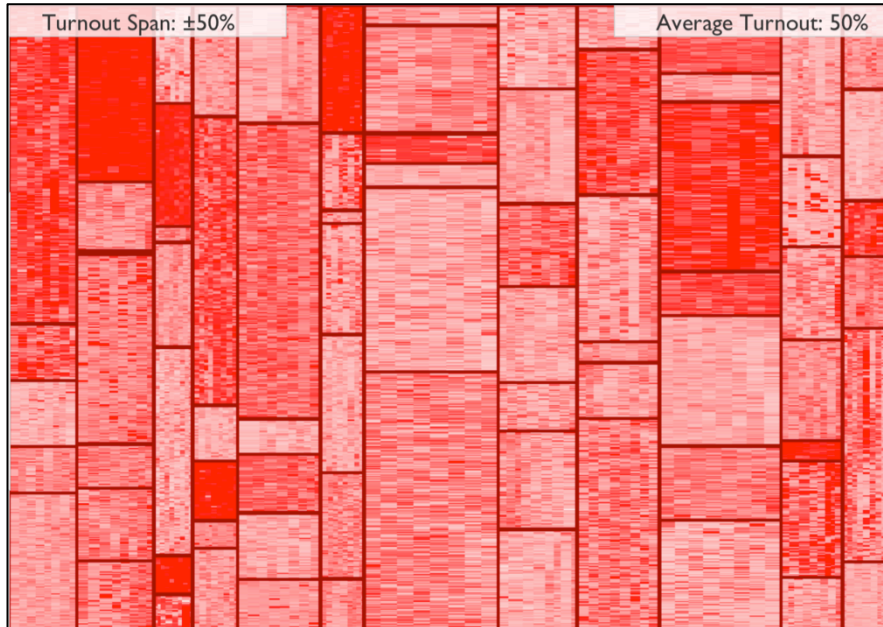
pass2_preorder_hbox():

child1.x := ...

child2.x := x + child1.w



Focus: Writing Parallel Layout Widgets




Main Page
From Wikipedia, the free encyclopedia

Jump to: [navigation](#) , [search](#)


Welcome to Wikipedia ,
the free encyclopedia that anyone can edit .
3,800,580 articles in English

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 1890, 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 character?

In the news

- [Vladimir Putin](#) (pictured) is el
- [A series of explosions at an ar](#) people and injure hundreds me
- [A train crash near Szczekocin](#)
- [A tornado outbreak in the Mid](#) fatalities.
- [BP agrees to pay US\\$7.8 billio](#)
- [English musician Davy Jones](#) .
- [Wikinews - Recent deaths](#) - [M](#)

On this day...
March 6 : [Independence Day in](#)


• 1447 - [Tomaso Parentucelli](#) be

• 1834 - [York](#) , [Upper Canada](#) , w

• 1853 - [Giuseppe Verdi](#) 's [La tr](#) so bad that it caused the [Ital](#)

• 1899 - [German chemical and p](#)

