



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

Συστήματα Παράλληλης & Κατανεμημένης Επεξεργασίας

Ενότητα 1: OpenMPI

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών



Πανεπιστήμιο Δυτικής Μακεδονίας



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Σκοπός της Ενότητας

- Η κατανόηση των βασικών τεχνικών παραλληλοποίησης αλγορίθμων.
- Η παρουσίαση των εισαγωγικών εννοιών στο MPI.



Εισαγωγή

- Εισαγωγή στον παράλληλο προγραμματισμό:
 - Μοντέλα και υλοποιήσεις.
- TCP/IP stack.
- Message Passing Interface:
 - Βασικές κλήσεις.
 - Παραδείγματα.



Τι είναι ο παράλληλος προγραμματισμός;

- Παράλληλη και ταυτόχρονη εκτέλεση διεργασιών σε πολλαπλούς επεξεργαστές με στόχο την ελαχιστοποίηση του συνολικού χρόνου εκτέλεσης.
- Κάθε επεξεργαστής έχει ένα αντίγραφο του ίδιου εκτελέσιμου.
- Επικοινωνία μεταξύ των επεξεργαστών στη βάση ενός πρωτοκόλλου:
 - MPI.
 - OpenMP.
 - PVM.



Κατάλληλο υλικό για παράλληλο προγραμματισμό

- Πολυεπεξεργαστές Κοινής Μνήμης (SMPs – Symmetric MultiProcessors)
 - Συστήματα με επεξεργαστές πολλαπλών πυρήνων (Multi-core, Many-core)
 - Συστήματα με Hyper-Threaded επεξεργαστές
 - Συστήματα με 2 ή περισσότερους επεξεργαστές
- Υπολογιστές διασυνδεδεμένοι με γρήγορο δίκτυο
 - Networks of Workstations (NOWs)
 - Networks of SMPs
 - Συστάδες υπολογιστών (Clusters)
 - Πλέγματα (Grids) με έμφαση στον υπολογισμό (Computational Grids) ή στα δεδομένα (Data Grids)
- Ετερογενείς αρχιτεκτονικές
 - Cell Broadband Engine
 - Σχήματα CPU με GPUs (GPGPUs – General Purpose computing on Graphical Processing Units)



Υποδομή – Λογισμικό Συστήματος

- Όλα τα σύγχρονα λειτουργικά συστήματα υποστηρίζουν πολυεπεξεργαστά συστήματα (Linux, Windows, MacOSX, [Free,Open,Net]BSD, Solaris).
- Στο πλαίσιο του μαθήματος επιλέγεται το FreeBSD ή το Ubuntu.
- Οι τελικές μετρήσεις και τα πειράματα θα διενεργηθούν σε συστήματα του τμήματος με FreeBSD ή του ωκεανός με Ubuntu.
- Ωστόσο για την σταδιακή υλοποίηση των εργαστηριακών ασκήσεων παρέχεται μερική υποστήριξη για ανάπτυξη σε Windows, MacOSX.
- Εργαλεία για την διευκόλυνση της εγκατάστασης FreeBSD/Ubuntu και της ανάπτυξης στο σπίτι:
 - Virtualization μέσω του VMware Player (free).
 - Virtualization μέσω του VirtualBox (free).
 - Απομακρυσμένη σύνδεση με SSH σε διακομιστή.



Προγραμματιστικά Περιβάλλοντα

- Κάθε προγραμματιστικό μοντέλο κάνει διαθέσιμο και ένα περιβάλλον ανάπτυξης.
- Αυτά τα περιβάλλοντα συνήθως αποτελούνται από:
 - τις απαραίτητες βιβλιοθήκες λογισμικού (run-time system),
 - ένα συμβατό μεταφραστή (compiler),
 - εργαλεία ανάπτυξης (editors, debuggers, profilers).



Παράλληλος Προγραμματισμός με ανταλλαγή μηνυμάτων

- Χρήση του προτύπου MPI (Message Passing Interface).
- Ορίζεται για τις γλώσσες C/C++ και Fortran.
- Διαθέσιμο σε διάφορες υλοποιήσεις (MPICH, LAM/MPI, OpenMPI).
- Χαρακτηριστικά του MPI:
 - Ρητή ενορχήστρωση του παραλληλισμού με ανταλλαγή μηνυμάτων.
 - Διαθέσιμο τόσο σε αρχιτεκτονικές κατανεμημένης όσο και σε κοινής μνήμης.
 - Παροχή μεγάλης ευελιξίας στην ανταλλαγή δεδομένων και την επικοινωνία των διεργασιών.
 - De facto πρώτη επιλογή για προγραμματισμό υψηλών επιδόσεων σε αρχιτεκτονικές κατανεμημένης μνήμης.



