Υπολογιστική υψηλών επιδόσεων με τη χρήση Web Workers

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΠΜΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΜΠΙΛΜΠΙΛΗΣ ΒΑΣΙΛΕΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΚΑΣΚΑΛΗΣ ΘΕΟΔΩΡΟΣ

Τι είναι οι Web Workers;

- Με τον όρο web worker αναφερόμαστε σε ένα script της Javascript το οποίο εκτελείται στο παρασκήνιο σε ένα νέο thread, ανεξάρτητα από το κεντρικό thread της σελίδας
- Επειδή εκτελούνται ανεξάρτητα από το κεντρικό thread που διαχειρίζεται το UI της ιστοσελίδας, όσο χρονοβόρα κι αν είναι η εργασία που θα κληθούν να εκτελέσουν δεν επηρεάζεται η αποκρισιμότητα της εφαρμογής
- Αποτελούν OS-Level threads με υψηλό κόστος σε πόρους για την δημιουργία τους, επομένως η χρήση πολλών web workers μπορεί να επιβαρύνει σημαντικά τις επιδόσεις του συστήματος στο οποίο εκτελούνται
- Χρησιμοποιούν το Messaging API της HTML5 για την επικοινωνία με το κεντρικό thread

Στόχος της Διπλωματικής

Κύριος στόχος είναι να αναδειχθούν οι δυνατότητες των Web Workers και να διερευνηθεί σε ποιες περιπτώσεις θα μπορούσαν να επωφεληθούν οι σύγχρονες συσκευές όπως tablets, smartphones και υπολογιστές, οι οποίες πλέον διαθέτουν multi-core επεξεργαστές.

Πιο συγκεκριμένα, οι τομείς οι οποίοι διερευνήθηκαν στο πλαίσιο της διπλωματικής είναι οι εξής:

- Ποιες εφαρμογές θα μπορούσαν να επωφεληθούν περισσότερο
- Διαφορές επιδόσεων μεταξύ των γνωστών browsers αλλά και μεταξύ εκδόσεων του ιδίου browser
- Ποιες συσκευές που θα μπορούσαν να επωφεληθούν περισσότερο από την χρήση Web Workers
- Ποιος είναι ο βέλτιστος αριθμός threads ώστε η απόδοση να είναι η καλύτερη δυνατή
- Διαφορές στις επιδόσεις εφαρμογών με την χρήση των Web Workers ώστε να δούμε σε τι ποσοστό έχουμε βελτίωση στην απόδοση

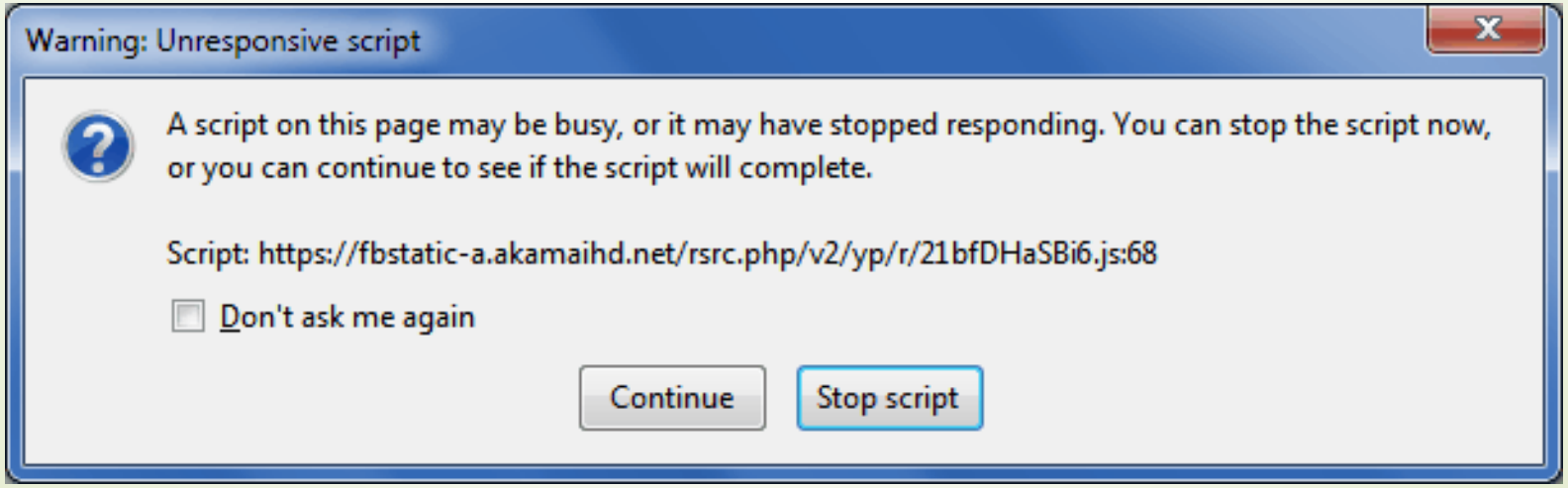
Περιγραφή του προβλήματος (1/2)

- Όλο και περισσότερες εφαρμογές εκτελούνται μέσω ενός browser, με τις εφαρμογές αυτές να εξελίσσονται γρήγορα και να γίνονται πιο απαιτητικές σε υπολογιστικούς πόρους
- Παράλληλα, οι περισσότερες συσκευές που χρησιμοποιούμε καθημερινά για την πρόσβαση μας σε web εφαρμογές διαθέτουν πλέον επεξεργαστές με πολλούς πυρήνες
- Αξιοποιούμε όμως πραγματικά όλη την επεξεργαστική ισχύ που αυτές οι συσκευές μπορούν να μας παρέχουν;
- Δεδομένου ότι η Javascript είναι βασισμένη σε ένα single-thread μοντέλο, δεν μπορούμε να επωφεληθούμε από την multi-core αρχιτεκτονική των σύγχρονων επεξεργαστών

Περιγραφή του προβλήματος (2/2)

Το πρόβλημα που μπορεί να αντιμετωπίσουμε λόγω του μοντέλου εκτέλεσης της Javascript είναι αυτό που εμφανίζεται στην παρακάτω εικόνα.