• 1945 - [Petru Groza](#) (pictured) c

• [Communist Party](#) -dominated g

• 1964 - In a radio broadcast, [Ni](#) boxer [Cassius Clay](#) would chan

New Layouts

Treemap, circos, time series

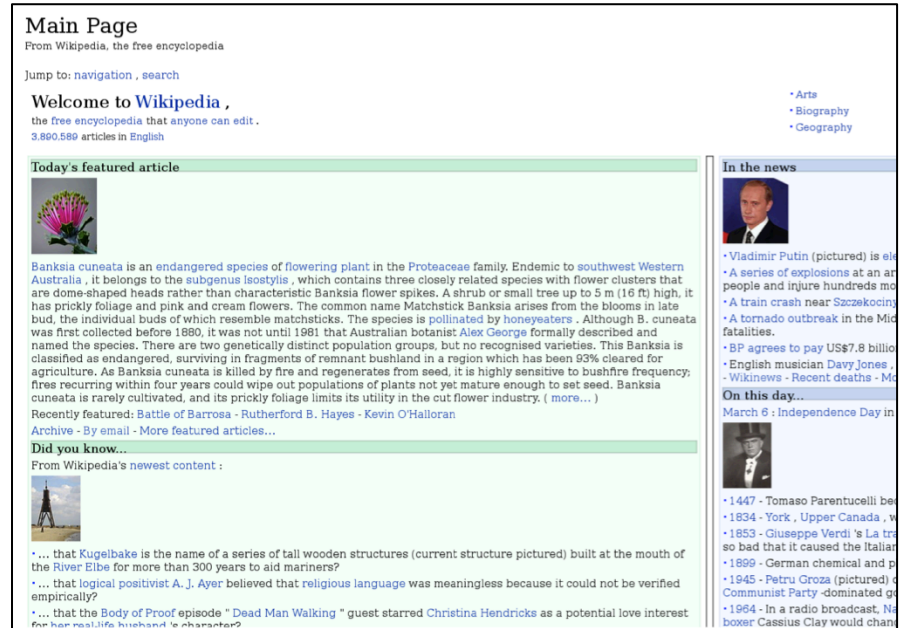
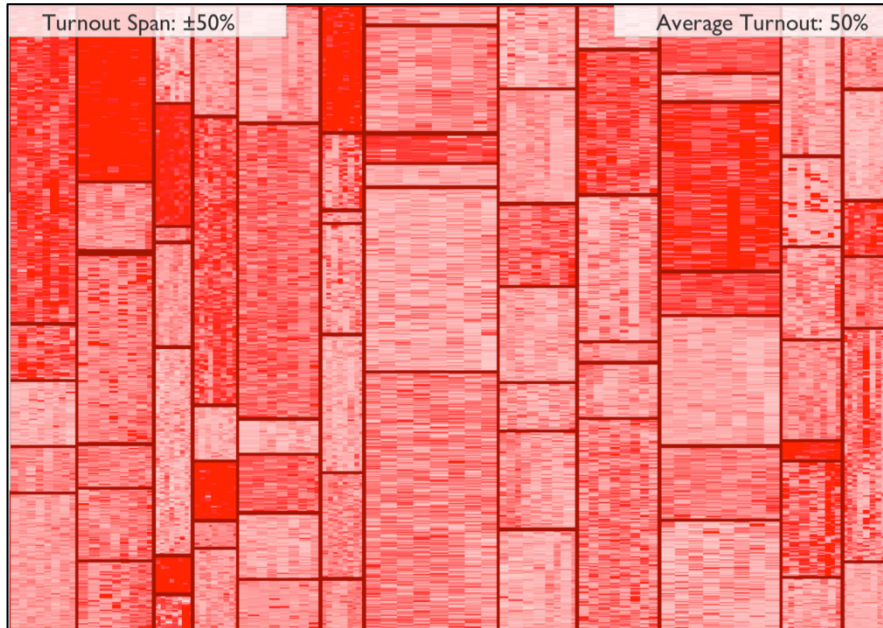
CSS 2.1:

floats, nested text, tables,

...

[Tricky: Qualcomm, Mozilla, ...]

Studies: GPU Big Data Viz & Multicore CSS



Treemap, circos, time series

- 1,000,000 nodes on GPU
- WebCL + JS libraries

CSS: floats, nested text, tables

- 9 pass sketch: synth in 65s
- 3x multicore speedup

Schedule Language Example: HBox

preorder{y}; postorder{w,h}; preorder{x}

pass0_preorder_hbox():

child1.y := y

child2.y := y

Schedule sound
for \forall input trees

pass1_postorder_hbox():

