

# Θέματα Κατανεμημένων και Παράλληλων Συστημάτων

## ΜΑΘΗΜΑ #1

### Εισαγωγή

# Κατανεμημένα Συστήματα

## Ορισμός #1

- Μια συλλογή από αυτόνομους υπολογιστές που συνδέονται μεταξύ τους μέσω ενός δικτύου και χρησιμοποιούν ειδικά σχεδιασμένο λογισμικό για την παροχή ενοποιημένων υπολογιστικών υπηρεσιών
  - Οι διεργασίες που εκτελούνται από τους δικτυωμένους υπολογιστές, επικοινωνούν μεταξύ τους και συντονίζουν τις κινήσεις τους μέσω της ανταλλαγής μηνυμάτων
- Πρόκειται περί ενός πληροφοριακού συστήματος πολλαπλών αυτόνομων υπολογιστικών στοιχείων, τα οποία συνεργάζονται για την επίτευξη ενός κοινού στόχου (Burns & Willings)

# Κατανεμημένα Συστήματα

## Ορισμός #2

- Κατανεμημένο σύστημα είναι ένα σύνολο από ανεξάρτητους υπολογιστές το οποίο παρουσιάζεται στους χρήστες σαν ένα λογικό ενιαίο σύστημα.
- Ο ορισμός έχει 2 πλευρές:
  - Το **υλικό** - τα μηχανήματα είναι αυτόνομα
  - Το **λογισμικό** - οι χρήστες θεωρούν ότι έχουν να κάνουν με ένα ενιαίο σύστημα. Οι διαφορές των υπολογιστών μεταξύ τους ή ο τρόπος με τον οποίο επικοινωνούν δεν απασχολούν τους χρήστες
- “Γνωρίζεις ότι έχεις ένα κατανεμημένο σύστημα όταν η δυσλειτουργία ενός αγνώστου συστήματος δεν σου επιτρέπει να κάνεις τη δουλειά σου” – Leslie Lamport

# Κατανεμημένα Συστήματα

- Δύο εξελίξεις της τεχνολογίας επέτρεψαν την εμφάνιση Κατανεμημένων Συστημάτων (ΚΣ)
  - ισχυροί και φθηνοί μικροεπεξεργαστές
  - δίκτυα δεδομένων ψηλών ταχυτήτων (LANs, WANs)
- Η ανάγκη **ευέλικτης χρήσης κοινών πόρων** σε τοπική κλίμακα (επιχειρήσεις και οργανισμοί) οδήγησαν στα πρώτα ΚΣ τοπικής εμβέλειας.
- Το διαδίκτυο σε συνδυασμό με τη ραγδαία εξάπλωση του προσωπικού υπολογιστή οδήγησαν σε πιο αποκεντρωμένα (και χαλαρά συνδεδεμένα) ΚΣ σε παγκόσμια κλίμακα.
- Τα ΚΣ **επεκτείνονται** πλέον και σε ενσωματωμένα και φορητά συστήματα.

# Χαρακτηριστικά Κατανεμημένων Συστημάτων

- Οι διαφορές μεταξύ των διαφόρων υπολογιστών και οι τρόποι με τους οποίους αυτοί επικοινωνούν παραμένουν **κρυφοί για τους χρήστες**
- Το ίδιο ισχύει και για την **εσωτερική οργάνωση** των κατανεμημένων συστημάτων.
- Οι χρήστες και οι εφαρμογές μπορούν να αλληλεπιδρούν με ένα κατανεμημένο σύστημα **με συνεπή και ομοιόμορφο τρόπο**, ανεξάρτητα από το που και πότε πραγματοποιείται η αλληλεπίδραση
- Τα κατανεμημένα συστήματα θα πρέπει να επιτρέπουν με σχετική **ευκολία την επέκτασή τους** ή την προσαρμογή της κλίμακας του μεγέθους τους

# Τομείς Εφαρμογής

## Ιατρική, βιολογία και γενετική

(διάγνωση νοσημάτων μέσω  
επεξεργασίας ιατρικών εικόνων)

## Αγορές χρήματος

(χρηματοοικονομικές  
αναλύσεις)

Κατανεμημένα  
Συστήματα



## Επιστημονικό και Εκπαιδευτικό κλάδο

(ερευνητικά ινστιτούτα, δημόσια  
έρευνα)

## Βιομηχανικό και Κατασκευαστικό τομέα

(αυτοκινητοβιομηχανία)

# Παραδείγματα

- Διαδίκτυο, παγκόσμιος ιστός, κινητή τηλεφωνία.
- Τηλε\* (εκπαίδευση, ιατρική, συνεργασία, ...).
- Ενδο-και δια-επιχειρησιακά συστήματα (όμιλος εταιρειών, τράπεζες, ηλεκτρονικό επιχειρείν, ...).
- Αυτόματος έλεγχος(παραγωγή, μεταφορές, ...).
- Ενοποιημένοι υπολογιστικοί και αποθηκευτικοί πόροι σε παγκόσμια κλίμακα (grid computing).
- Ομότιμα συστήματα υπηρεσιών (peer-to-peer).
- Δίκτυα αισθητήρων (sensor networks).
- Έξυπνα κτήρια, συστήματα και περιβάλλοντα αλληλεπίδρασης (pervasive/ubiquitous computing).