Παράλληλος Προγραμματισμός με νήματα

- Για πολυνηματικό προγραμματισμό με χρήση της γλώσσας C, επιλέγεται στο πλαίσιο του μαθήματος η βιβλιοθήκη των νημάτων Posix Threads (Pthreads).
- Χαρακτηριστικά των Pthreads:
 - Ευρεία χρήση.
 - Ώριμη υλοποίηση και πλατιά υποστήριξη από την κοινότητα.
 - Αποδοτικές υλοποιήσεις στα σύγχρονα λειτουργικά συστήματα.
 - Ικανοποιητικός βαθμός μεταφερσιμότητας.
 - Διαθέσιμη βιβλιογραφία μέσω Διαδικτύου.
 - Εύχρηστη και καλά ορισμένη προγραμματιστική διεπαφή.



Προγραμματισμός σε ετερογενή σχήματα

- Για τον προγραμματισμό σε υβριδικό σχήμα συνεργασίας CPU και GPU ορίζονται επίσης διεπαφές (APIs) και διατίθενται κατάλληλα περιβάλλοντα ανάπτυξης:
 - π.χ., το περιβάλλον ανάπτυξης CUDA για τον προγραμματισμό NVIDIA multi-core GPUs το οποίο χρησιμοποιεί σαν όχημα τη γλώσσα προγραμματισμού C.
- Αντίστοιχα για τον επεξεργαστή Cell υπάρχει διαθέσιμο το περιβάλλον ανάπτυξης Cell SDK από την IBM το οποίο περιλαμβάνει επιπλέον και προσομοιωτή.



Μετάφραση

Παράλληλων προγραμμάτων

- Πολλοί μεταφραστές διαθέσιμοι για υποστήριξη των παραπάνω προγραμματιστικών μοντέλων.
- Στο πλαίσιο του μαθήματος θα χρησιμοποιηθεί ο μεταφραστής GCC (GNU C COMPILER).
- Από την έκδοση 4.2 και έπειτα υποστηρίζει και το OpenMP (4.1 στο Fedora Linux).
- Κατά την ανάπτυξη χρησιμοποιήστε μόνο τις βασικές ή τις απαραίτητες παραμέτρους.
- Κατά τις τελικές μετρήσεις μπορείτε να αξιοποιήσετε βελτιστοποιήσεις που παρέχονται από τον μεταφραστή (παραμέτροι `-O`, `-O2`, `-O3`).
- Κατά τη χρήση του MPI ο `gcc` χρησιμοποιείται από το εκτελέσιμο `mpicxx`.



Εργαλεία ανάπτυξης

- Για την ανάπτυξη των εργαστηριακών ασκήσεων και την καλύτερη επόπτευση σε κάθε στάδιο προτείνουμε την χρήση απλών εργαλείων.
 - Ένας editor για την συγγραφή (vim, emacs, nano).
 - Χρήση του μεταφραστή gcc από τη γραμμή εντολών.
 - Χρήση του debugger gdb όπου αυτό ενδείκνυται.
 - Χρήση του εργαλείου GNU make για την συνολική μετάφραση των προγραμμάτων.
 - Εκτέλεση απευθείας από τη γραμμή εντολών (στο MPI με την εντολή mpirun).



Άλλα περιβάλλοντα

- Επίσης υπάρχουν και περιβάλλοντα IDE (Integrated Development Environment) όπως το Eclipse, το Visual Studio, το Dev-c++, KDevelop. Ωστόσο τέτοια περιβάλλοντα δεν θα είναι διαθέσιμα μέσω απομακρυσμένης πρόσβασης στα συστήματα που θα πάρετε μετρήσεις.
- Για περιβάλλον Windows στην περίπτωση που θέλετε να χρησιμοποιήσετε οπωσδήποτε ένα τέτοιο περιβάλλον, προτείνεται το Dev-c++ για την καλύτερη συνεργασία του με το MinGW και τον παρεχόμενο gcc compiler.



Makefile

```
CC = gcc
CFLAGS = -Wall -g
INCLUDES = -I/path/to/custom/include
LIBS = -L/path/to/custom/lib

all: ask1 ask2

ask1: ask1.o
    $(CC) $(LIBS) -o ask1

ask2: ask2.o
    $(CC) $(LIBS) -o ask2

ask1.o: ask1.c ask1.h common.c
    $(CC) $(CFLAGS) $(INCLUDES) -c ask1.c

ask2.o: ask2.c ask2.h common.c
    $(CC) $(CFLAGS) $(INCLUDES) -c ask2.c

clean:
    rm-f ask1 ask2 *.o core.*
```



Επισκόπηση προγραμμάτων

- Το debugging των παράλληλων εφαρμογών αποτελεί σύνθετη διαδικασία ενώ συχνά αποδεικνύεται πρακτικά ανέφικτο.
- Η βηματική εκτέλεση δεν υφίσταται όπως στην περίπτωση των ακολουθιακών προγραμμάτων λόγω της παρουσίας πολλών ροών εκτέλεσης ταυτόχρονα.