Η εκτέλεση ενός μεγάλου script μπορεί να «παγώσει» τον browser και να αναγκάσει τον χρήστη είτε να κλείσει τον browser είτε να περιμένει πότε αυτός θα «ξεπαγώσει»



Προτεινόμενη λύση και οφέλη

Την λύση στο πρόβλημα αυτό έρχονται να δώσουν οι Web Workers, οι οποίοι μπορούν να προσδώσουν τα παρακάτω οφέλη:

- Δυνατότητα εκτέλεσης απαιτητικών web εφαρμογών χωρίς να χάνεται η αποκρισιμότητα του UI
- Αξιοποίηση των πολλών πυρήνων που διαθέτουν οι σύγχρονοι επεξεργαστές για καλύτερες επιδόσεις
- Εκτέλεση είτε σε client-side είτε server-side επίπεδο, επομένως μεγάλο μέρος των υπαρχόντων εφαρμογών μπορούν να επωφεληθούν χωρίς σημαντικές αλλαγές στον κώδικα

Πιθανές χρήσεις των Web Workers

- Πολύπλοκοι μαθηματικοί υπολογισμοί
- Ταξινόμηση ή επεξεργασία μεγάλου μεγέθους πινάκων
- Επεξεργασία μεγάλων JSON Objects (parsing)
- Επισήμανση του συντακτικού ενός κώδικα ή κάποιου άλλου τύπου ανάλυση κειμένου σε πραγματικό χρόνο
- Ανάλυση ή επεξεργασία δεδομένων video, εικόνας και ήχου
- Χειρισμός αιτημάτων μεταξύ client και server (e.g. Background I/O)
- Συντρέχοντα αιτήματα προς μια βάση δεδομένων

Περιορισμοί και Δυνατότητες (1/2)

Λόγω της multi-threaded φύσης τους, οι web workers υπόκειται σε κάποιους περιορισμούς οι οποίοι περιγράφονται παρακάτω:

- Δεν έχουν πρόσβαση Document Object Model (DOM) και επομένως δεν έχουν πρόσβαση στα window, document και parent objects
- Δεν έχουν πρόσβαση σε βιβλιοθήκες Javascript οι οποίες χρησιμοποιούν τα παραπάνω objects, όπως π.χ. η jQuery
- Δεν υπάρχει κοινόχρηστη μνήμη μεταξύ main thread και web worker προκειμένου να μοιράζονται δεδομένα

Περιορισμοί και Δυνατότητες (2/2)

Παρ’ όλο που οι Web Workers έχουν πρόσβαση σε ένα περιορισμένο κομμάτι των δυνατοτήτων της Javascript, υπάρχουν ακόμη αρκετές χρήσιμες δυνατότητες τις οποίες μπορούν να αξιοποιήσουν, όπως είναι οι παρακάτω:

- Πρόσβαση στα navigator object και location object (read-only)
- Δυνατότητα χρήσης της application cache
- Δυνατότητα χρήσης του XMLHttpRequest object για την αποστολή και λήψη αντικειμένων με χρήση της AJAX
- Δυνατότητα εισαγωγής άλλων script για χρήση μέσα σε ένα web worker με την μέθοδο importScripts()

Web Worker Types

Dedicated Web Worker

- Ο πιο συνηθισμένος τύπος Web Worker
- Επικοινωνεί μόνο με το κεντρικό thread
- Δημιουργείται είτε σε ξεχωριστό αρχείο javascript που περιέχει τον κώδικα του, είτε μέσα στο script του κεντρικού thread (Inline Worker)

Shared Web Worker

- Είναι ακριβώς τις ίδιες δυνατότητες με έναν dedicated web worker, επιπλέον όμως μπορεί να επικοινωνεί και με πολλαπλά παράθυρα ή καρτέλες της ίδιας εφαρμογής
- Κάθε shared worker έχει το δικό του port ώστε να επικοινωνεί μέσω αυτού με την κεντρική σελίδα ή και άλλους shared workers
- Χρησιμοποιούνται συνήθως για τον συγχρονισμό δεδομένων μεταξύ πολλαπλών σελίδων ή καρτελών της ίδιας εφαρμογής

Web Worker Code Example (1/2)

```
// Instantiating the Worker
var myHelloWorker = new Worker('helloworkers.js');
// Getting ready to handle the message sent back
// by the worker
myHelloWorker.addEventListener("message", function (event) {
    document.getElementById("output").textContent = event.data;
}, false);

// Starting the worker by sending a first message
myHelloWorker.postMessage("David");

// Stopping the worker via the terminate() command
myHelloWorker.terminate();
```

HelloWorkers.html

Web Worker Code Example (2/2)

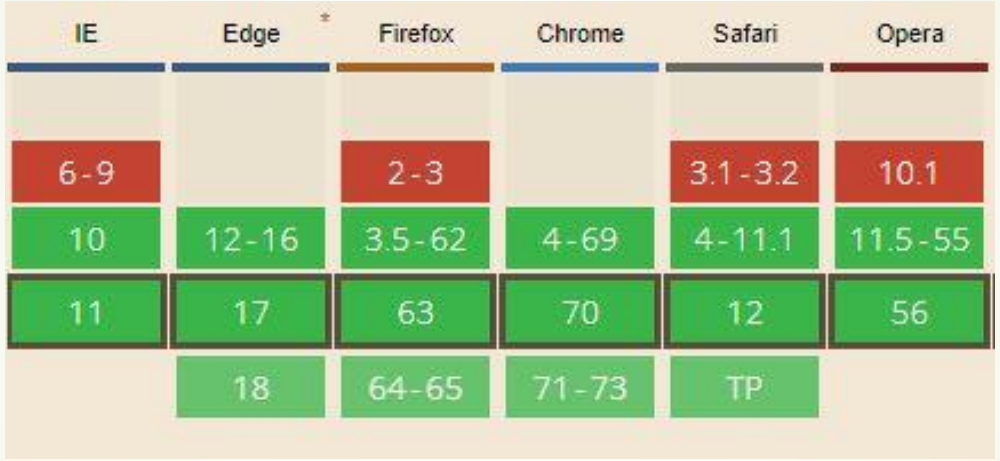
```
function messageHandler(event) {
    // Accessing to the message data sent by the main page
    var messageSent = event.data;
    // Preparing the message that we will send back
    var messageReturned = "Hello " + messageSent + " from a separate thread!";
    // Posting back the message to the main page
    this.postMessage(messageReturned);
}

// Defining the callback function raised when the main page will call us
this.addEventListener('message', messageHandler, false);
```