# Εφαρμογή: Ο Παγκόσμιος Ιστός

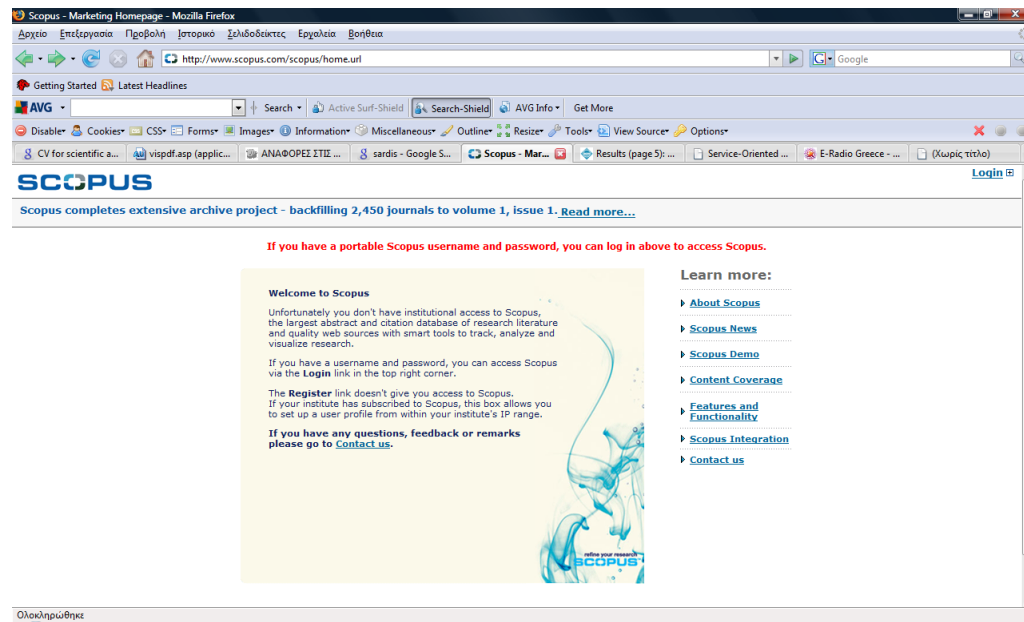
- Αποτελεί ένα τεράστιο σύστημα πληροφοριών βασιζόμενο σε έγγραφα, όπου κάθε έγγραφο έχει ένα δικό του όνομα με τη μορφή μιας διεύθυνσης URL
- Από **λογική άποψη** φαίνεται σαν να υπάρχει ένας και μόνο διακομιστής
- Από **φυσική άποψη** ο Ιστός είναι κατανεμημένος σε έναν τεράστιο αριθμό διακομιστών, κάθε ένας από τους οποίους χειρίζεται έναν αριθμό εγγράφων του Ιστού.
- Το όνομα του διακομιστή που χειρίζεται ένα έγγραφο είναι **κωδικοποιημένο** μέσα στη διεύθυνση URL αυτού του εγγράφου

# Εφαρμογή: Ο Παγκόσμιος Ιστός

- Αποτελεί ένα τεράστιο σύστημα πληροφοριών βασιζόμενο σε έγγραφα, όπου κάθε έγγραφο έχει ένα δικό του όνομα με τη μορφή μιας διεύθυνσης URL
- Από **λογική άποψη** φαίνεται σαν να υπάρχει ένας και μόνο διακομιστής
- Από **φυσική άποψη** ο Ιστός είναι κατανεμημένος σε έναν τεράστιο αριθμό διακομιστών, κάθε ένας από τους οποίους χειρίζεται έναν αριθμό εγγράφων του Ιστού.
- Το όνομα του διακομιστή που χειρίζεται ένα έγγραφο είναι **κωδικοποιημένο** μέσα στη διεύθυνση URL αυτού του εγγράφου

# Εφαρμογή: Ο Παγκόσμιος Ιστός

- Ο Παγκόσμιος Ιστός (WWW) παρέχει ένα απλό και ομοιόμορφο μοντέλο κατανεμημένων εγγράφων.
- Για να δει ο χρήστης ένα έγγραφο χρειάζεται μόνο να ενεργοποιήσει μια αναφορά, και το έγγραφο εμφανίζεται στην οθόνη.
- Η δημοσίευση ενός εγγράφου είναι απλή. Χρειάζεται μόνο να δώσουμε ένα μοναδικό όνομα με την μορφή μιας διεύθυνσης URL η οποία παραπέμπει σε ένα τοπικό αρχείο που περιέχει το περιεχόμενο του εγγράφου.



# Κατανεμημένη επεξεργασία

- Πολλοί υπολογιστικοί κόμβοι συνεργάζονται χαλαρά (*loosely coupled nodes*)
- Στόχος η κατανομή υπηρεσιών, εφαρμογών, επεξεργαστικής ισχύος, ή δεδομένων
- Δεύτερος στόχος είναι η ταχύτητα
- Η επικοινωνία δεν είναι συνήθως συχνή
- Παρότι οι κόμβοι είναι ετερογενείς το δίκτυο είναι τυποποιημένο
- Οι κόμβοι προστίθενται στο δίκτυο με βάση τη γεωγραφική τους θέση

# Κατανεμημένη επεξεργασία: Παράδειγμα

- Διαχείριση Αυτόματων Ταμειακών Μηχανών (ΑΤΜ) μιας τράπεζας που καλύπτει όλη την Ελλάδα
- Απαιτήσεις: Να μπορεί ο πελάτης να λαμβάνει και να καταθέτει χρήματα σε οποιοδήποτε υποκατάστημα χωρίς να έχει εκεί λογαριασμό

# Παράλληλος Υπολογισμός

Στόχος: Η εύρεση γρήγορων λύσεων σε μεγαλύτερα και περισσότερο σύνθετα προβλήματα:

- προβλήματα επιστημονικού υπολογισμού
- συνδυαστικά προβλήματα και προβλήματα βελτιστοποίησης
- προβλήματα γραφικών και επεξεργασίας εικόνας

# **Need for Parallel Computing - Scientific Computing Demand**

**“Computational modeling and simulation are among the most significant developments in the practice of scientific inquiry in the 20th Century. Scientific computing has become an important contributor to all scientific disciplines.**

**It is particularly important for the solution of research problems that are insoluble by traditional scientific theoretical and experimental approaches, hazardous to study in the laboratory, or time consuming or expensive to solve by traditional means”.**