Χρονομέτρηση

- Αρκετοί τρόποι ...
- Ωστόσο άλλοι περισσότερο και άλλοι λιγότερο ακριβείς ...
 - Η πιο διαδεδομένη και μεταφέρσιμη επιλογή για την C είναι η συνάρτηση `gettimeofday()`.
 - Παρέχει ικανοποιητική ακρίβεια και υποστηρίζεται από τα περισσότερα λειτουργικά συστήματα.
- Επιστρέφει τον χρόνο από μια συγκεκριμένη στιγμή στο παρελθόν (Epoch - (00:00:00 UTC, January 1, 1970) σε μια δομή που περιέχει τα `seconds` και τα `microseconds`.
- Χρησιμεύει ...



Εργαστήριο...

- Μετά τον καθορισμό των ομάδων θα σας δοθεί πρόσβαση σε υπολογιστικά συστήματα του τμήματος ή του GRNET για την ανάπτυξη και την διενέργεια μετρήσεων.
- Στα συγκεκριμένα συστήματα θα μπορείτε να έχετε μόνο απομακρυσμένη πρόσβαση.
- Συγκεκριμένα θα μπορείτε να έχετε πρόσβαση σε login shell μέσω του προγράμματος ssh, ενώ για την μεταφορά αρχείων θα χρησιμοποιείτε το αντίστοιχο sftp (command line εντολές σε Linux).
- Για κάθε γραφικό περιβάλλον υπάρχουν κατάλληλοι clients για χρήση ssh/sftp (π.χ. Windows: Putty/psftp, WinScp, MacOSX: Cyberduck κ.λ.π.).



Εργαστήριο..

- Τα υπολογιστικά συστήματα θα είναι εφοδιασμένα με το απαραίτητο λογισμικό για να αναπτύξετε/διορθώσετε/εκτελέσετε τις εφαρμογές σας σύμφωνα με τον προτεινόμενο τρόπο που αναφέρθηκε προηγουμένως (text editor, gcc, make, ...). Δεν παρέχεται πρόσβαση σε γραφικό περιβάλλον, εκτός αν το εγκαταστήσετε μόνοι σας.
- Χρήσιμη εφαρμογή για την χρήση πολλαπλών τερματικών σε μια απομακρυσμένη σύνδεση και την εκτέλεση πειραμάτων στο background είναι η εφαρμογή **screen** (man screen).



Προτεινόμενοι τρόποι διάγνωσης

- Για τις ανάγκες του μαθήματος, οι προτεινόμενοι τρόποι διάγνωσης προβλημάτων και επισκόπησης της εκτέλεσης είναι:
 - Εκτύπωση κατάλληλων μηνυμάτων σε συγκεκριμένα σημεία του προγράμματος με χρήση της `fprintf` και ανακατεύθυνση στο `stderr` π.χ. `fprintf(stderr, "[%d] message data values ... \n", id, var1, var2);`
 - Χρήση debugger για τον εντοπισμό του σημείου όπου συμβαίνουν κρίσιμα σφάλματα.
 - π.χ. χρήση `gdb` για τον εντοπισμό `segmentation faults`.
 - Ανακατεύθυνση μηνυμάτων στο τερματικό ή σε ξεχωριστά αρχεία για κάθε νήμα/διεργασία.



Χαρακτηριστικά ετερογενών συστημάτων

- Απαιτούν ρητή και προσεκτικά σχεδιασμένη ενορχήστρωση του παραλληλισμού με υβριδικά σχήματα κοινής μνήμης και ανταλλαγής μηνυμάτων για να λειτουργήσουν αποδοτικά.
- Στοχεύουν στην αποδοτική εκτέλεση κατηγοριών εφαρμογών με συγκεκριμένα χαρακτηριστικά (εφαρμογές γραφικών, επιστημονικές εφαρμογές).
- Ο παραγόμενος κώδικας είναι εξαιρετικά εξειδικευμένος και δεν είναι μεταφέρσιμος σε διαφορετικά σχήματα.