Helloworkers.js

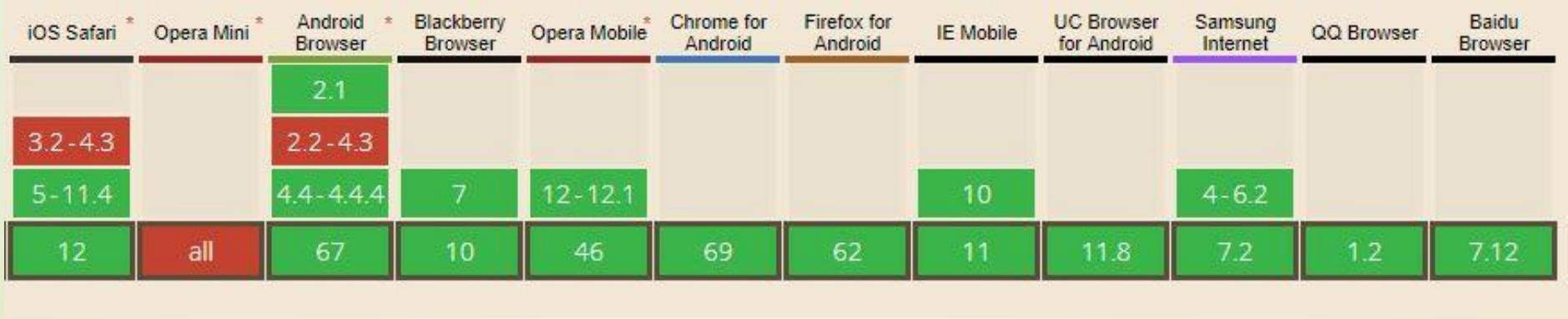
Web Workers Browser Support

Desktop Browser Support



IE	Edge *	Firefox	Chrome	Safari	Opera
6-9		2-3		3.1-3.2	10.1
10	12-16	3.5-62	4-69	4-11.1	11.5-55
11	17	63	70	12	56
	18	64-65	71-73	TP	

Mobile Browser Support



iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser
		2.1									
3.2-4.3		2.2-4.3									
5-11.4		4.4-4.4.4	7	12-12.1			10		4-6.2		
12	all	67	10	46	69	62	11	11.8	7.2	1.2	7.12

Shared Workers Browser Support

Desktop Browser Support

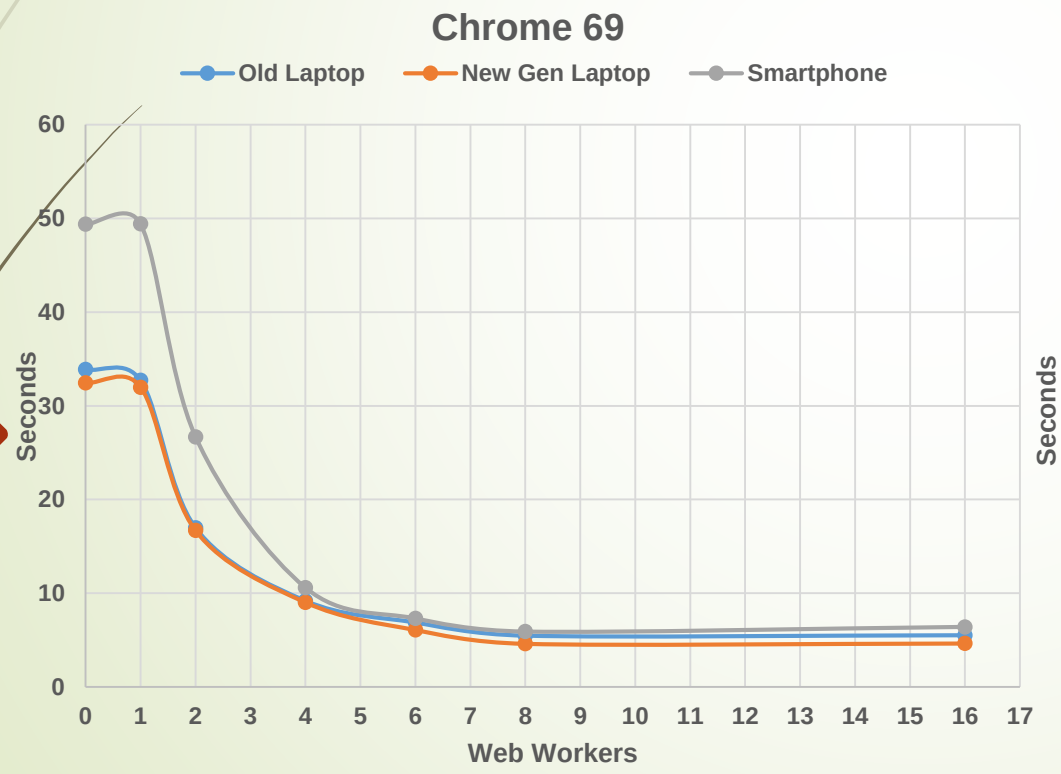
IE	Edge *	Firefox	Chrome	Safari	Opera
				3.1-4	
		2-28		5-6	10.1
6-10	12-16	29-62	4-69	6.1-11.1	11.5-55
11	17	63	70	12	56
	18	64-65	71-73	TP	