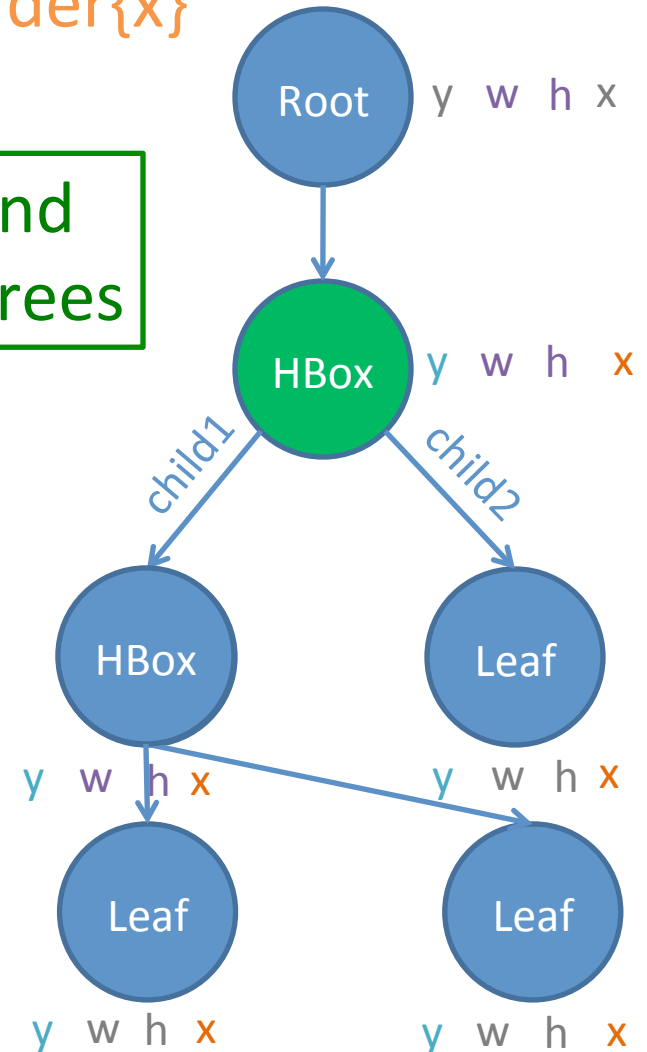
w := child1.w + child2.w

h := max(child1.h, child2.h)

pass2_preorder_hbox():

child1.x := x

child2.x := x + child1.w



Schedule Language Example: HBox

preorder{y}; postorder{w,h}; preorder{x}

pass0_preorder_hbox():

child1.y := y

child2.y := y

Schedule sound
for \forall input trees

pass1_postorder_hbox():

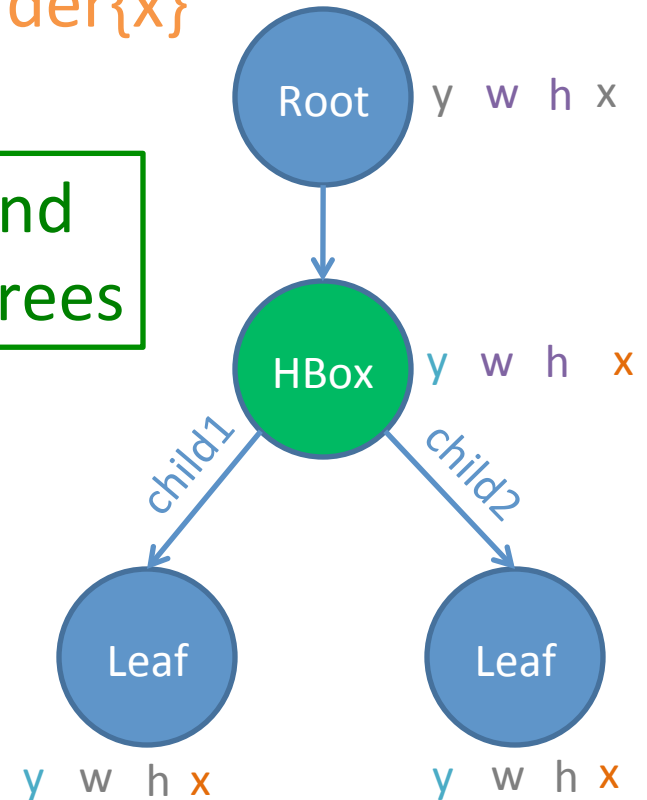
w := child1.w + child2.w

h := max(child1.h, child2.h)

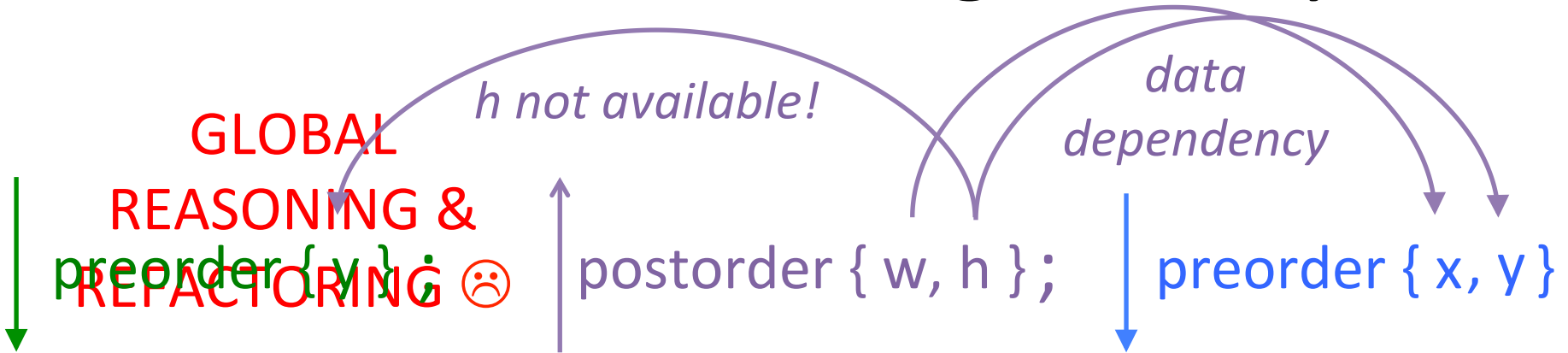
pass2_preorder_hbox():