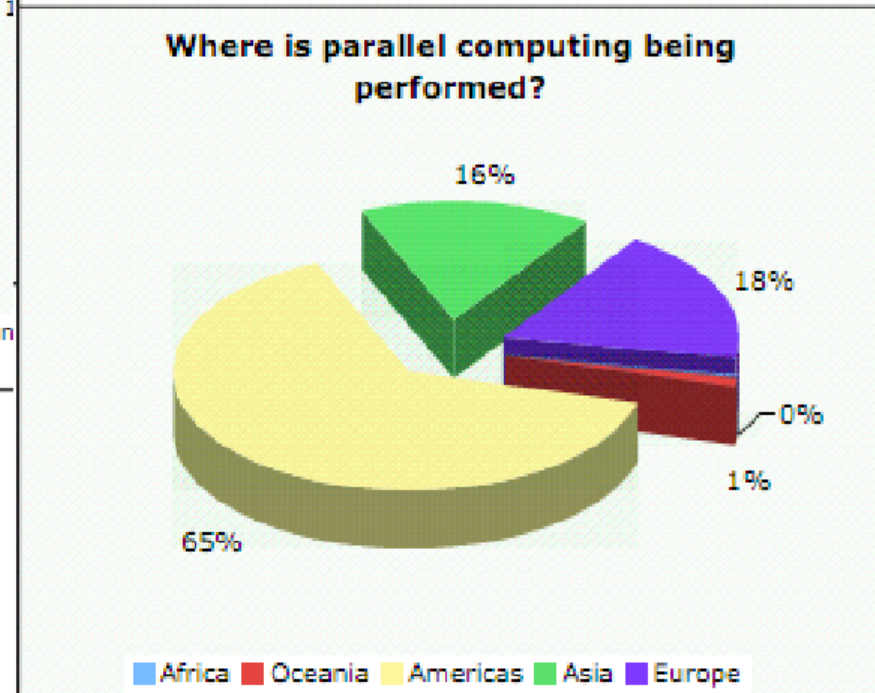
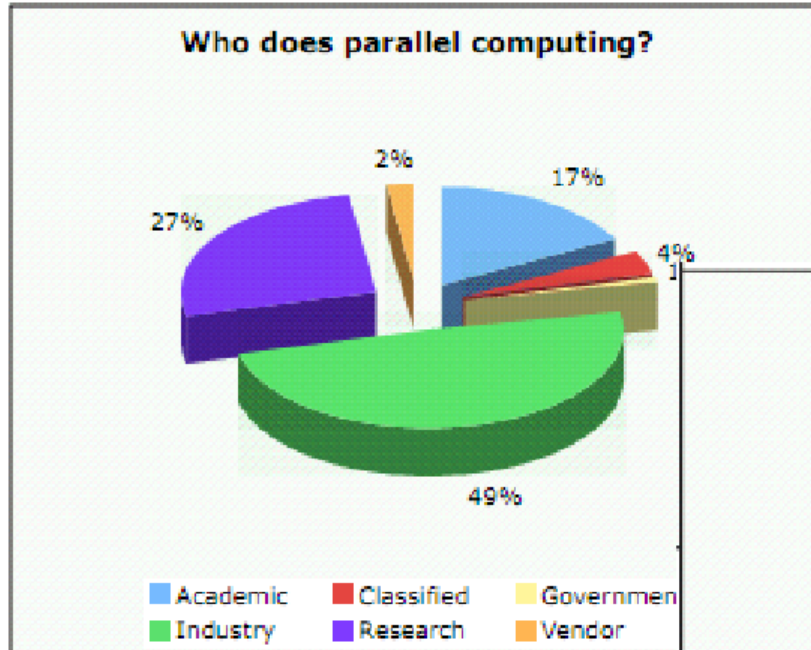