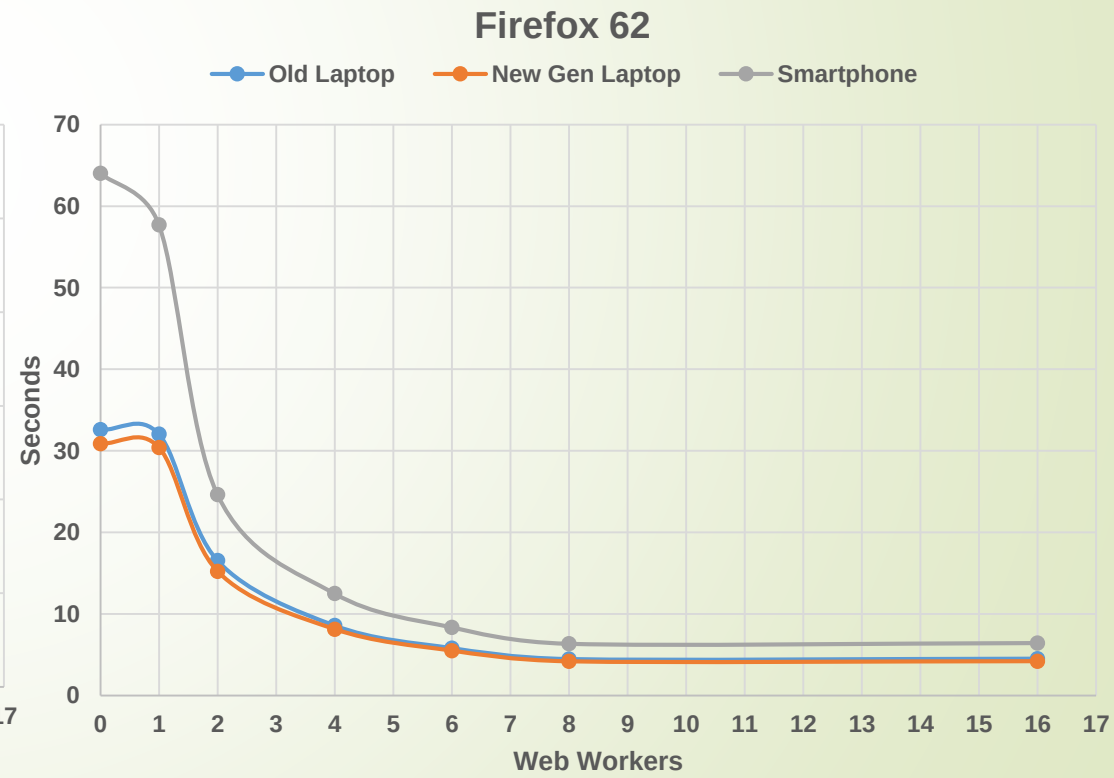
Mobile Browser Support

iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser
3.2-4.3											
5-6.1									4		
7-11.4		2.1-4.4.4	7	12-12.1			10		5-6.2		
12	all	67	10	46	69	62	11	11.8	7.2	1.2	7.12