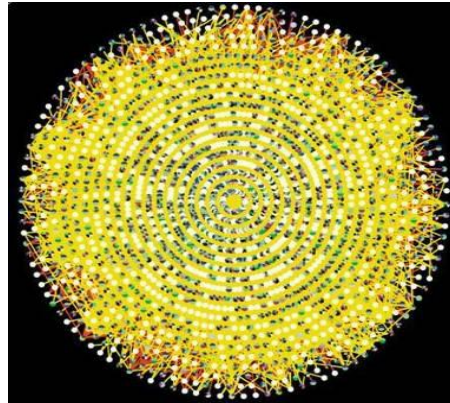
***“Scientific Discovery through Advanced Computing”  
DOE Office of Science, 2000***

# Ανάλυση δεδομένων μεγάλης κλίμακας

**Astrophysics:** massive datasets,  
temporal variations



**Bioinformatics:** data  
quality,  
heterogeneity



**Social Informatics:** new analytics  
challenges

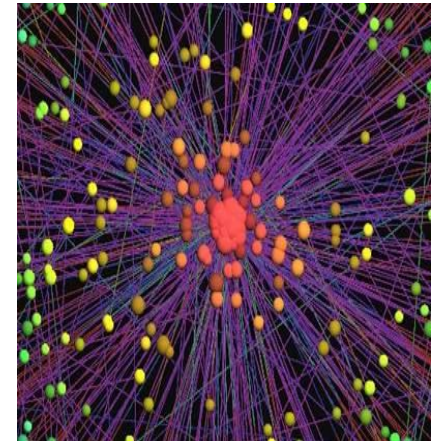


Image sources:


[http://physics.nmt.edu/images/astro/hst\\_starfield.jpg](http://physics.nmt.edu/images/astro/hst_starfield.jpg)


[www.visualComplexity.com](http://www.visualComplexity.com)







# Δρομολόγηση σε συγκοινωνιακά δίκτυα

## Driving Directions

To: Washington, D.C.

 Berkeley, CA  
[Edit or drag the route](#) • [Save this location](#)

↗ A-B: 2809.3 miles, 40 hr 10 min  [Add to route](#)

- 1 Depart Milvia St 0.2 miles
- 2 Turn left onto University Ave 1.8 miles  
*Pass 76 in 0.6 mi*
- 3 Take ramp right for I-80 West / I-580 East / Eastshore Fwy toward Richmond / Sacramento 1.3 miles
- 4 Keep left to stay on I-80 East / Eastshore Fwy 69.3 miles  
 Stop for toll booth
- 5 Take ramp right for I-80 East toward Airport / Reno 651.8 miles  
 Entering Nevada  
 Entering Utah
- 6 Take ramp for I-15 South / I-80 East toward Las Vegas / Cheyenne 2.8 miles
- 7 At exit 304, take ramp right for I-80 East toward Cheyenne 935.0 miles  
 Entering Wyoming  
 Entering Nebraska  
 Entering Iowa



H. Bast et al., “Fast Routing in Road Networks with Transit Nodes”, Science 27, 2007.

# Ανάλυση δεδομένων σε βιολογικά συστήματα

Μελέτη των  
αλληλεπιδράσεων  
των διαφόρων  
συστατικών μερών  
ενός βιολογικού  
συστήματος

