1. Εισαγωγή

1.1. Ρυθμός δημιουργίας δεδομένων

Τα τελευταία χρόνια παρατηρείται ραγδαία αύξηση του ρυθμού παραγωγής δεδομένων. Πόσο μεγάλη είναι αυτή η αύξηση; Έρευνα της IDC για λογαριασμό της DELL EMC αναφέρει πως το 2020 αναμένεται ως ανθρωπότητα να έχουμε συσσωρεύσει 40 zettabytes δεδομένων. Σύμφωνα με την ίδια έρευνα, το 2010 είχαμε συσσωρεύσει μόλις 1.2 zettabytes. Αυτός ο ρυθμός αύξησης των δεδομένων, κατά 2Χ ανά δυο έτη, τροφοδοτείται από μια σειρά από τεχνολογίες και κοινωνικά φαινόμενα. Κάποιες από τις κυριότερες πηγές αύξησης του όγκου των δεδομένων είναι το user generated content, το οποίο οδηγείται κατά κύριο λόγο από τη δυνατότητα εγγραφής εικόνας και βίντεο των σύγχρονων smartphones και τη δυνατότητα άμεσης αποστολής του περιεχομένου στα κοινωνικά δίκτυα, όπως για παράδειγμα το Facebook, το Instagram, το Twitter και το ΥouΤube. Το 2019 κάθε λεπτό προβάλλονται 4.5 εκατομμύρια βίντεο στο YouTube, ενώ 1 εκατομμύρια χρήστες κάνουν login στο Facebook. Η δυνατότητα άμεσου σχολιασμού αυτών των δεδομένων στις πλατφόρμες αυτές δημιουργεί με τη σειρά της νέα δεδομένα. Οι ανάγκες επικοινωνίας των ανθρώπων με κάθε μορφή, όπως tweets, γραπτά μηνύματα, μήνυμα email, ηχητική ή κλήση βίντεο, σύντομα βίντεο ή gifs αποτελούν σημαντικό κομμάτι της δημιουργίας νέων δεδομένων σε καθημερινή βάση. Στην ετήσια έκθεση της DOMO ‘Data Never Sleeps 7.0’ φαίνεται ότι κάθε λεπτό ανεβαίνουν στο Twitter 511.200 tweets, αποστέλλονται 188.000.000 emails, ενώ πραγματοποιούνται 231.840 κλήσεις μέσω Skype. Είναι σημαντικό επίσης να αναφέρουμε ότι κάθε συναλλαγή που κάνουμε με υπηρεσίες που δεν έχουν να κάνουν άμεσα με ανταλλαγή δεδομένων, όπως πλατφόρμες ενοικίασης κατοικιών (Airbnb), υπηρεσίες ταξί (Uber, Beat) δημιουργούμε νέα δεδομένα τόσο όταν ανεβάζουμε μια αγγελία όσο και με την κάθε επαφή μας με την υπηρεσία όπως πχ οι διαδρομές στο ταξί.

Επιπλέον, με την αύξηση του bandwidth που είναι διαθέσιμο στους χρήστες, χάρη στα δίκτυα νέας γενιάς (5G, VDSL, Fiber to The Home) ο ρυθμός παραγωγής νέων δεδομένων αναμένεται να συνεχίσει να αυξάνεται με γοργούς ρυθμούς.