Χρήση των Web Workers για Πολύπλοκους Μαθηματικούς Υπολογισμούς

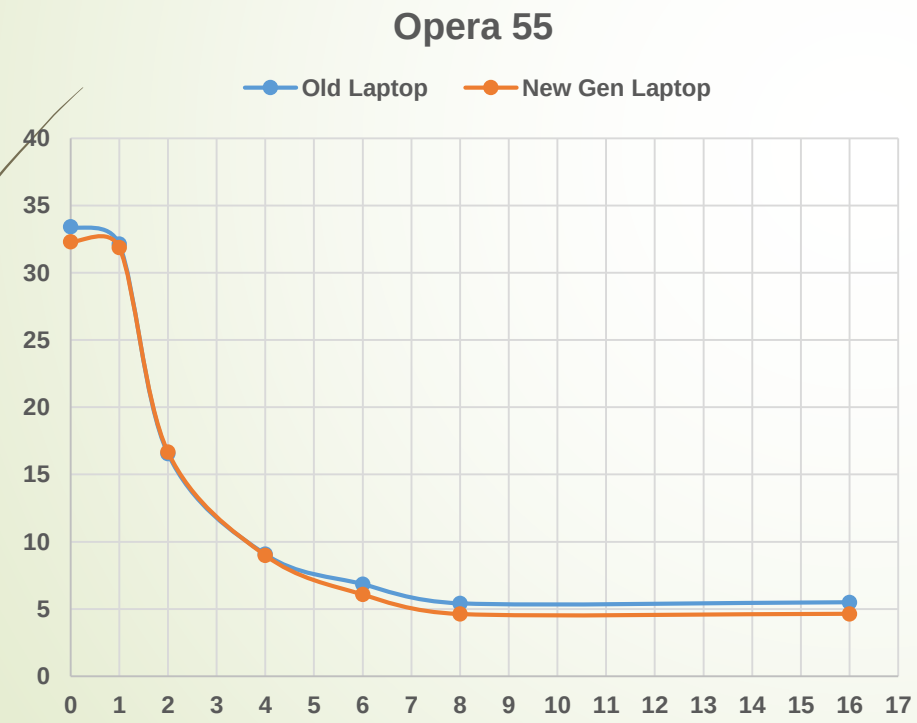
Calculation of 2^{4096M} mod 97777

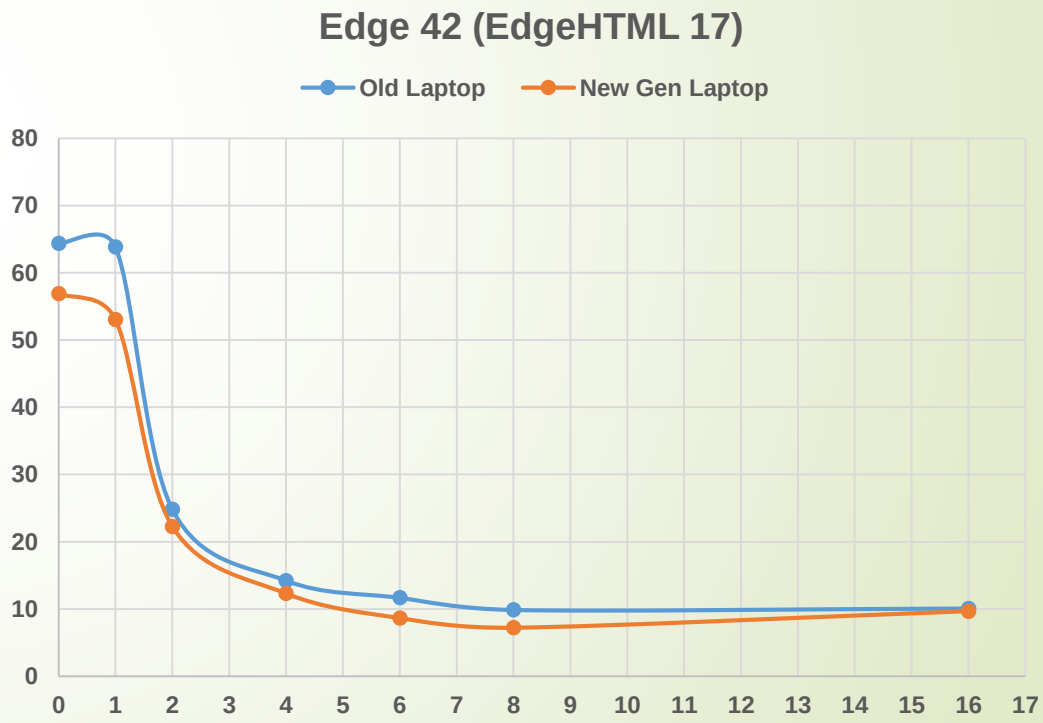




Χρήση των Web Workers για Πολύπλοκους Μαθηματικούς Υπολογισμούς

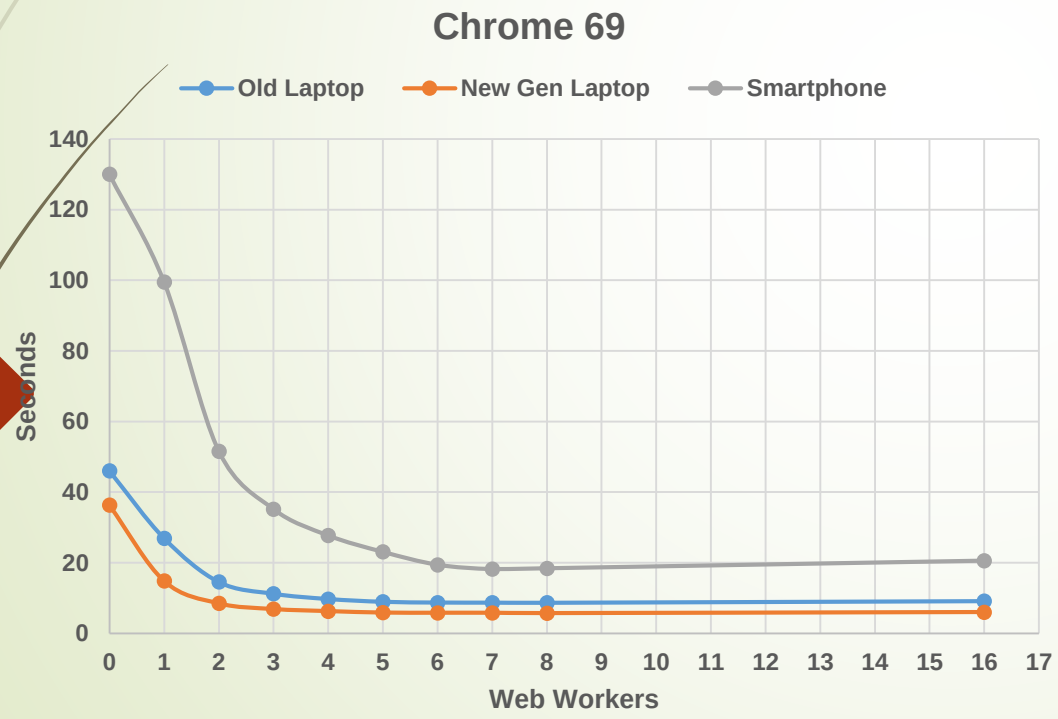
Calculation of $2^{4096M} \bmod 97777$

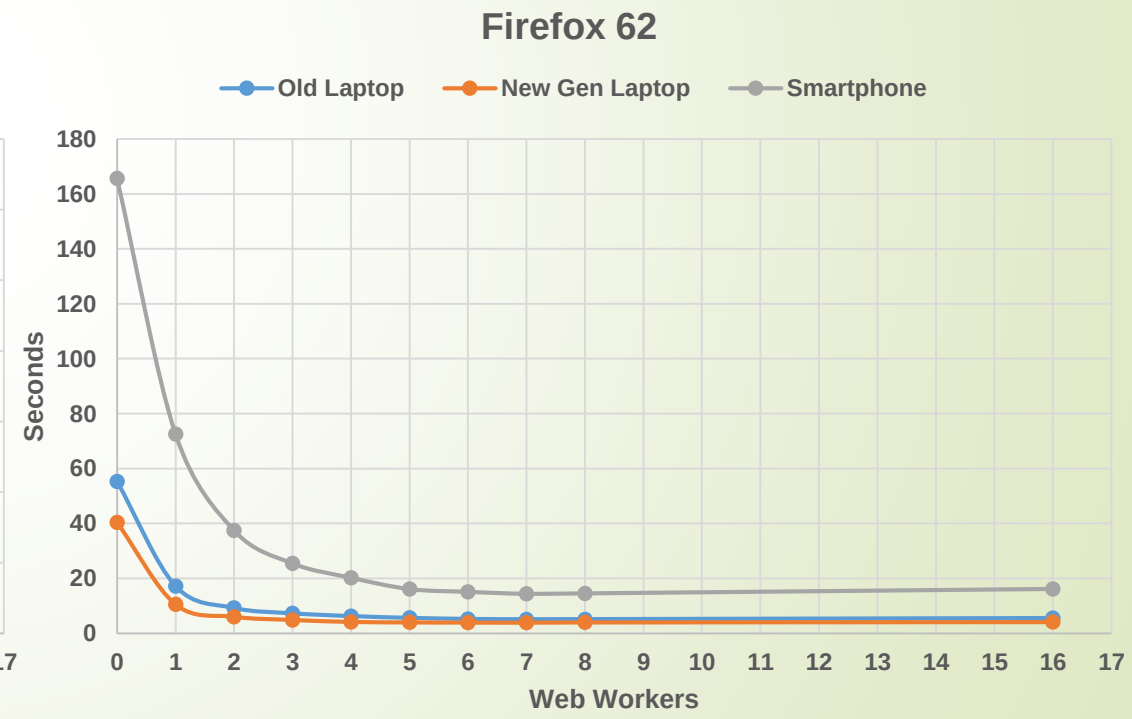