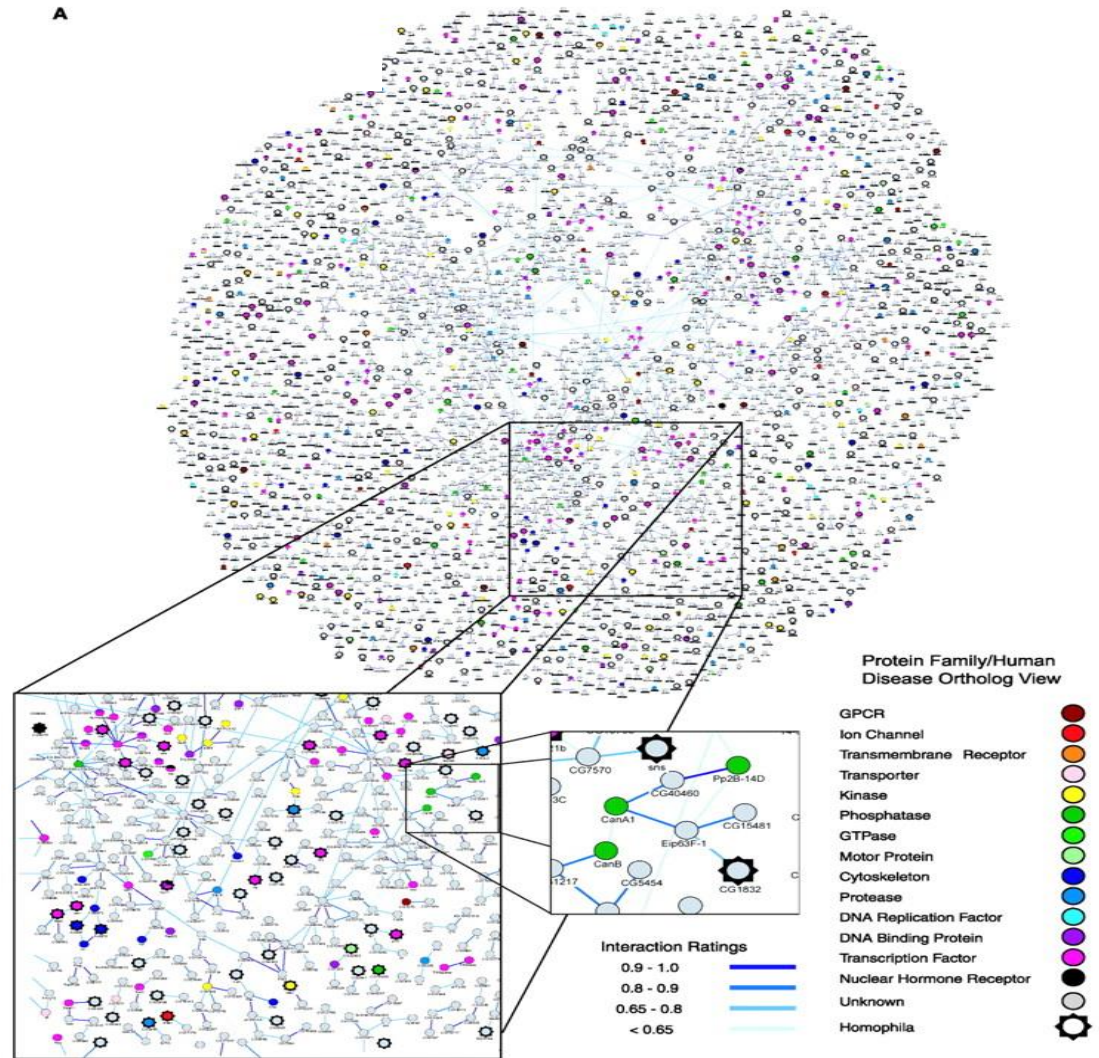
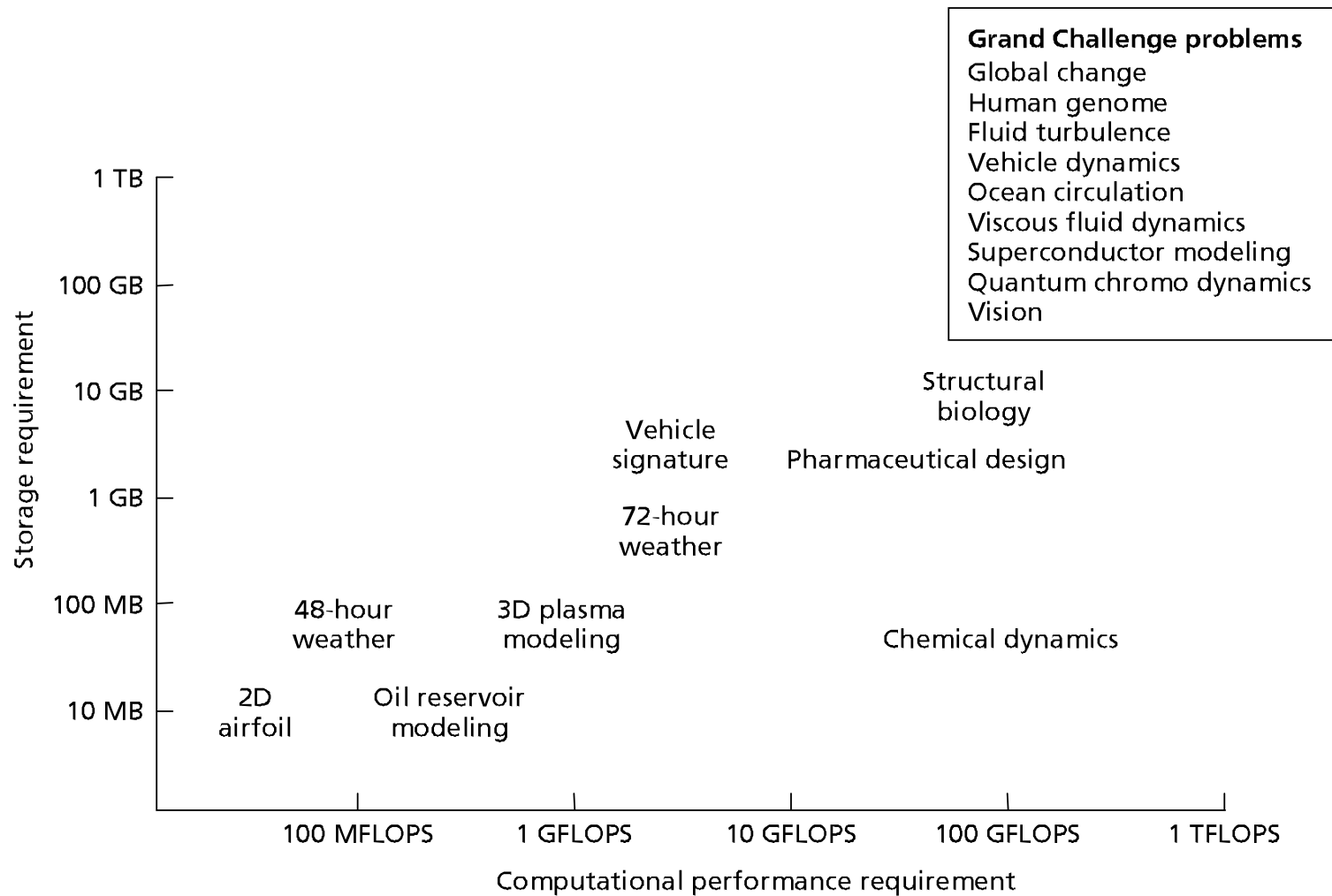


Image Source: Giot et al., “A Protein Interaction Map of *Drosophila melanogaster*”,  
*Science* 302, 1722-1736, 2003.

# Απαιτήσεις Επιστημονικού Υπολογισμού



# FLOPs/ MIPs

**FLOPs: Floating Point Operations per Second**

**MIPs: Million Instructions per Second**

Έστω ότι έχουμε έναν επεξεργαστή που κάνει 1 πράξη κινητής υποδιαστολής σε κάθε κύκλο ρολογιού:

- Αν η συχνότητά του είναι 1GHz, τότε έχει απόδοση 1 GFLOPs

Αν ολοκληρώνει 1 εντολή σε κάθε κύκλο, τότε έχει απόδοση 1000MIPs

**TeraFLOPS (TFLOPS) =  $10^{12}$  FLoating-point Operations Per Second**

**PetaFLOPS =  $10^{15}$  FLoating-point Operations Per Second**

***Petascale Systems:* Supercomputers with  $\geq 1$  PetaFLOPS (peak) performance**

***Aim:* support computationally challenging problems i.e, simulations that are intrinsically multi-scale, or simulations involving interaction of multiple processes**

# Παράλληλα Συστήματα

- ▶ Οι πολλαπλοί επεξεργαστές είναι τοποθετημένοι στην ίδια υπολογιστική μονάδα.
- ▶ Η επικοινωνία μεταξύ των επεξεργαστών είναι γρήγορη.
- ▶ Δεν υπάρχουν προβλήματα συγχρονισμού – υπάρχει ένα καθολικό ρολόι.
- ▶ Οι υπολογιστικές μονάδες είναι υψηλής ποιότητας – σπάνια αποτυγχάνουν.