Pthreads και λειτουργικά συστήματα

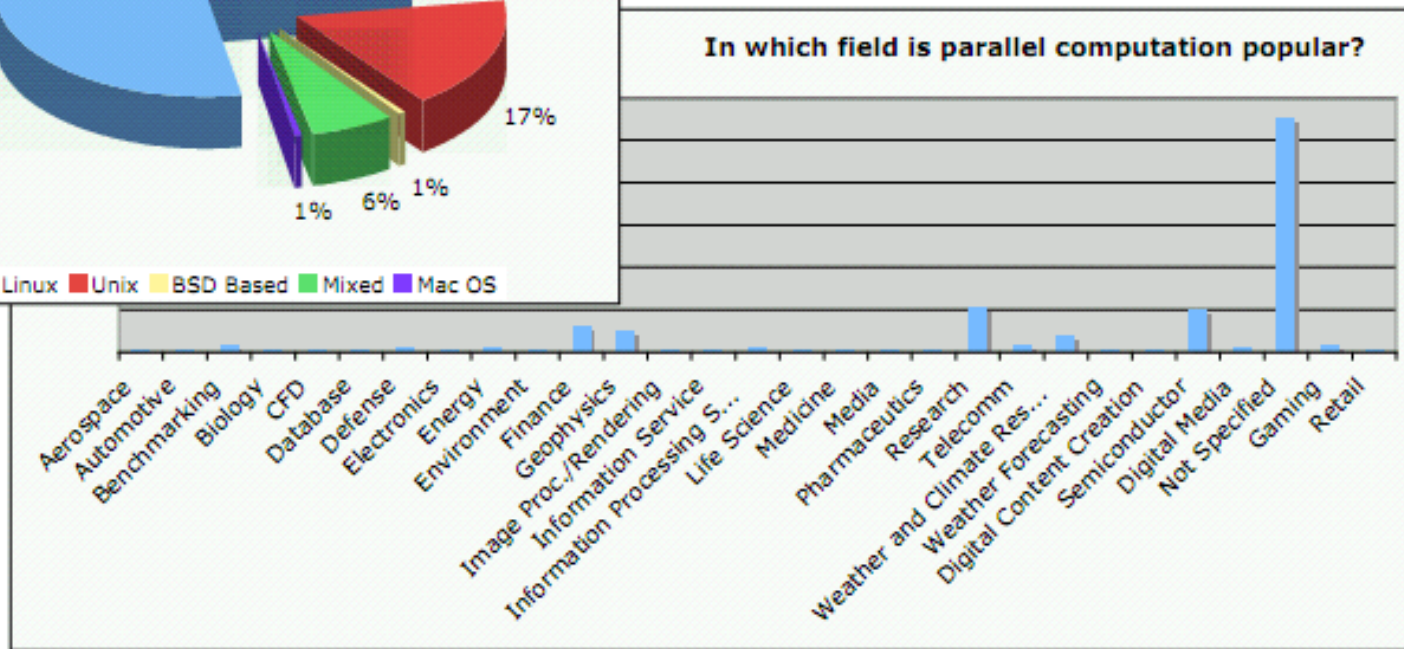
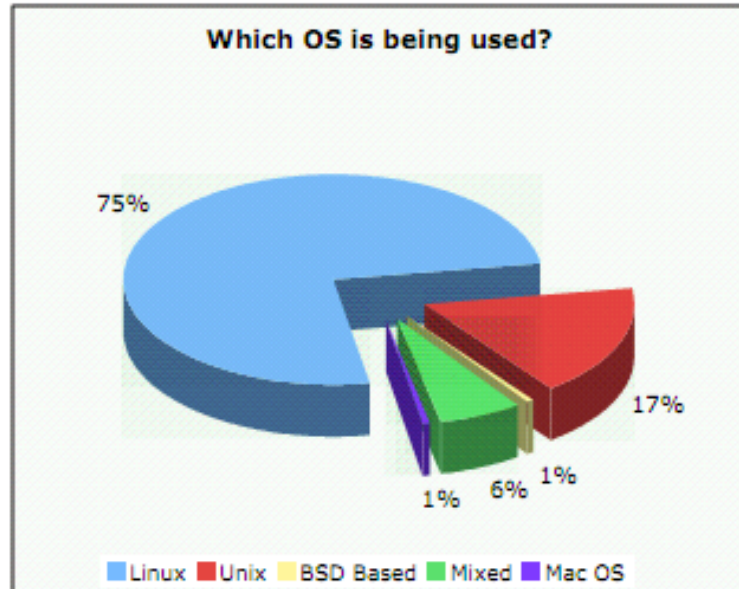
- Σε Linux και MacOSX βρίσκονται προεγκατεστημένα.
- Σε Windows προτείνουμε την εγκατάσταση τους με χρήση του πακέτου.
- Pthreads-win32 και των εργαλείων MinGW.
 - <http://sourceware.org/pthreads-win32>
 - <http://www.mingw.org/>
- Μετά την εγκατάσταση οι εντολές θα είναι διαθέσιμες από την γραμμή εντολών.



Λίγα στατιστικά για τον παράλληλο προγραμματισμό (1/2)



Λίγα στατιστικά για τον παράλληλο προγραμματισμό (2/2)



4 είναι τα μοντέλα παράλληλου προγραμματισμού

- Διαμοιραζόμενη μνήμη.
- Message Passing:
 - Send, Receive, Broadcast.
- Transparent:
 - Ο compiler αναλαμβάνει τον παραλληλισμό.
- Directive based:
 - Στη βάση εντολών ο compiler παράγει παράλληλο κώδικα.
- Λοιπά (BSP, openMP).