Αξιοποίηση των Web Workers για Σχεδίαση σε HTML5 Canvas Element

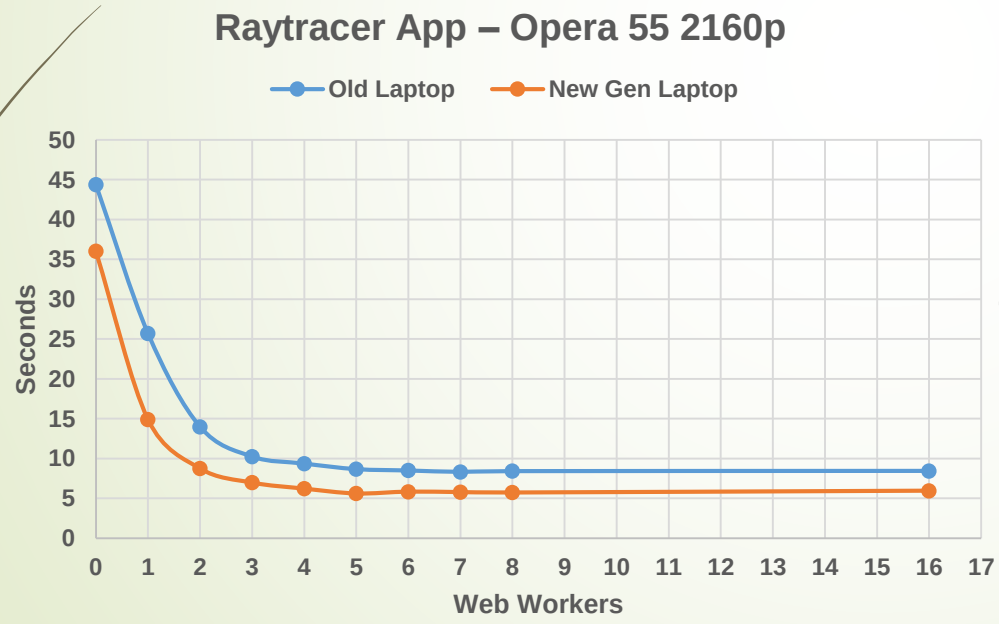
Σχεδίαση εικόνας με ανάλυση 2160p

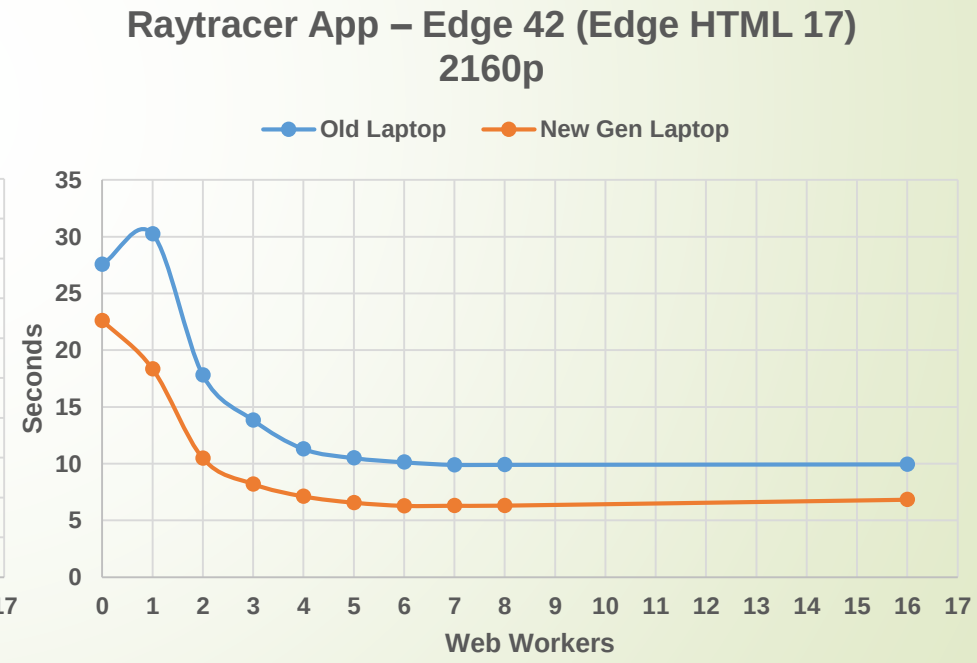




Αξιοποίηση των Web Workers για Σχεδίαση σε HTML5 Canvas Element

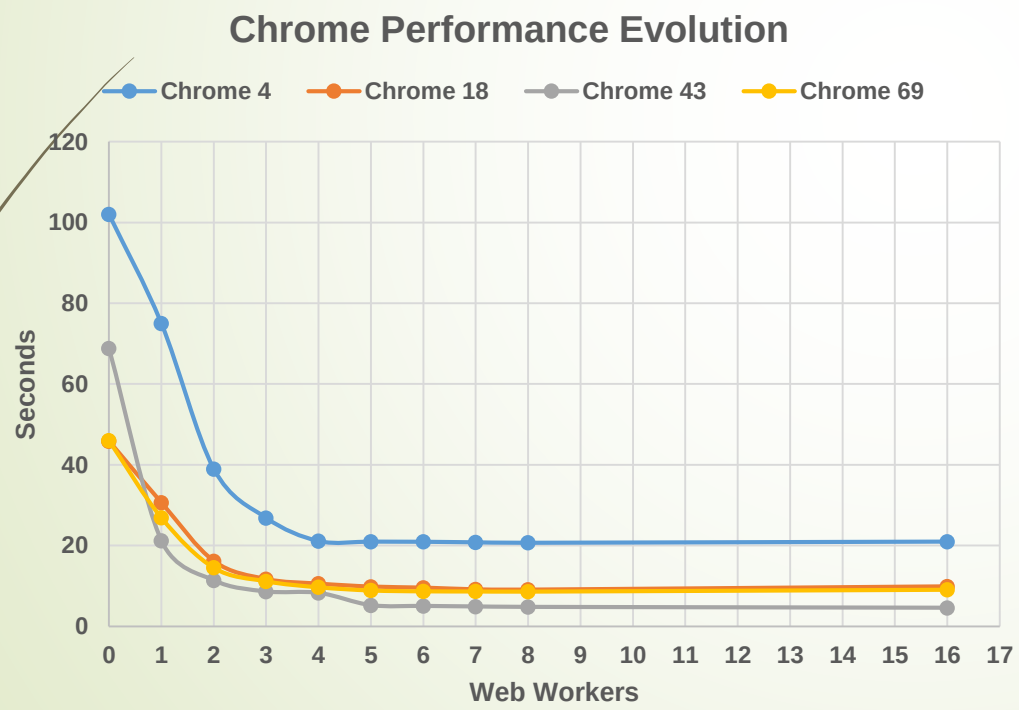
Σχεδίαση εικόνας με ανάλυση 2160p





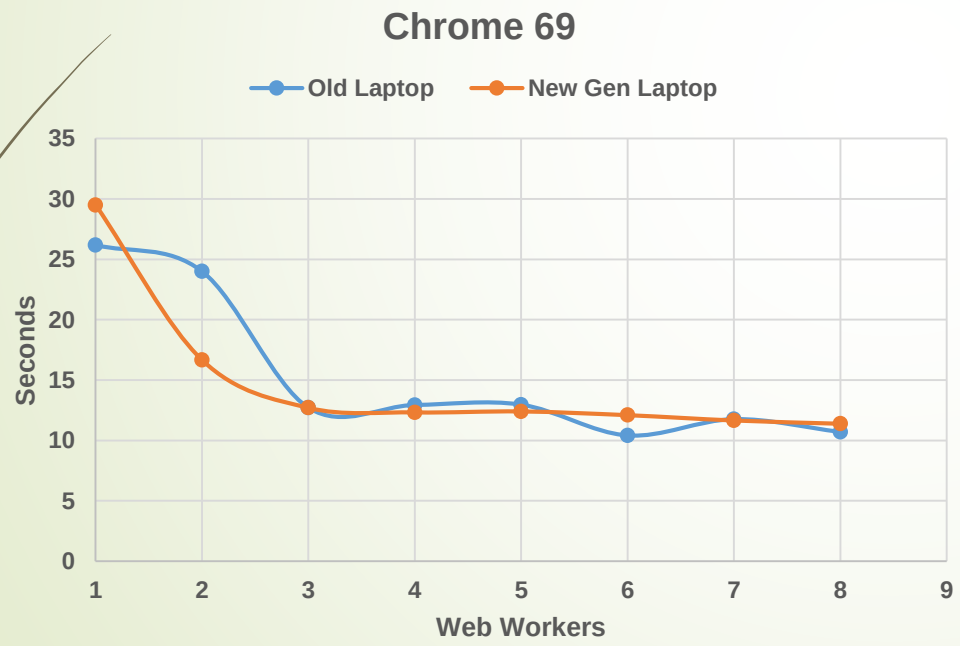
Αξιοποίηση των Web Workers για Σχεδίαση σε HTML5 Canvas Element