# Πρώιμοι παράλληλοι υπολογιστές

Σχεδιάστηκαν από τον Daniel Slotnick στο Πανεπιστήμιο του Illinois:

- ο Solomon, ο οποίος κατασκευάστηκε από την Westinghouse Electric Company στις αρχές της δεκαετίας του 1960 και
- ο ILLIAC IV, ο οποίος συναρμολογήθηκε από την Burroughs Corporation στις αρχές της δεκαετίας του 1970.

**12/1996** : ιστορική διάσπαση του υπολογιστικού φράγματος του 1 TFLOPS από τον υπερυπολογιστή Intel ASCI Teraflops (Sandia National Laboratory των Η.Π.Α., στο New Mexico)

**6/2008** : ιστορική διάσπαση του φράγματος του 1 PetaFLOP από τον Roadrunner της IBM (Los Alamos National Laboratory) με απόδοση 1.026 TFLOPS και 122.400 επεξεργαστές.

**8/2/2011:** η IBM ανακοινώνει την κατασκευή του BlueGene/Q: 16,32 PetaFLOPS.

**Από 11/2013 έως και 11/2015:** υψηλότερη απόδοση στον κόσμο από τον Tianhe-2, ο οποίος αναπτύχθηκε από το Εθνικό Πανεπιστήμιο Τεχνολογίας της Άμυνας στην Κίνα: 33,86 PetaFLOPS.

**Από 6/2016 έως 6/2018:** Sunway TaihuLight  
είναι το νέο No. 1 παράλληλο σύστημα: 93  
PetaFLOPS

- Αναπτύχθηκε από το National Research Center of Parallel Computer Engineering & Technology (NRCPC) και βρίσκεται στο National Supercomputing Center στο Wuxi.
- Sunway TaihuLight: 40.960 κόμβοι με 10.649.600 πυρήνες

## TOP500 List - June 2007 (1-10)

Site	Computer	Processors	Year	$R_{\max}$ TF/s
<a href="#">DOE/NNSA/LLNL</a> USA	<a href="#">BlueGene/L - eServer Blue Gene Solution</a> , IBM	131072	2005	280.6
<a href="#">Oak Ridge National Laboratory</a> , USA	<a href="#">Jaguar - Cray XT4/XT3</a> , Cray Inc.	23016	2006	101.7
<a href="#">NNSA/Sandia National Laboratories</a> , USA	<a href="#">Red Storm - Sandia/ Cray Red Storm, Opteron 2.4 GHz dual core</a> , Cray Inc.	26544	2006	101.4
<a href="#">IBM Thomas J. Watson Research Center</a> , USA	<a href="#">BGW - eServer Blue Gene Solution</a> , IBM	40960	2005	91.29
<a href="#">Stony Brook/BNL, NY Center for Computational Sciences</a> , USA	<a href="#">New York Blue - eServer Blue Gene Solution</a> , IBM	36864	2007	82.16
<a href="#">DOE/NNSA/LLNL</a> USA	<a href="#">ASC Purple - eServer pSeries p5 575 1.9 GHz</a> , IBM	12208	2006	75.76
Rensselaer Center for Nanotechnology, USA	<a href="#">eServer Blue Gene Solution</a> IBM	32768	2007	73.03
<a href="#">NCSA</a> , USA	<a href="#">Abe - PowerEdge 1955, 2.33 GHz</a> , Infiniband, Dell	9600	2007	62.68
<a href="#">Barcelona Supercomputing Center</a> Spain	<a href="#">MareNostrum - BladeCenter JS21 Cluster, PPC 970, 2.3 GHz</a> , Myrinet, IBM	10240	2006	62.63
<a href="#">Leibniz Rechenzentrum</a> Germany	<a href="#">HLRB-II - Altix 4700 1.6 GHz</a> , SGI	9728	2007	56.52

# Source: [www.top500.org](http://www.top500.org) – Top List 2018

Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	<b>Summit</b> - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/SC/Oak Ridge National Laboratory United States	2,282,544	122,300.0	187,659.3	8,806
2	<b>Sunway TaihuLight</b> - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
3	<b>Sierra</b> - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM DOE/NNSA/LLNL United States	1,572,480	71,610.0	119,193.6	
4	<b>Tianhe-2A</b> - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 , NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482
5	<b>AI Bridging Cloud Infrastructure (ABCI)</b> - PRIMERGY CX2550 M4, Xeon Gold 6148 20C 2.4GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR , Fujitsu National Institute of Advanced Industrial Science and Technology (AIST) Japan	391,680	19,880.0	32,576.6	1,649

	Παράλληλα Συστήματα	Κατανεμημένα Συστήματα
Στόχος	Επιτάχυνση	1. Τοπολογική κατανομή εφαρμογών, επεξεργαστών ή δεδομένων 2. Ταχύτητα
Δίκτυο	Έμφαση στην ταχύτητα	Ευρείας χρήσης (π.χ. ISDN, Ethernet)
Εφαρμογές	Επιστημονικές, εξομοιώσεις	Υπηρεσίες, Βάσεις δεδομένων
Αποστάσεις	<1m	>1m συχνά >1Km
Τοπολογία	Γεωμετρική	Αυθαίρετη
Κόμβοι	Ομοιογενείς	Πιθανώς ετερογενείς

# Concurrent Programming

***Concurrent Πρόγραμμα:*** περιέχει δύο ή περισσότερες διεργασίες (processes), οι οποίες εργάζονται μαζί για την επίλυση του ίδιου προβλήματος.