Οι streaming πλατφόρμες όπως το Netflix και το Spotify όχι μόνο δημιουργούν μεγάλη κίνηση στο δίκτυο μέσω της αποστολής video και μουσικής αντίστοιχα, αλλά δημιουργούν με τη σειρά τους νέα δεδομένα όπως στατιστικά χρήσης της υπηρεσίας ανά χρήστη, χώρα, είδος ταινίας ή μουσικής κλπ. Ένας άλλος παράγοντας που συμβάλλει στην αύξηση του ρυθμού παραγωγής δεδομένων είναι οι κάθε λογής έξυπνες συσκευές. Η παραγωγή οικονομικών αισθητήρων κάθε είδους (θερμοκρασίας, φωτεινότητας, ατμοσφαιρικής πίεσης, καρδιακών παλμών κλπ.) σε συνδυασμό με τους υπολογιστές ολοκληρωμένους σε μια ψηφίδα (System on Chip ή αλλιώς SoC) οι οποίοι ενσωματώνουν σε ένα ολοκληρωμένο Chip όλες τις βασικές λειτουργίες ενός υπολογιστικού συστήματος (CPU, RAM, WiFi, Bluetooth κλπ.) έχει δημιουργήσει μια σειρά από έξυπνες συσκευές, όπως smartwatches, έξυπνες λάμπες, θερμοστάτες οι οποίοι λειτουργούν με εξαιρετικά χαμηλή κατανάλωση ενέργειας και παίρνουν διαρκώς μετρήσεις τις οποίες αποστέλλουν στο διαδίκτυο.

Εκτός του ανθρώπου, πληροφορίες πλέον ανταλλάσσουν και οι ίδιες οι μηχανές μεταξύ τους. Τα τελευταία χρόνια όλο και περισσότερες συσκευές, όπως ψυγεία, τηλεοράσεις και άλλες οικιακές αλλά και βιομηχανικές συσκευές αποκτούν δυνατότητα σύνδεσης στο διαδίκτυο, συνθέτοντας αυτό που αποκαλούμε Internet of Things (IoT). Αυτό τους δίνει τη δυνατότητα της μεταξύ τους συνεργασίας καθώς και της καλύτερης επαφής με τον άνθρωπο.

Η παραγωγή τόσο μεγάλου όγκου δεδομένων θέτει νέες προκλήσεις στην αποτελεσματική αποθήκευση και επεξεργασία τους. Η ανάγκη αποθήκευσης των δεδομένων είναι κάτι κατανοητό. Οι χρήστες περιμένουν τα δεδομένα που ανεβάζουν στο διαδίκτυο να είναι διαθέσιμα κάθε στιγμή. Επίσης τα δεδομένα καταγραφής των έξυπνων μετρητών πρέπει να είναι διαθέσιμα για αναφορά, για παράδειγμα οι τιμές της θερμοκρασίας κάθε μέρα, τα τελευταία 40 έτη σε διάφορες περιοχές του πλανήτη.