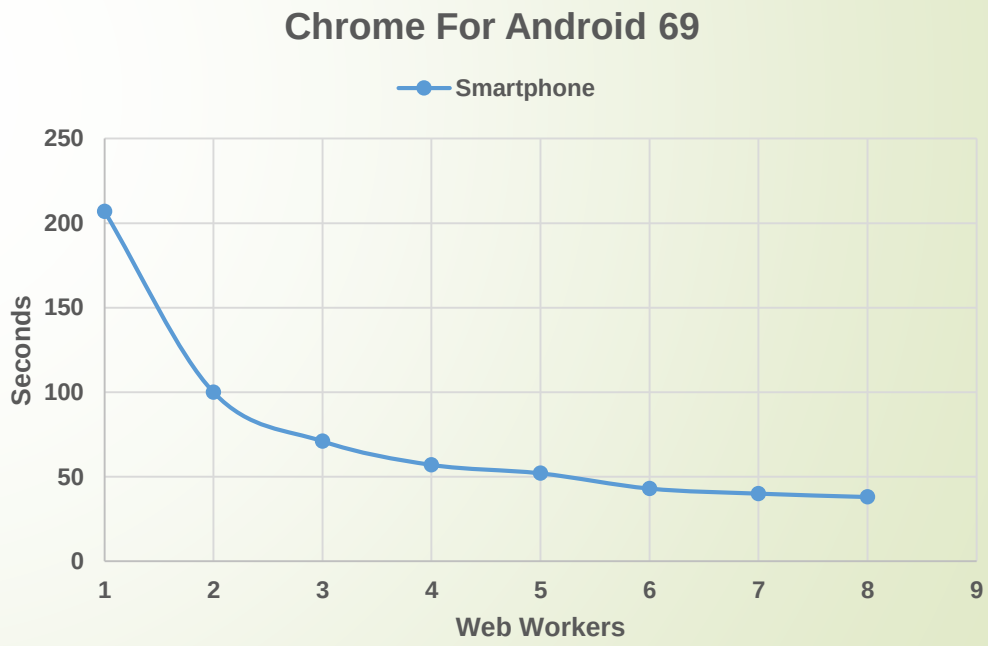
Σχεδίαση εικόνας με ανάλυση 2160p



Χρήση των Web Workers για τον Υπολογισμό MD5 Hash

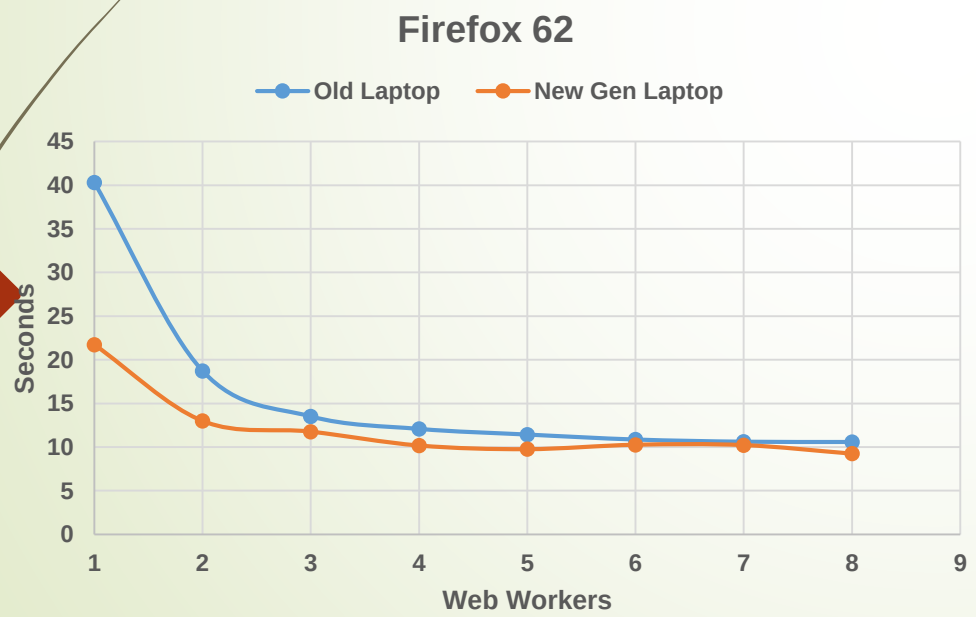
MD5 Brute Force Cracking to calculate passphrase “heya”