child1.x := x

child2.x := x + child1.w



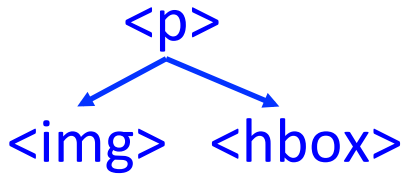
Manual Scheduling is Tricky



```
class HBox : Box
  w := ...
  h := ...
  child1.x := ...
  child2.x := x + child1.w
  child1.y := ...
  child2.y := ...
```

```
class VBox : Box
  h := ...
  w := ...
  child1.y := ...
  child2.y := y + child1.h
  child1.x := ...
  child2.x := ...
```

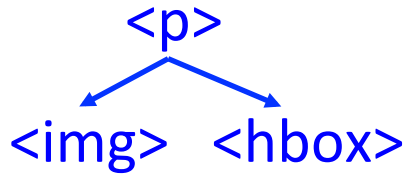
Browser Architecture



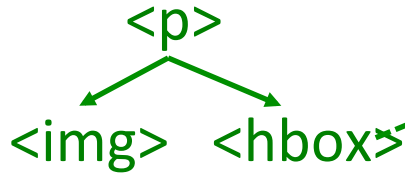
```
p hbox {  
  width: 50%  
  border: 2px  
}
```



Browser Architecture



```
p hbox {  
  width: 50%  
  border: 2px  
}
```



```
width: 50%  
border: 2px
```



Parser

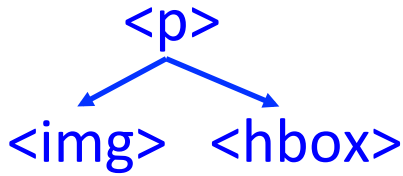


**HTML
+ CSS**

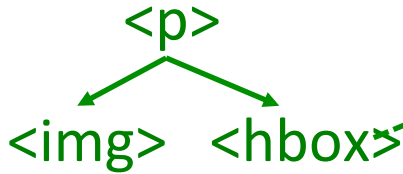


Selectors

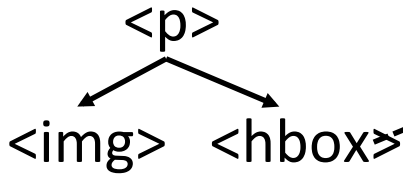
Browser Architecture



```
p hbox {  
  width: 50%  
  border: 2px  
}
```



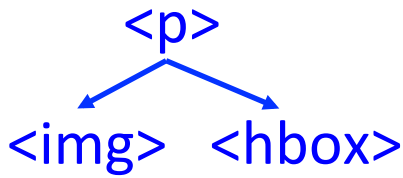
width: **50%**
border: 2px



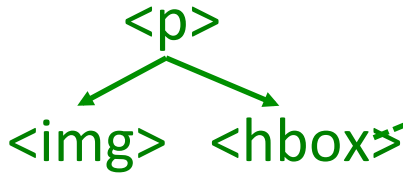
width: **78px**
border: 2px



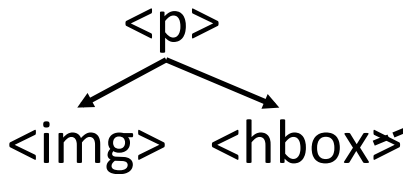
Browser Architecture



```
p hbox {  
  width: 50%  
  border: 2px  
}
```



```
width: 50%  
border: 2px
```



```
width: 78px  
border: 2px
```