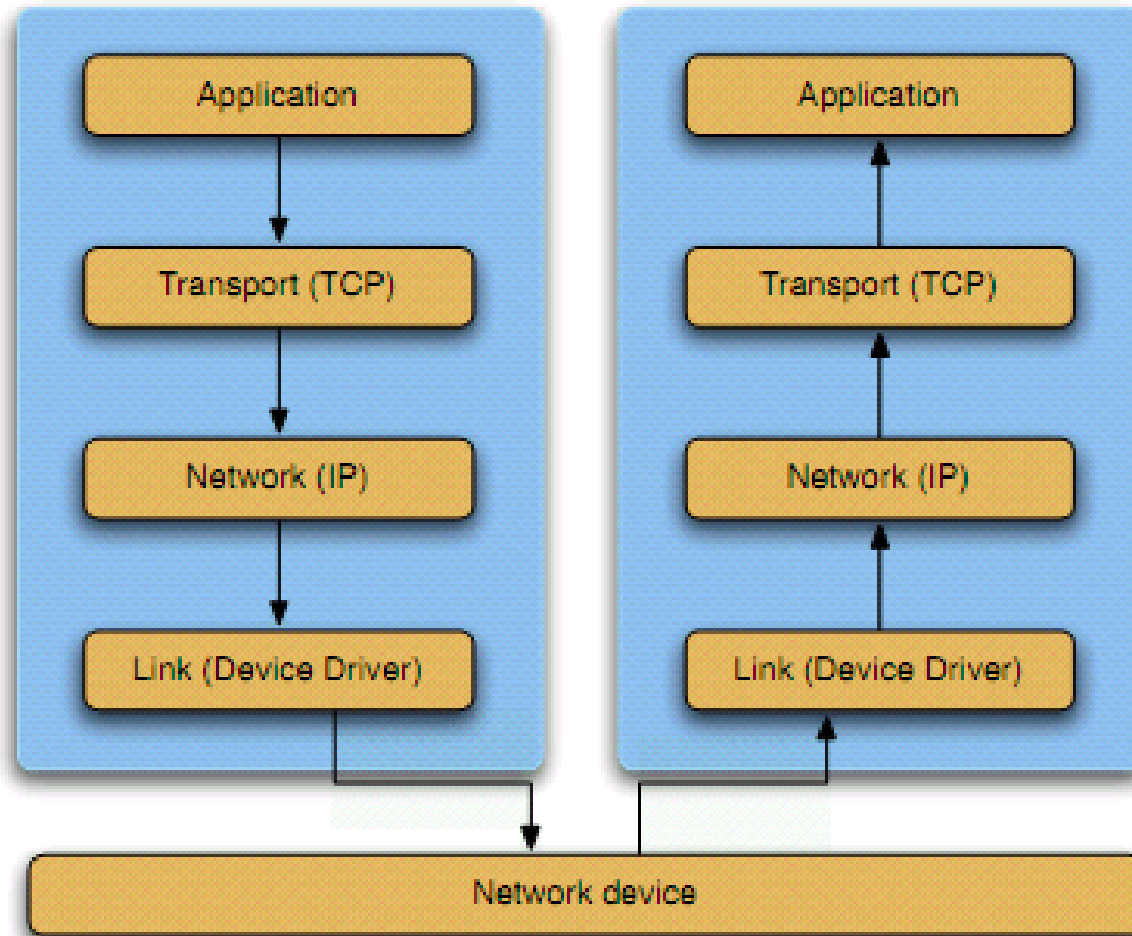


Το μοντέλο Message Passing

- Το εκτελέσιμο αντιγράφεται πολλαπλές φορές και η κάθε κópια εκτελείται από κάποια μοναδική διεργασία.
- Η μνήμη της κάθε διεργασίας είναι μοναδική και προσβάσιμη μόνο από την ίδια διεργασία.
- Ο προγραμματιστής αναλαμβάνει:
 - να μετακινεί τα δεδομένα ανάμεσα στις διεργασίες,
 - την κατανομή δεδομένων ανάμεσα στις διεργασίες,
 - το συντονισμό των διεργασιών για συλλογικές πράξεις.