***Διεργασία (Process):*** Ακολουθιακό πρόγραμμα (ακολουθία εντολών)

Οι πολλαπλές διεργασίες ενός παράλληλου προγράμματος συνεργάζονται επικοινωνώντας η μια με την άλλη.

***Επικοινωνία με***

- Κοινών μεταβλητών (shared variables ) ή
- Μεταβίβασης Μηνυμάτων (message passing )

# Applications of Concurrent Programming (1/3)

*Multithreaded Computing:* more than one process (thread) that share CPU time

Goal: Good way to organize modern software systems such as:

- Windows on PCs or workstations
- Time-shared and multiprocessor OS
- Real time systems that control application software (power plant, spacecraft, etc)

**Communication via shared variables**

# Applications of Concurrent Programming (2/3)

*Distributed Computing:* Processes execute on machines connected by a communication network

## Goals:

- To off-load processing (file servers in a network)
- To provide access to remote data (DBs & Web)
- To integrate and manage inherently distributed data (enterprise systems) & to increase reliability (fault-tolerant systems).

**Communication via message passing or invoking remote operations.**

# Applications of Concurrent Programming (3/3)

*Parallel Computing:* Processes execute on their own processor on a parallel computer system

Goal: To find faster solutions to larger and more complex problems: scientific computing problems, large combinatorial or optimization problems, graphics and image processing problems

**Communication via shared variables or message passing depending on the parallel architecture.**

# Parallel/Distributed Computing Paradigms

- **Iterative Parallelism**
- **Recursive Parallelism**
- **Producers and Consumers**
- **Clients and Servers/Interacting Peers**

# Iterative Parallelism

**An iterative parallel program contains two or more iterative processes (contain one or more loops)**

- **Each process computes results for a subset of the data.**
- **Then the results are combined**

***Example: Matrix Multiplication***

# **Matrix Multiplication – An embarrassingly parallel application: *A multitude of independent operations***

- **Two operations are independent if the write set of each (the set of variables that it alters and possibly also reads) is disjoint from both the write set and the read set of the other (the set of variables that it reads but does not alter).**
- **It is always safe for two processes to read variables that do not change!**
- ***Matrix multiplication*: the computations of inner products are independent operations**

## *Η εντολή ‘pardo’*

**pardo**

εντολή 1;

...

εντολή N;

**odpar**

**Οι εντολές 1 έως N εκτελούνται παράλληλα**

**Η εντολή “pardo” ολοκληρώνεται όταν ολοκληρωθεί η εκτέλεση όλων των εντολών.**

## *Η εντολή 'pardo'*

```
pardo [ i=0 to n-1 ] {  
    a[i] = 0; b[i] = 0;  
}
```

**Δημιουργεί n διεργασίες – μία για κάθε διαφορετικό i**

# Πολλαπλασιασμός Πινάκων: Ακολουθιακός Υπολογισμός

```
double a[n,n], b[n,n], c[n,n];
```

```
for [i = 0 to n-1] {
```

```
    for [j = 0 to n-1] {
```

```
        # υπολογισμός του γινομένου του a[i,*] με το b[* ,j]
```

```
            c[i,j] = 0.0;
```

```
            for [k = 0 to n-1]
```

```
                c[i,j] = c[i,j] + a[i,k]*b[k,j];
```

```
        }
```

```
}
```

## Πολλαπλασιασμός Πινάκων: Επαναληπτικός Παραλληλισμός κατά Σειρές

```
pardo [i = 0 to n-1] { # παράλληλος υπολογισμός των γραμμών  
  for [j = 0 to n-1] {  
    c[i,j] = 0.0;  
    for [k = 0 to n-1]  
      c[i,j] = c[i,j] + a[i,k]*b[k,j];  
  }  
}
```

Πολλαπλασιασμός Πινάκων:  
Παραλληλισμός κατά Στήλες

```
parodo [j = 0 to n-1] { # παράλληλος υπολογισμός των στηλών  
  for [i = 0 to n-1] {  
    c[i,j] = 0.0;  
    for [k = 0 to n-1]  
      c[i,j] = c[i,j] + a[i,k]*b[k,j];  
  }  
}
```

Πολλαπλασιασμός Πινάκων: Παραλληλισμός  
κατά Σειρές και κατά Στήλες

```
parado [i = 0 to n-1, j = 0 to n-1] {  
    c[i,j] = 0.0;  
    for [k = 0 to n-1]  
        c[i,j] = c[i,j] + a[i,k]*b[k,j];  
}
```

# Recursive Parallelism: MERGESORT

*Input:* An unsorted array  $S$  of  $n$  numbers

**MERGESORT(s)** partitions  $S$  into 2 parts and calls itself.  
The recursive calls are executed in parallel.

**MERGESORT(S)**

float  $S[n]$ ;

if (  $|S|=1$ )

    return  $S$ ;

else

    pardo

$S1 := \text{MERGESORT}(S[0 .. \lfloor |S|/2 \rfloor]);$

$S2 := \text{MERGESORT}(S(\lfloor |S|/2 \rfloor .. |S|));$

return  $\text{MERGE}(S1, S2)$ ;