1.2. Η ανάγκη επεξεργασίας τεράστιου όγκου δεδομένων

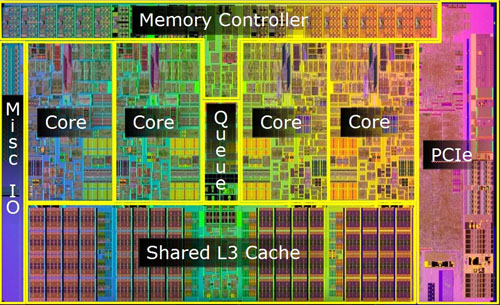
Όμως για ποιο λόγο να θέλουμε επεξεργασία των δεδομένων αυτών;   
Ανέκαθεν οι άνθρωποι μελετούσαν διάφορα φαινόμενα όπως θερμοκρασία, μετανάστευση πτηνών, ηλιοφάνεια κλπ. προκειμένου να βρουν κάποια συσχέτιση (pattern) μεταξύ των φαινομένων αυτών και αλλαγών στο περιβάλλον τους, όπως ξηρασία ή απότομες καιρικές αλλαγές. Σήμερα, ο τεράστιος όγκος των δεδομένων που παράγονται καθημερινά μπορεί να επηρεάσει δραστικά κάθε κλάδο και ανθρώπινη δραστηριότητα. Για παράδειγμα αναλύοντας ιατρικά δεδομένα μπορούμε να βρούμε συσχετίσεις μεταξύ αύξησης μιας σειράς χαρακτηριστικών, όπως συσσώρευση ουσιών στο αίμα και συγκέντρωση πρωτεϊνών, με την πιθανότητα εκδήλωσης διάφορων ασθενειών. Σύμφωνα με έρευνα της McKinsey η χρήση δεδομένων μεγάλης κλίμακας στην Ιατρική πράξη έχει τη δυνατότητα να συσχετίσει διάφορες μεταβολές στον οργανισμό και να βοηθήσει στην αποτελεσματικότερη πρόληψη. Αντίστοιχα, κοιτώντας τις θερμοκρασίες περιοχών του πλανήτη σε βάθος χρόνου μπορούμε να βρούμε αυτές που παρουσιάζουν την μεγαλύτερη αύξηση στη θερμοκρασία αλλά και να συνδέσουμε την αύξηση αυτή με πιθανά αίτια όπως αύξηση διοξειδίου του άνθρακα ή μείωση του πρασίνου στις περιοχές αυτές. Αλλά και επιχειρήσεις μπορούν να καταλήξουν σε χρήσιμα συμπεράσματα για τους πελάτες τους όπως για παράδειγμα ποιες ιστοσελίδες επισκέπτονται συχνά, τι άλλα προϊόντα αγοράζουν παράλληλα με το δικό τους, σε ποιες περιοχές ή ηλικιακές ομάδες έχουν μεγαλύτερη απήχηση κλπ. Η γνωστότερη ίσως υπηρεσία streaming ταινιών, το Netflix χρησιμοποιεί εκτενώς στατιστικά από τα interactions των χρηστών με την εφαρμογή, όπως ποιες ταινίες τους άρεσαν ή πόσο χρόνο παρακολούθησαν μια σειρά. Εφαρμόζοντας αλγορίθμους μηχανικής μάθησης πάνω στα δεδομένα των χρηστών του μπορεί να ανακαλύπτει συσχετίσεις μεταξύ των ταινιών που άρεσαν στο χρήστη και ταινιών που είναι πιθανό να του αρέσουν στο μέλλον, δημιουργώντας έτσι αποτελεσματικότερα recommender systems. Τα recommender systems αυτά υπολογίζεται πως συνεισφέρουν κάθε χρόνο 1 δις. δολάρια στα έσοδα της υπηρεσίας. Όπως γίνεται αντιληπτό, από την ανάλυση δεδομένων μεγάλης κλίμακας μπορεί να προκύψει νέα γνώση για την πρόληψη και καταπολέμηση ασθενειών, την επιβράδυνση ή και αναστολή της υπερθέρμανσης του πλανήτη καθώς και για την αύξηση της κερδοφορίας των επιχειρήσεων ή ακόμα και δημιουργία καλύτερων προϊόντων.

Αφού λοιπόν καταλάβαμε γιατί θα πρέπει να επεξεργαστούμε όλα αυτά τα δεδομένα, μένει να βρούμε τον κατάλληλο τρόπο για να το πραγματοποιήσουμε.

1.3. Ρυθμός αύξησης υπολογιστικής ισχύος

Είπαμε παραπάνω ότι ο ρυθμός παραγωγής δεδομένων αυξάνεται συνεχώς, οδηγώντας σε διπλασιασμό των διαθέσιμων δεδομένων κάθε 2 χρόνια. Αλήθεια, με τι ρυθμό αυξάνεται η υπολογιστική ισχύς; Στο διάσημο πλέον ρητό του, το 1977 ο Gordon Moore, συνιδρυτής της Intel ισχυρίστηκε πως ο αριθμός των ολοκληρωμένων κυκλωμάτων που μπορούν να ενσωματωθούν σε ένα chip θα διπλασιάζεται κάθε δυο χρόνια. Παραδόξως ο ισχυρισμός αυτός προσεγγιστικά ισχύει ακόμα και σήμερα. Ο διπλάσιος όμως αριθμός transistor δεν ισοδυναμεί σε διπλασιασμό της απόδοσης, καθώς μεγάλο μέρος των transistor δεν χρησιμοποιείται για τη μονάδα επεξεργασίας, αλλά σε άλλα βοηθητικά υποσυστήματα όπως κρυφές μνήμες, ελεγκτές μνήμης (integrated memory controllers ή αλλιώς IMC) και για τη διασύνδεση των υποσυστημάτων αυτών μεταξύ τους.

