

# CSE 633 Parallel Algorithms Fall 2010

Computer Science and Engineering

University at Buffalo

John Longanecker

Parallel Computing with Browsers

Bin Packing Edition

November 4<sup>th</sup>, 2010

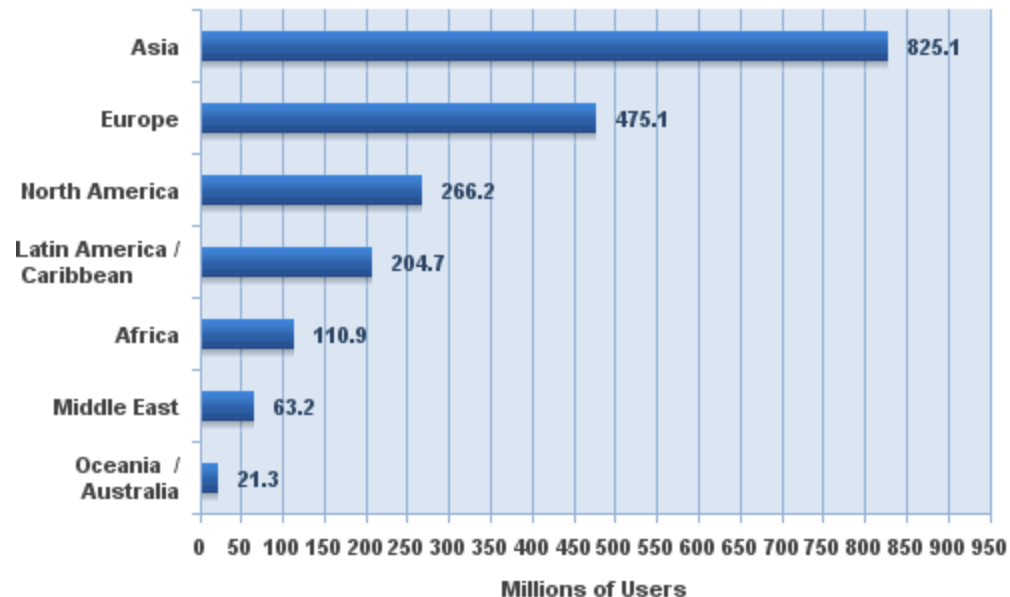
# Itinerary

- Description of the browser
- Advantages
- Disadvantages
- How I plan to harness it
- Packing Algorithm
- Technologies used
- Results
- Scheduling

# Imagine..

- Nearly 2 Billion computes worth of idle CPU Time

**Internet Users in the World  
by Geographic Regions - 2010**



Source: Internet World Stats - [www.internetworldstats.com/stats.htm](http://www.internetworldstats.com/stats.htm)  
Estimated Internet users are 1,966,514,816 on June 31, 2010  
Copyright © 2010, Miniwatts Marketing Group

- Huge Potential

# Browsers



# Browsers

- HTML – Markup Language
- HTML is not a programming language
- One native scripting language JavaScript
- Standards exists but not followed well

# Web Architecture

- Server side
- Client side
- Client enters website into the browser
- Browser sends a request to the server
- Server sends the information to the browser
- Browser receives the information

How Can We  
Use this?