# Producers and Consumers

***A producer process* computes and outputs a stream of results.**

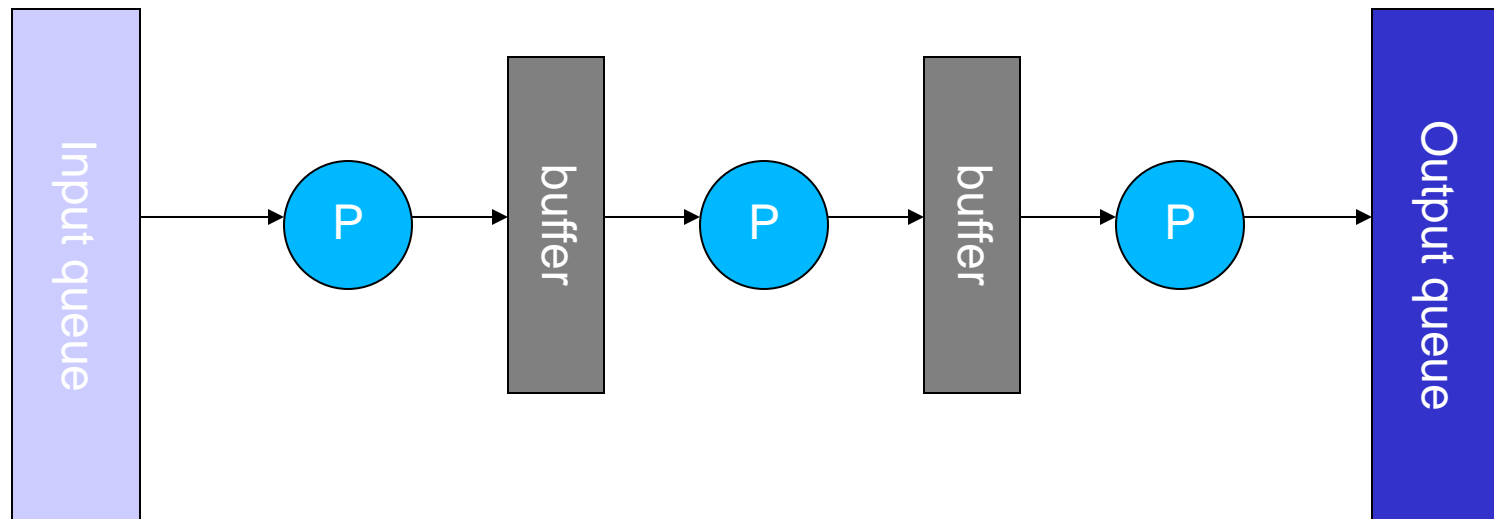
***A consumer process* inputs and analyzes a stream of values.**

**Pipeline: a sequence of processes in which each consumes the output its predecessor and produces output for its successor.**

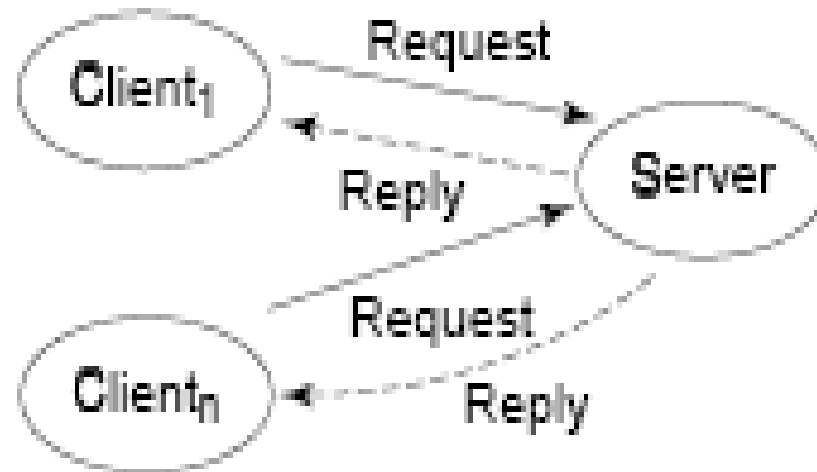
# Pipeline (Σωλήνωση)

- Οι διεργασίες αντιστοιχούν σε στάδια της σωλήνωσης.
- Τα δεδομένα περνούν από τις διαδοχικές διεργασίες (στάδια σωλήνωσης) κάθε μία από τις οποίες εκτελούν κάποια εργασία σε αυτά.

# Pipeline (Σωλήνωση)



# Clients and Servers



Clients and servers.

# Interacting Peers: Distributed Matrix Multiplication Using Coordinator/Worker Interaction

```
double a[n,n]; # source matrix a  
double b[n,n]; # source matrix b  
double c[n,n]; # result matrix c
```

Στην αρχή για κάθε  $i$ , η διεργασία **Worker  $i$**  παραλαμβάνει από τη διεργασία **coordinator** την γραμμή  $i$  του πίνακα **a** και ολόκληρο τον πίνακα **b**.

Η **Worker  $i$**  υπολογίζει την γραμμή  $i$  του πίνακα **c** και τη στέλνει στη διεργασία **coordinator**.

# Πολλαπλασιασμός Πινάκων: Κώδικας coordinator

```
process coordinator {  
    double a[n,n]; # source matrix a  
    double b[n,n]; # source matrix b  
    double c[n,n]; # result matrix c  
    initialize a and b;  
    for [i = 0 to n-1] {  
        send row i of a to worker[i];  
        send all of b to worker[i];    }  
  
    for [i = 0 to n-1]  
        receive row i of c from worker[i];  
    print the results, which are now in matrix c; }
```

# Πολλαπλασιασμός Πινάκων: Κώδικας Worker

```
process worker[i = 0 to n-1] {  
    double a[n]; # row i of matrix a  
    double b[n,n]; # all of matrix b  
    double c[n]; # row i of matrix c  
    receive initial values for vector a and matrix b;  
    for [j = 0 to n-1] {  
        c[j] = 0.0;  
        for [k = 0 to n-1]  
            c[j] = c[j] + a[k] * b[k,j];  
    }  
    send result row (vector c) to the coordinator process;  
}
```