Αυτό φαίνεται και στην παρακάτω εικόνα όπου παρουσιάζεται το die ενός intel 4-core επεξεργαστή:



Επιπλέον υπάρχει ένα όριο στο πόσους υπολογιστικούς πυρήνες μπορούμε να προσθέσουμε σε ένα chip, καθώς μετά θα πρέπει να αυξήσουμε αντίστοιχα τις κρυφές μνήμες και τους memory controllers προκειμένου να μην έχουμε starvation λόγω έλλειψης bandwidth. Εκτός από αυτό, υπάρχουν θερμικά όρια σχετικά με το πόση ποσότητα θερμότητας μπορεί να απαχθεί από μια τόσο μικρή επιφάνεια. Χαρακτηριστικό είναι ότι ένας κορυφαίος server CPU το 2010 είχε 8 πυρήνες, ενώ το 2019 ένας αντίστοιχος έχει 64 πυρήνες. Αυτή η αύξηση του αριθμού των υπολογιστικών πυρήνων κατά 8X είναι πράγματι εντυπωσιακή. Αν όμως αναλογιστούμε ότι το 2010 είχαμε 1.2 Zettabytes δεδομένων, ενώ το 2019 έχουμε αντίστοιχα 40 Zettabytes, μια άνω του 32Χ αύξηση μεγέθους, η αύξηση αυτή φαίνεται αισθητά μικρότερη. Εκτός από αυτό, οι επεξεργαστές αυτοί είναι μια τάξη μεγέθους πιο ακριβοί από τους απλούς επεξεργαστές που χρησιμοποιούν οι οικιακοί υπολογιστές.

1.4. Κατανεμημένα συστήματα

Οι μεγάλες εταιρίες του διαδικτύου όπως η Google και η Yahoo! ήταν οι πρώτοι που αντιμετώπισαν τέτοιο όγκο δεδομένων, ήδη από τις αρχές της χιλιετίας. Βλέποντας λοιπόν ότι υπάρχει όριο στην υπολογιστική ισχύ που μπορεί να χωρέσει σε ένα σύστημα (vertical scaling) στράφηκαν στη λύση της χρήσης πολλών απλών υπολογιστών, σαν αυτούς που υπάρχουν σπίτια μας, οι οποίοι συνδεδεμένοι μεταξύ τους με δίκτυο Ethernet λειτουργούν σαν ένα σύστημα αυξημένης υπολογιστικής ισχύος. Ο κάθε υπολογιστής διαθέτει το δικό του λειτουργικό σύστημα, ενώ ένα λογισμικό, το οποίο ονομάζεται middleware, αναλαμβάνει να διαχειριστεί τους διαθέσιμους πόρους και να τους παρουσιάσει στον χρήστη σαν ένα ενιαίο σύστημα. Το ενδιάμεσο αυτό λογισμικό είναι η βάση αυτού που αποκαλούμε κατανεμημένο σύστημα (distributed system).

Χαρακτηριστικό σύστημα είναι το GFS της Google, το οποίο διαχειρίζεται τον αποθηκευτικό χώρο σε κάθε υπολογιστή και τον κάνει να εμφανίζεται σαν ενιαίος. Παράλληλα ασχολείται με θέματα που αφορούν το failover, δηλαδή την αστοχία κόμβων μέσω του replication ενός block δεδομένων σε πολλαπλούς κόμβους. Πάνω στο σύστημα αυτό, σχεδιάστηκε το σύστημα κατανεμημένης επεξεργασίας της Google, το Map Reduce. Το σύστημα αυτό αναλαμβάνει να δρομολογήσει τις εργασίες στους κόμβους του συστήματος, με τρόπο αποδοτικό σύμφωνα με τα δεδομένα που έχει ο κάθε κόμβος, την επικοινωνία των κόμβων και την ανάκαμψη από αστοχία υλικού (failover).