# How Can We Use This?

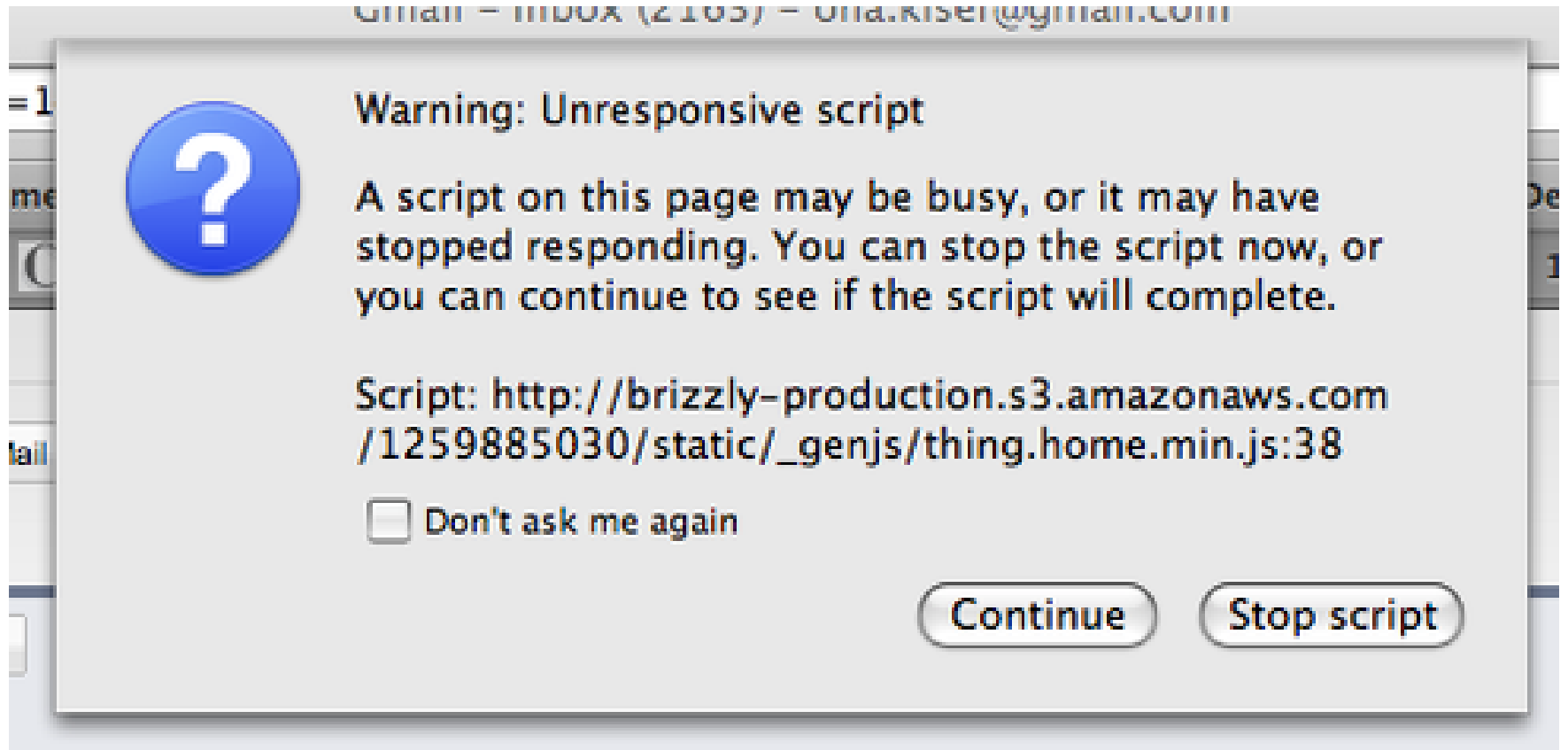
- The client visits a web page
- Server sends the client instructions
- The client side processes instructions with JavaScript
- Client sends back answers to the web server



# Additional Information

- Browsers are not made to do heavy computations
- In fact they are made to stop heavy computations
- Browsers become unresponsive
- JavaScript is single threaded

# Browsers Stop Heavy Computation



# Web Workers



## Web Workers

Editor's Draft 14 December 2010

**Latest Published Version:**

<http://www.w3.org/TR/workers/>

**Latest Editor's Draft:**

<http://dev.w3.org/html5/workers/>

**Previous Versions:**

<http://www.w3.org/TR/2009/WD-workers-20090423/>

<http://www.w3.org/TR/2009/WD-workers-20091029/>

**Editor:**

[Ian Hickson](#), Google, Inc.

Copyright © 2010 W3C<sup>®</sup> (MIT, ERCIM, Keio). All Rights Reserved. W3C liability, trademark and document use rules apply.  
The bulk of the text of this specification is also available in the WHATWG [Web Applications 1.0](#) specification, under a license that permits reuse of the specification text.

## Abstract

This specification defines an API that allows Web application authors to spawn background workers running scripts in parallel to their main page. This allows for thread-like operation with message-passing as the coordination mechanism.

## Status of This document

# Disadvantages of this Model

- Web Workers are new
- Not widely supported
- Internet Speed
- Potential loss of computed results
- JavaScript is Slow
- Not good for real-time problem solving
- Not dependable

# Advantages of this Model

- Multi-Platform
- Easy to Use
- No installation required just a browser
- No installation
- JavaScript is getting faster
- Applications moving to browser
- Could work on local network
- Browsers to support hardware acceleration

How Am I  
Going to  
Use This?

# Bin Packing

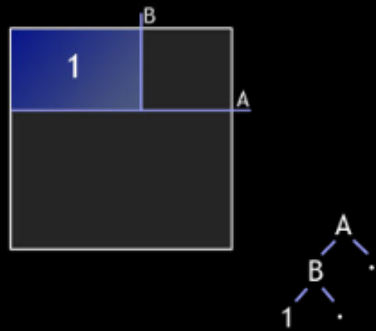


# Packing Algorithm

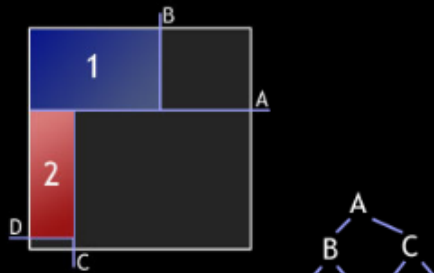
## Packing Lightmaps

Pop Quiz: You have 765,618 lightmaps for a scene and very few of them have power of 2 dimensions, what do you do? If your answer was to rescale them and call `CreateTexture` 765,618 times please slap yourself. If your answer had anything to do with `glTexImage2D`, you might want to leave now. What you really want to do is smush them all into a couple larger textures, and this text will show you one way of doing it.