Το πρωτόκολλο επικοινωνίας TCP/IP



Message Passing Interface

- Το MPI είναι ένα πρωτόκολλο:
 - <http://www.mpi-forum.org/>
 - MPI-1 και MPI-2.
- Από το πρωτόκολλο στις βιβλιοθήκες.
- MPI-1:
 - mpich-1: <http://www-unix.mcs.anl.gov/mpi/mpich1/>
 - lam/mapi: <http://www.lam-mpi.org/>
 - MPI-2:
 - mpich-2: <http://www-unix.mcs.anl.gov/mpi/mpich2/>
 - openMPI: <http://www.open-mpi.org/>



MPI-2

- Το MPI-2 είναι μια βελτίωση του MPI-1 ως προς:
 - Προτυποποίηση συναρτήσεων. Όλες ξεκινάνε με `mri_`.
 - Την ανοχή σε σφάλματα. Αν μια διεργασία στο παρελθόν κολλούσε τότε δημιουργούνταν πρόβλημα στους communicators.
 - Τη δυναμική προσαρμογή του συστήματος. Μπορούν να προστεθούν νέοι κόμβοι επεξεργασίας.
 - Υποστήριξη ως προς την C++.
 - Παράλληλη Είσοδο/Εξοδο.
 - Συνεργασία διεργασιών δημιουργημένων σε διαφορετικές γλώσσες π.χ. C, Fortran κτλ.



Συναρτήσεις MPI

- Τύποι κλήσεων που παρέχει το πρωτόκολλο:
 - Έναρξη, λήξη και διαχείριση επικοινωνίας.
- Επικοινωνία ανά ζεύγη διεργασιών (σημείο-σε-σημείο, point-to-point, blocking, non-blocking):
 - Αποστολή δεδομένων.
 - Παραλαβή δεδομένων.
- Συλλογική επικοινωνία διεργασιών (collective: broadcasting, scattering, gather, reduce):
 - Μαζική αποστολή δεδομένων.
 - Μαζική παραλαβή δεδομένων.
- Υπολογιστικές κλήσεις:
 - Reduction operations.



Οι 6 βασικές Κλήσεις

- `MPI_INIT` : Αρχικοποίηση MPI session.
- `MPI_COMM_SIZE` : Αριθμός διεργασιών.
- `MPI_COMM_RANK` : Ταυτότητα διεργασιών.
- `MPI_SEND` : Αποστολή δεδομένων (1 προς 1).
- `MPI_RECV` : Παραλαβή δεδομένων (1 προς 1).
- `MPI_FINALIZE` : Λήξη MPI session.



Πρότυπα συναρτήσεων

MPI_Init

Το πρότυπό της έχει την μορφή:

- `int MPI_Init (int *argc, char **argv);`
- Αρχικοποίηση του MPI – καλείται μόνο 1 φορά στην αρχή.
- Παράδειγμα:
 - **`MPI_Init(&argc, &argv);`**



Πρότυπα συναρτήσεων

MPI_Finalize

Το πρότυπό της έχει την μορφή:

- `int MPI_Finalize (void);`
- Τερματισμός του MPI – καλείται μόνο 1 φορά στο τέλος.
- Παράδειγμα:
 - **`MPI_Finalize();`**



Πρότυπα συναρτήσεων

MPI_Comm_rank

- `int MPI_Comm_rank (MPI_Comm comm, int * rank);`
- Η συνάρτηση `MPI_Comm_rank` επιστρέφει την τάξη της τρέχουσας διεργασίας.
- Η μεταβλητή `comm` χρησιμοποιείται για τον καθορισμό του communicator όπου ανήκει η διεργασία:
 - `communicator` (χειριστής): αντιπροσωπεύει μια ομάδα εργασιών και είναι το μέσο επικοινωνίας των διεργασιών αυτών,
 - `rank`: ακέραιος που περιέχει την τάξη της τρέχουσας διεργασίας.
- Παράδειγμα:
 - **`MPI_Comm_rank(MPI_COMM_WORLD, &rank);`**



Πρότυπα συναρτήσεων

MPI_Comm_size

- `int MPI_Comm_size (MPI_Comm comm, int * size);`
- Η συνάρτηση `MPI_Comm_size` επιστρέφει το μέγεθος του communicator.
 - Η μεταβλητή `comm` χρησιμοποιείται για τον καθορισμό του communicator όπου ανήκει η διεργασία.
 - `size`: ακέραιος που περιέχει το πλήθος των διεργασιών που περιλαμβάνονται στον communicator.
- Παράδειγμα:
 - `MPI_Comm_size (MPI_COMM_WORLD, &size);`



Πρότυπα συναρτήσεων

MPI_Get_processor_name

- `int MPI_Get_processor_name (char * name, int * len);`
- Επιστρέφει το όνομα του επεξεργαστή (`name`) στον οποίο στον οποίο τρέχει η διεργασία, και το μήκος του string σε αριθμό (`len`) χαρακτήρων
- Παράδειγμα:
 - `MPI_Get_processor_name(name, &len);`
- Απαιτείται δήλωση της μορφής:
 - `int len;`
 - `char name[MPI_MAX_PROCESSOR_NAME];`