Η μεγαλύτερη όμως επανάσταση που έφερε το Map Reduce αφορά το προγραμματιστικό μοντέλο, δηλαδή τον τρόπο που οι προγραμματιστές γράφουν κώδικα για την επίλυση ενός προβλήματος. Ενώ ο συμβατικός τρόπος είναι οι προγραμματιστές να δημιουργούν explicitly threads στα οποία αναθέτουν ένα κομμάτι του προβλήματος και στη συνέχεια είναι υπεύθυνοι για το συγχρονισμό τους, στο Map Reduce η είσοδος ενός προβλήματος είναι μια σειρά από γραμμές. Κάθε γραμμή μπορεί να είναι ένας αριθμός, μια σειρά από αριθμούς ή ένα κείμενο. Στο μοντέλο αυτό υπάρχουν μόνο δυο συναρτήσεις, η Map και η Reduce. Η Map παίρνει μια γραμμή ως key-value pair, για παράδειγμα (αριθμός γραμμής, περιεχόμενα γραμμής) και με τη σειρά της αφού κάνει κάποιους μετασχηματισμούς στα δεδομένα αυτά, παράγει στην έξοδο επίσης ένα key-value pair. Στη συνέχεια μια συνάρτηση Reduce συγκεντρώνει όλα τα key-value pairs που έχουν το ίδιο key και τα ενώνει σε ένα.

Για παράδειγμα μπορούμε να πολλαπλασιάσουμε όλα τα στοιχεία μιας λίστας με το 2 και να τα αθροίσουμε ως εξής:



Σε ένα πρόγραμμα Map Reduce μπορεί να υπάρχουν πολλά στάδια όπου η έξοδος του ενός είναι η είσοδος του επόμενου. Για παράδειγμα μπορούμε εύκολα να υπολογίσουμε πόσες φορές εμφανίζεται μια λέξη σε ένα κείμενο ως εξής:



Το μοντέλο αυτό ήταν τόσο επαναστατικό που επηρέασε τον τρόπο που γράφονται παράλληλα προγράμματα και οδήγησε στη δημιουργία πολλών projects ανοιχτού λογισμικού που υιοθετούν την αρχιτεκτονική του, με κυριότερο το Hadoop.

1.4.1. Hadoop

Η δημιουργία του Hadoop επηρεάστηκε άμεσα από το αντίστοιχο paper της Google για το Map Reduce. Η ιδέα ήταν να φτιαχτεί ένα τέτοιο σύστημα ευρείας χρήσης, ώστε να μπορεί να χρησιμοποιηθεί από οποιονδήποτε. Το Hadoop παρέχει ένα σύστημα κατανεμημένης αποθήκευσης αρχείων και κατανεμημένης επεξεργασίας αντίστοιχο με το Map Reduce.

Συγκεκριμένα αποτελείται από τα παρακάτω μέρη:

* Hadoop Distributed File System (HDFS): είναι το σύστημα που διαχειρίζεται το σύστημα αρχείων του κάθε υπολογιστή και το εμφανίζει στο χρήστη σαν ένα ενιαίο αποθηκευτικό χώρο, επιτρέποντας την αποθήκευση μεγάλου όγκου δεδομένων. Το κάθε αρχείο στο HDFS χωρίζεται σε blocks συνήθως των 128 ΜΒ και τα blocks αυτά μοιράζονται στους κόμβους του συστήματος. Ένα block μπορεί να αποσταλεί σε περισσότερους του ενός κόμβους (replication) τόσο για λόγους απόδοσης όσο και προστασίας από την απώλεια δεδομένων.
* Hadoop YARN: το σύστημα αυτό ασχολείται με τη δρομολόγηση των εργασιών στους κόμβους του συστήματος. Όταν ένας χρήστης υποβάλλει μια εργασία αυτή φτάνει στο master node. Αυτός ο κόμβος είναι υπεύθυνος για την δρομολόγηση των εργασιών στους υπολογιστικούς κόμβους, αλλά και για την παρακολούθηση της λειτουργία τους ώστε να σε περίπτωση αστοχίας ενός κόμβου να δρομολογήσει την εργασία σε άλλον κόμβο.
* Map Reduce: Το προγραμματιστικό μοντέλο το οποίο περιγράψαμε παραπάνω, με βάση το οποίο γράφονται οι εφαρμογές των χρηστών.