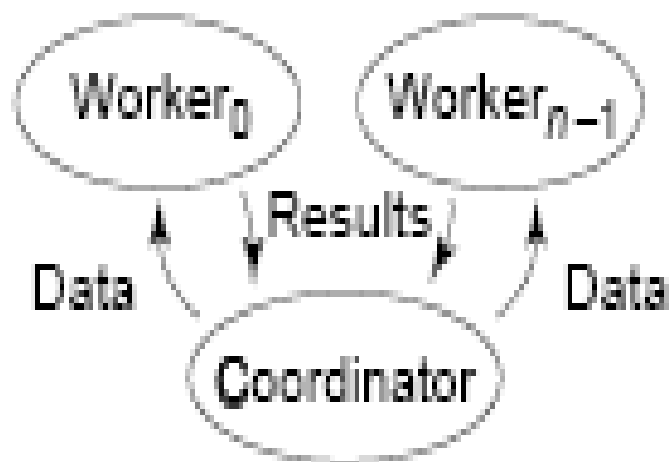
# Πολλαπλασιασμός Πινάκων: Σωλήνωση

Στην αρχή για κάθε  $i$ , η διεργασία **Worker  $i$**  διαθέτει την γραμμή  $i$  του πίνακα  $a$  και τη στήλη  $i$  του πίνακα  $b$ .

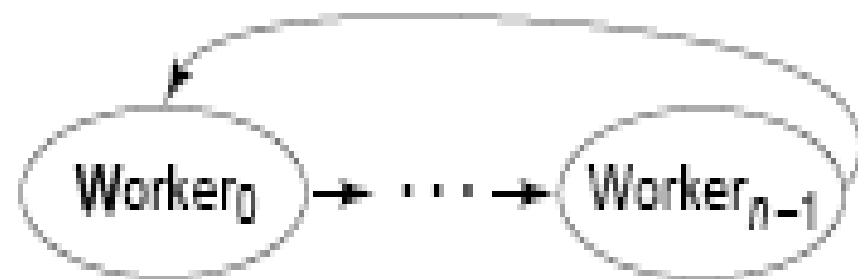
Η **Worker  $i$**  μπορεί να υπολογίσει μόνον την τιμή  $c[i,i]$ .

Κάθε διεργασία **Worker** εκτελεί έναν αριθμό από επαναλήψεις (rounds): σε κάθε επανάληψη στέλνει την στήλη του  $b$ , την οποία εκείνη τη στιγμή κατέχει, στην επόμενη διεργασία **Worker** και παραλαμβάνει μια διαφορετική στήλη του  $b$  από την προηγούμενη διεργασία **worker**.

Η διεργασία **Worker  $i$**  υπολογίζει τη γραμμή  $i$  του πίνακα  $c$ .



(a) Coordinator/worker interaction



(b) A circular pipeline

```

process worker[i=0 to n-1] # has row i of matrix a and column i of matrix b;
    double a[n];           # compute  $c[i,i] = a[i,*] \times b[* ,i]$ 
    double b[n];           for [k = 0 to n-1]
    double c[n];           sum = sum + a[k] * b[k];
    double sum = 0.0;      c[nextCol] = sum;
    int nextCol = i;      # circulate columns and compute rest of
                        c[i,*]
                        for [j = 1 to n-1] {
                            send my column of b to the next worker;
                            receive a new column of b from the
previous worker;
                            sum = 0.0;
                            for [k = 0 to n-1]
                                sum = sum + a[k] * b[k];
                            if (nextCol == 0)
                                nextCol = n-1;
                            else
                                nextCol = nextCol-1;
                            c[nextCol] = sum;

```

Πολλαπλασιασμός  
πινάκων με σωλήνωση

# Περιεχόμενο Μαθήματος

- Εισαγωγή
- Βασικές Έννοιες Παράλληλου Υπολογισμού. Στοιχεία Αρχιτεκτονικής Παράλληλων Μηχανών. Τοπολογίες Δικτύων Διασύνδεσης.
- Παράλληλοι Αλγόριθμοι σε Περιβάλλον Κοινής Μνήμης (αλγόριθμοι ταξινόμησης και συγχώνευσης, υπολογισμός προθεμάτων, υπολογισμοί σε λίστες).
- Προγραμματισμός Κοινής Μνήμης (processes, synchronization, concurrency, Critical Sections, Mutual Exclusion, Barrier Synchronization Semaphores and Monitors)
- Προγραμματισμός Κοινής Μνήμης με Χρήση OpenMP
- Προγραμματισμός επιταχυντών/GPUs (CUDA).

# Περιεχόμενο Μαθήματος

- Παράλληλοι Αλγόριθμοι Κατανεμημένης Μνήμης.
- Προγραμματισμός Κατανεμημένης Μνήμης με χρήση MPI.
- Παράλληλος προγραμματισμός σε υβριδικό περιβάλλον (MPI+OpenMP+CUDA).
- Οργάνωση Κατανεμημένων Συστημάτων ( Το Μοντέλο Πελάτη- Διακομιστή , Ομότιμα Συστήματα (P2P Systems))
- Ενδιάμεσο Λογισμικό, Μοντέλα Ενδιάμεσου Λογισμικού (κλήση απομακρυσμένων διαδικασιών (RPC), κλήση απομακρυσμένων αντικειμένων (RMI))
- Βασικά Θέματα Κατανεμημένου Υπολογισμού (χρονosφραγίδες, εκλογή αρχηγού, αμοιβαίος αποκλεισμός)
- Υπολογισμός σε περιβάλλον συστοιχίας υπολογιστών (Cluster computing) και σε υπολογιστικά πλέγματα (Grid computing).

# Διαδικασία Αξιολόγησης Μαθήματος

(1) Δύο ασκήσεις – προγραμματιστικά θέματα σε OpenMP και MPI αντίστοιχα

(2) Τελική γραπτή εξέταση στο τέλος του εξαμήνου

Ο Τελικός Βαθμός του μαθήματος προσδιορίζεται ως εξής:

$$0,35 \times \text{Βαθμός\_Ασκήσεων} + 0,65 \times \text{Βαθμός\_Τελικής\_Εξέτασης}$$