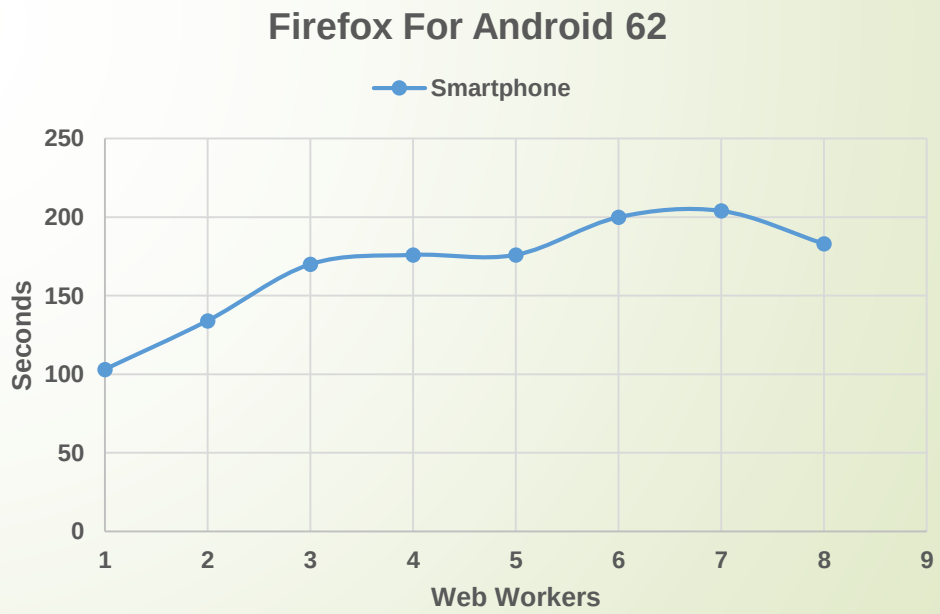




Χρήση των Web Workers για τον Υπολογισμό MD5 Hash

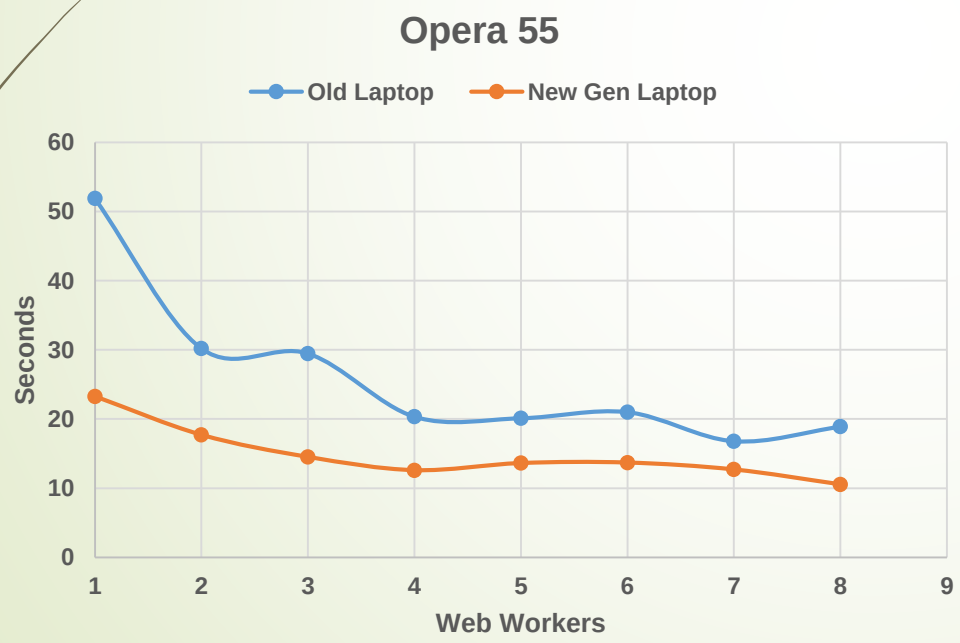
MD5 Brute Force Cracking to calculate passphrase “heya”

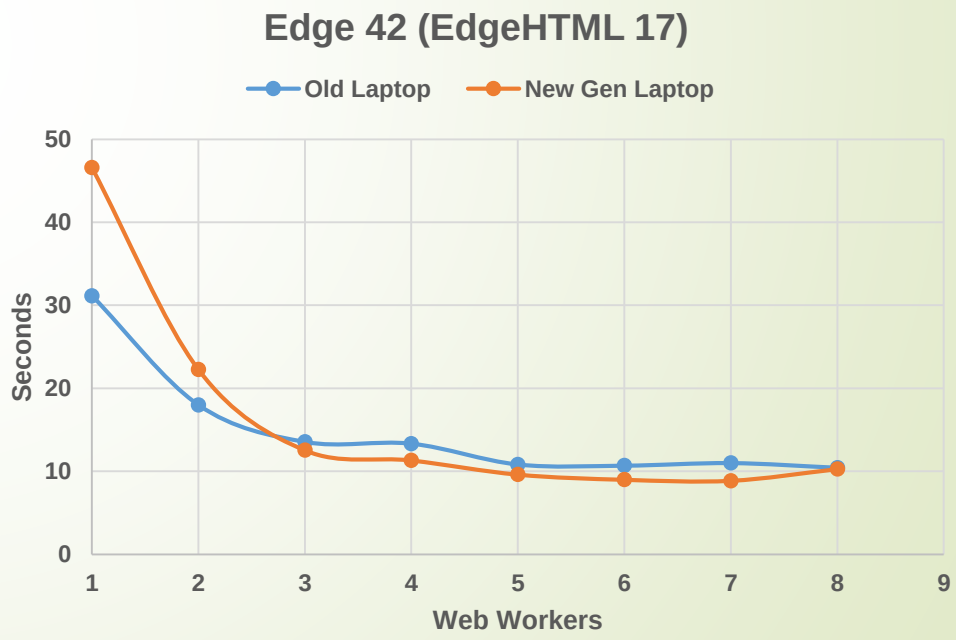




Χρήση των Web Workers για τον Υπολογισμό MD5 Hash

MD5 Brute Force Cracking to calculate passphrase “heya”





Συμπεράσματα

- Είναι ξεκάθαρο πως οι Web Workers μπορούν να συνεισφέρουν σημαντικά στην υλοποίηση web εφαρμογών με σημαντικό υπολογιστικό φόρτο, καθιστώντας εφικτή την χρήση τέτοιων εφαρμογών σε επίπεδο client-side
- Η χρήση των Web Workers ακόμη σε συσκευές με χαμηλότερη επεξεργαστική δύναμη όπως ένα smartphone, μπορούν να βελτιώσουν σε πολύ μεγάλο βαθμό τις επιδόσεις δεδομένου ότι πλέον χρησιμοποιούν επεξεργαστές με πολλούς πυρήνες
- Ο αριθμός των Web Workers που θα πρέπει να χρησιμοποιήσει μια εφαρμογή ώστε να επιτύχει την καλύτερη απόδοση εξαρτάται από διάφορους παράγοντες και δεν είναι πάντα εύκολο να υπολογισθεί
- Η χρήση αριθμού web workers ίσου με τους πραγματικούς πυρήνες ενός συστήματος συνήθως αποδίδει ένα αποτέλεσμα πολύ κοντά ή και ίσο με την βέλτιστη απόδοση
- Ο φυλλομετρητής που θα χρησιμοποιηθεί για την εκτέλεση μιας web εφαρμογής είδαμε ότι επηρεάζει την απόδοση αλλά επίσης σημαντικό ρόλο παίζει ο τύπος της εφαρμογής, οπότε δεν είναι εύκολο να ξεχωρίσουμε κάποιον browser ως προς την απόδοση του

Ευχαριστώ για την προσοχή σας!

Ερωτήσεις;