What we'll do is recursively divide the larger texture into empty and filled regions. We start off with an empty texture and after inserting one lightmap we get this:



Here we've split the texture in half by line A then split the upper half by B and inserted the first lightmap to the left of B. When we go to insert the next one, we'll first check if it can fit above A and if it can we check to see if it can fit left of B, nope full, then right of B. If it fits there we split B exactly how we split the original texture, otherwise we insert and split below A. After inserting a second lightmap the texture becomes:





# Bin Packing

- Big bag of 3D boxes
- Client will only see 2 dimensions
- Client will get a bunch of boxes
- Leftovers get sent back
- Results sent back to server
- Server will see all 3 Dimensions
- Server will optimize for Height (3<sup>rd</sup> dimension)

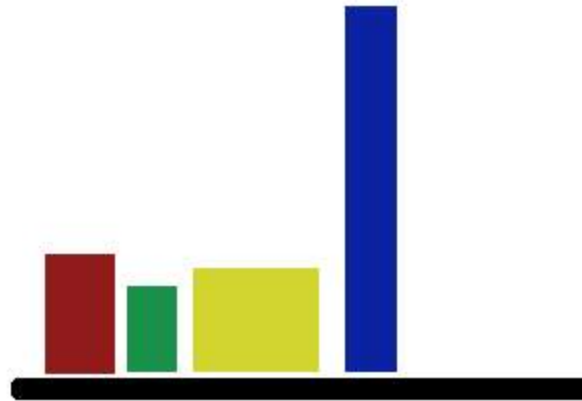
# Data I am Using

- Unit: pixels
- Client level size: 640 x 480
- Box dimensions range from 1 to 8
- Boxes are randomly generated
- Probability is the same for all sizes  
(Example 8 is as likely to be picked as 1)

# Technologies Used

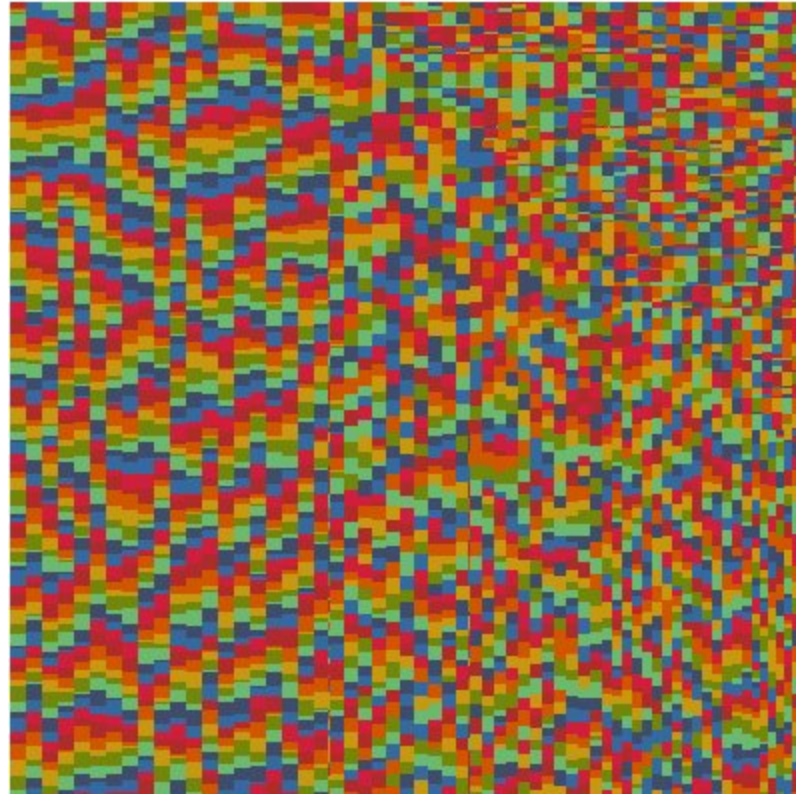
- Server Side: PHP, MySQL (shared hosting)
- Client Side: JavaScript (Chrome, Firefox)
- Server Side: Why PHP and MySQL?
- Client Side Why not Java or Flash?

# Problems: Height of Boxes Range



Pictured above is one level (what a client gets) from the side.  
We see the 3<sup>rd</sup> dimension. Clients only sees 2 Dimensions.

