Παράλληλη Επεξεργασία

Φροντιστήριο:

- *Προετοιμασία για προγραμματισμό στην Παράλληλη Επεξεργασία*
- *Τεχνικές και διαθέσιμα εργαλεία*

Εργαστήριο Πληροφοριακών Συστημάτων Υψηλής Επίδοσης

Parallel and Distributed Systems Group

Κατάλληλο υλικό

- 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 Linux
  (Προτεινόμενες διανομές: Fedora, CentOS, Ubuntu).
- Οι τελικές μετρήσεις και τα πειράματα θα διενεργηθούν σε συστήματα του τμήματος με Linux.
- Ωστόσο για την σταδιακή υλοποίηση των εργαστηριακών ασκήσεων παρέχεται μερική υποστήριξη για ανάπτυξη σε
  Windows, MacOSX
- Εργαλεία για την διευκολύνση της εγκατάστασης Linux και της ανάπτυξης στο σπίτι:
  - Virtualization μέσω του VMware Player (free)

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)

Πολυνηματικός Προγραμματισμός

- 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

- 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

Παράλληλος Προγραμματισμός σε Κοινή Μνήμη

- Χρήση του προτύπου OpenMP (Open Multi-Processing).
- Ορίζεται για τις γλώσσες C/C++ και Fortran.
- Από την έκδοση 2.5 και έπειτα η προγραμματιστική διεπαφή είναι κοινή.

- Χαρακτηριστικά του OpenMP:
  - Πολυνηματισμός σε πιο υψηλό και αφαιρετικό επίπεδο.
  - Απλό σύνολο από οδηγίες προς τον μεταφραστή και το σύστημα χρόνου εκτέλεσης (run-time system).
  - Ευνοείται η σταδιακή παραλληλοποιήση καθώς και η παραλληλοποίηση σε συγκεκριμένα μόνο τμήματα.
  - Διευρύνεται η υποστήριξη από τους μεταφραστές.
  - Απαλλαγή του προγραμματιστή από δυσκολίες στον συντονισμό του πολυνηματισμού.

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.

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

- 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.

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 πουπαρέχονται από τον μεταφραστή (παράμετροι -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)
- 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)

- Τα αρχεία που θα προκύψουν από την παραπάνω διαδικασία θα αποτελέσουν μέρος των παραδοτέων των εργαστηριακών ασκήσεων

- 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.

GNU Make

Παράδειγμα 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.*
```

Λεπτομέρειες http://www.gnu.org/software/make/manual

Επισκόπηση προγραμμάτων – Debugging

- 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.

- Για τις ανάγκες του μαθήματος, οι προτεινόμενοι τρόποι διάγνωσης προβλημάτων και επισκόπησης της εκτέλεσης είναι:

  - Εκτύπωση κατάλληλων μηνημάτων 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.

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 …

Χρήση των Υπολογιστικών Συστημάτων του Τμήματος

- Μετά τον καθορισμό των ομάδων θα σας δοθεί πρόσβαση σε υπολογιστικά συστήματα του τμήματος 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.)
- 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 παρέχετε πρόσβαση σε γραφικό περιβάλλον
- Χρήσιμη εφαρμογή για την χρήση πολλαπλών τερματικών σε μια απομακρυσμένη σύνδεση και την εκτέλεση πειραμάτων στο background είναι η εφαρμογή screen (> man screen)

Παράδοση εργασιών και παρουσίαση αποτελεσμάτων

- Στα παραδοτέα των εργαστηριακών ασκήσεων συμπεριλαμβάνετε τον πηγαίο κώδικα καθώς και αρχεία που χρησιμεύουν στην μετάφραση/εκτέλεση του (makefiles,shell scripts κλπ) ούτως ώστε να διευκολύνεται ο έλεγχος της άσκησης σας.

- Για την παρουσίαση των αποτελεσμάτων σας μπορείτε να χρησιμοποιήσετε κάποιο πρόγραμμα γραφικών παραστάσεων, όπως τα gnuplot, MS Excel, κ.λ.π.

- Στην αναφορά μπορείτε να συμπεριλάβετε ορισμένα τμήματα του κώδικα σας που θεωρείτε απαραίτητο να σχολιαστούν. Δεν χρειάζεται η συμπερίληψη ολόκληρου του κώδικα σαν παράρτημα στην αναφορά.

- *Απορίες και συζήτηση μέσω του forum του μαθήματος στο My.CEID*
- *e-mail επικοινωνίας kik@hpclab.ceid.upatras.gr*
- *Ανακοίνωση ασκήσεων και υλικό στην ιστοσελίδα του μαθήματος http://parallel.hpclab.ceid.upatras.gr/*