Ο κάθε κόμβος του Hadoop cluster διαβάζει τα δεδομένα από το δίσκο του, τα επεξεργάζεται και στη συνέχεια γράφει τα αποτελέσματα πάλι εκεί. Επίσης οποιαδήποτε ενδιάμεσα αποτελέσματα στις εργασίες Map Reduce γράφονται επίσης στο δίσκο. Καθώς ο δίσκος είναι πολύ πιο αργός από την κύρια μνήμη, αυτό μπορεί να αποτελέσει bottleneck. Αυτό οδήγησε μια ομάδα ερευνητών από το πανεπιστήμιο Berkeley της California στη δημιουργία του Spark.

1.4.2. Spark

Το Spark, όπως και το Hadoop επιτρέπει την επεξεργασία δεδομένων μεγάλης κλίμακας. Αντίθετα με το Hadoop, έχει τη δυνατότητα να αποθηκεύει τα δεδομένα στη μνήμη, επιτρέποντας έως και 100 φορές πιο γρήγορη επεξεργασία δεδομένων για εφαρμογές που κάνουν έντονη χρήση IO. Τα δεδομένα αποθηκεύονται στη μνήμη ως RDDs (Resilient Distributed Datasets) και μπορούν να χρησιμοποιηθούν άμεσα για επεξεργασία. Το αποτέλεσμα της επεξεργασίας μπορεί επίσης να είναι ένα RDD και να χρησιμοποιηθεί για επόμενες εργασίες. Το Spark δεν είναι αντικαταστάτης του Hadoop, αλλά αντίθετα χρησιμοποιεί το κατανεμημένο σύστημα αρχείων του (HDFS).

Το σύστημα αποτελείται από ένα σύνολο κόμβων, τους workers, οι οποίοι αναλαμβάνουν την επεξεργασία δεδομένων, καθώς και ένα κόμβο, τον master, ο οποίος αναλαμβάνει την ανάθεση εργασιών στους workers και το συντονισμό τους.

Μια εγκατάσταση spark αποτελείται από τα παρακάτω μέρη:

* Worker: Ο κάθε worker είναι ένας αυτόνομος υπολογιστικός κόμβος, ο οποίος δεν έχει κάποια επικοινωνία με τους υπόλοιπους workers. Αντίθετα, περιμένει ανάθεση εργασιών από τον master, τις οποίες εκτελεί μέσω των executors που διαθέτει. Οι executors είναι διεργασίες που τρέχουν στον worker προκειμένου να εκτελεστούν οι εργασίες που έχουν ανατεθεί σε αυτόν. Οι υπολογισμοί εκτελούνται τοπικά, πάνω στα δεδομένα που διαθέτει ο worker. Συγκεκριμένα, ο κάθε executor παίρνει ένα partition του RDD και το επεξεργάζεται, με την διαδικασία αυτή να επαναλαμβάνεται μέχρι οι executors να επεξεργαστούν όλα τα partitions του worker που αντιστοιχούν στο υπό επεξεργασία RDD. Με τον τρόπο αυτό αποφεύγεται η μεταφορά δεδομένων στο δίκτυο. Μόλις οι διεργασίες εκτελεστούν, τα αποτελέσματα μεταφέρονται πίσω στον master.
* Master: Ο master είναι ο μόνος που γνωρίζει την ύπαρξη όλων των workers του cluster. Σε αυτόν τρέχει ο driver, η οποία είναι η διεργασία που τρέχει τον κώδικα του χρήστη. Στη διεργασία αυτή δημιουργείται το SparkContext, το αντικείμενο μέσω του οποίου πραγματοποιείται η επικοινωνία με τους workers. Mέσω heartbeats που λαμβάνει περιοδικά από τους workers, το SparkContext γνωρίζει την κατάστασή τους καθώς και την πορεία των εργασιών. Μόλις ένας executor ολοκληρώσει τη διαδικασία που του έχει ανατεθεί, στέλνει τα αποτελέσματα στο SparkContext.

Όμως για ποιο λόγο το Spark είναι τόσο ταχύτερο από το Hadoop;