# Problems: Client Is Only Looking at 2-Dimensions

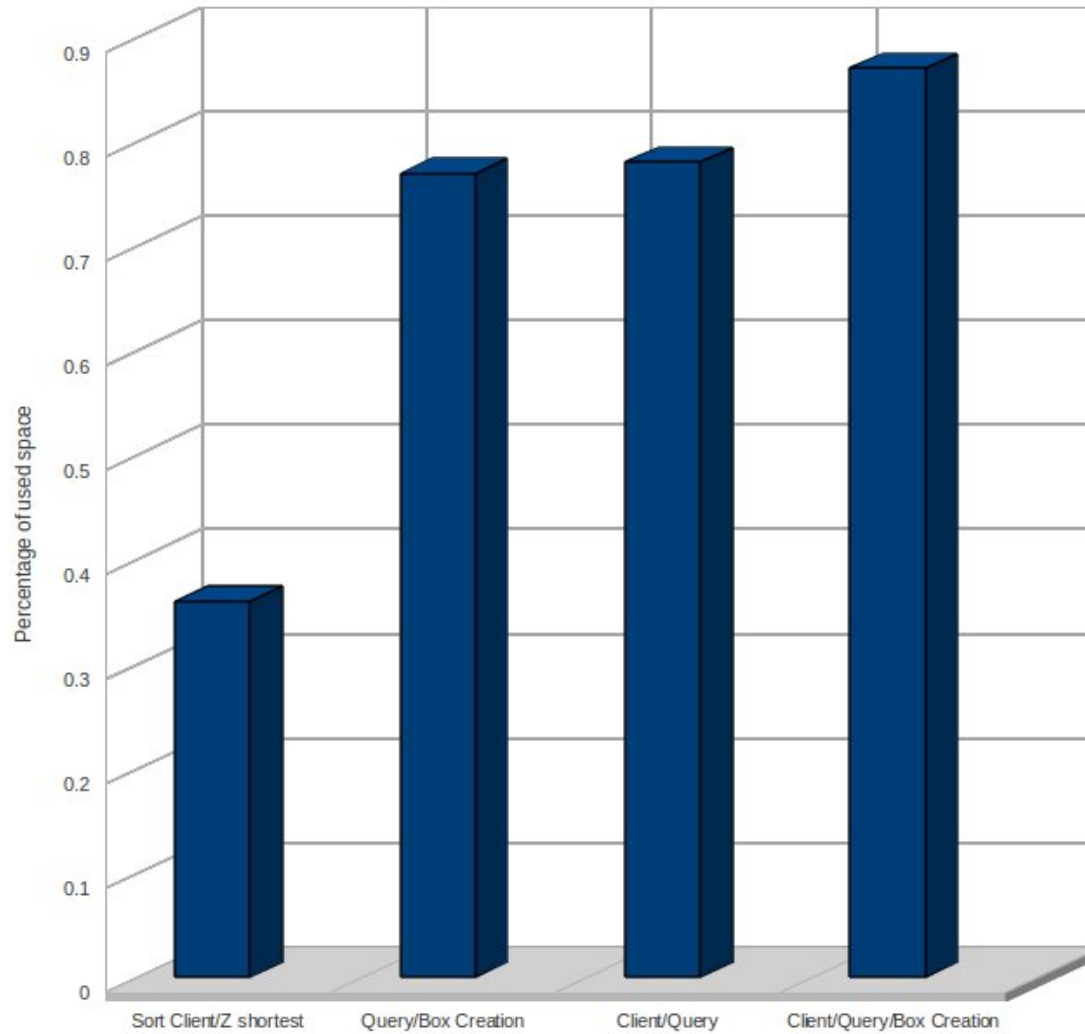


Pictured above is one level (what a client gets). Here we see a tightly packed level that the client has packed.

# Optimizing For 3 Dimensions

- Sort boxes so that the 3<sup>rd</sup> dimension is the smallest side (height)
- Sort the boxes in the database query
- Sort by the x value on the client side

# Results



# Scheduling

- Client receives a unique identifier
- The client checks out 16,000 boxes
- Boxes that have not been checked back in 3 minutes get put back into the pool
- If the client refreshed their page within the 3 minute window, they get the same set of boxes
- Each box has its own row in the database
- Results set is written back to the database
- Boxes are now flagged as being used and can never be checked out again



# Thank You For Your Time

## Contact Info:

[johnlonganecker@gmail.com](mailto:johnlonganecker@gmail.com)

<http://www.johnlonganecker.com>

# References

- Packing Light Maps Algorithm:

<http://www.blackpawn.com/texts/lightmaps/default.html>

- Bin Packing with JavaScript

<http://pollinimini.net/blog/rectangle-packing/>