Παράδειγμα στη C: hello world και mpi

```
#include <stdio.h>
#include <mpi.h>
int main(int argc, char
*argv[])
{
MPI_Init(&argc, &argv);
printf("Hello world!\n");
MPI_Finalize();
return 0;
}
```

ΠΡΟΣΟΧΗ: Σε κάθε πρόγραμμα πρέπει πάντα να υπάρχουν:

- include
- MPI_Init
- MPI_Finalize



Παράδειγμα στη Fortran: hello world και mpi

```
program main
include 'mpif.h'
integer ierr
call MPI_INIT(ierr)
print *, 'Hello world' print *, 'Hello world'
call MPI_FINALIZE(ierr)
end
```

- Μεταγλωττιστής (mpi77):
\$ mpif77 hello.f -o hello



Εκτέλεση σε επιλεγμένους κόμβους (1/3)

- Προϋποθέσεις:
 - Πρέπει να υπάρχει το ίδιο εκτελέσιμο.
 - Το εκτελέσιμο να μπορεί να εκτελεστεί στον κόμβο επεξεργασίας.
 - Πρέπει να βρίσκεται στην ίδια διαδρομή.



Εκτέλεση σε επιλεγμένους κόμβους (2/3)

- Πρέπει να κατασκευαστεί ένα αρχείο στο οποίο θα έχει γραμμές της μορφής:
 - `node_name slots=M max_slots=N`
 - όπου:
 - M, N ακέραιοι αριθμοί.
 - `node_name`, το FQDN όνομα των άλλων κόμβων.
 - `slots` ο αριθμός των επεξεργαστικών πυρήνων.

==>Για κάθε κόμβο της συστοιχίας θα υπάρχει μια γραμμή όπως παραπάνω.



Εκτέλεση σε επιλεγμένους κόμβους (3/3)

Παράδειγμα hostfile:

This is an example hostfile. Comments begin with

#

#

The following node is a single processor machine:

foo.example.com

The following node is a dual-processor machine:

bar.example.com slots=2

**# The following node is a quad-processor machine, and we
absolutely**

want to disallow over-subscribing it:

yow.example.com slots=4 max-slots=4



Δομικές μονάδες του MPI

- Στοιχειώδης μονάδα: διεργασία (process).
- Κάθε διεργασία έχει το δικό της id (process id).
- Τάξη: rank με τιμές από 0 έως $N - 1$, Όπου N το πλήθος των διεργασιών.
- Σύνολο των N διεργασιών: Ομάδα (group) με δικό της group id.
- Μέσο επικοινωνίας: communicator – επιτρέπει την επικοινωνία των διεργασιών.



Παραδείγματα σε Fortran και C++

```
PROGRAM hello
  INCLUDE 'mpif.h'
  INTEGER err
  CALL MPI_INIT(err)
  CALL MPI_COMM_RANK(MPI_COMM_WORLD,rank,ierr)
  CALL MPI_COMM_SIZE(MPI_COMM_WORLD,size,ierr)
  print *, 'I am ', rank, ' of ', size
  CALL MPI_FINALIZE(err)
END
```

```
#include <iostream>
#include <mpi.h>

using namespace std;

int main(int argc, char* argv[])
{
  int rank,size;
  MPI_Init( &argc, &argv );
  MPI_Comm_rank( MPI_COMM_WORLD, &rank );
  MPI_Comm_size( MPI_COMM_WORLD, &size );
  cout << "I am process " << rank
        << " out of " << size << endl;
  MPI_Finalize();
  return 0;
}
```



Βασικά στοιχεία

- Κάθε πρόγραμμα MPI περιλαμβάνει κλήσεις:
 - MPI_Init και MPI_Finalize.
- MPI_COMM_WORLD είναι το όνομα του συνόλου των διεργασιών στο MPI περιβάλλον.
- Κάθε διεργασία εκτελεί τις εντολές ανεξάρτητα από τις υπόλοιπες διεργασίες (π.χ. cout).
- Η σειρά με την οποία θα εκτελεστούν αυτές οι εντολές δεν είναι προκαθορισμένη.



Μεταγλώττιση και εκτέλεση

- Κάθε βιβλιοθήκη παρέχει wrapper scripts που αναλαμβάνουν την μεταγλώττιση (compilation) του κώδικα:
 - mpif77.
 - mpif90.
 - mpicc.
 - mpicxx ή mpiCC.
- Για την εκτέλεση MPI προγραμμάτων οι βιβλιοθήκες παρέχουν κατάλληλες εφαρμογές:
 - mpiexec.
 - mpirun.



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

