Παράλληλη Επεξεργασία

Φροντιστήριο:

- *Εισαγωγή στο OpenMP*

Εργαστήριο Πληροφοριακών Συστημάτων Υψηλής Επίδοσης

Parallel and Distributed Systems Group

Τι είναι το OpenMP

- Πρότυπο – Επέκταση στη C/C++ και τη Fortran
  - Έκφραση παραλληλισμού για Η/Υ κοινής μνήμης
  - Μεταφέρσιμο, εφαρμόσιμο σε πολλές πλατφόρμες συμπεριλαμβανομένου του Unix και των Windows
  - Διαθέσιμες υλοποιήσεις για C/C++ και Fortran
  - Έχει δημιουργηθεί και υποστηρίζεται από πλήθος σημαντικών κατασκευαστών υλικού και λογισμικού
  - Υποστηρίζει λεπτό και αδρό καταμερισμό του παραλληλισμού (fine/coarse – grained parallelism)
- Το πρότυπο OpenMP ορίζει τρία βασικά στοιχεία
  - Οδηγίες προς το μεταγλωττιστή (compiler directives)
  - Σύνολο συναρτήσεων χρόνου εκτέλεσης (run-time library functions)
  - Μεταβλητές περιβάλλοντος (environment variables)

Στόχοι του OpenMP

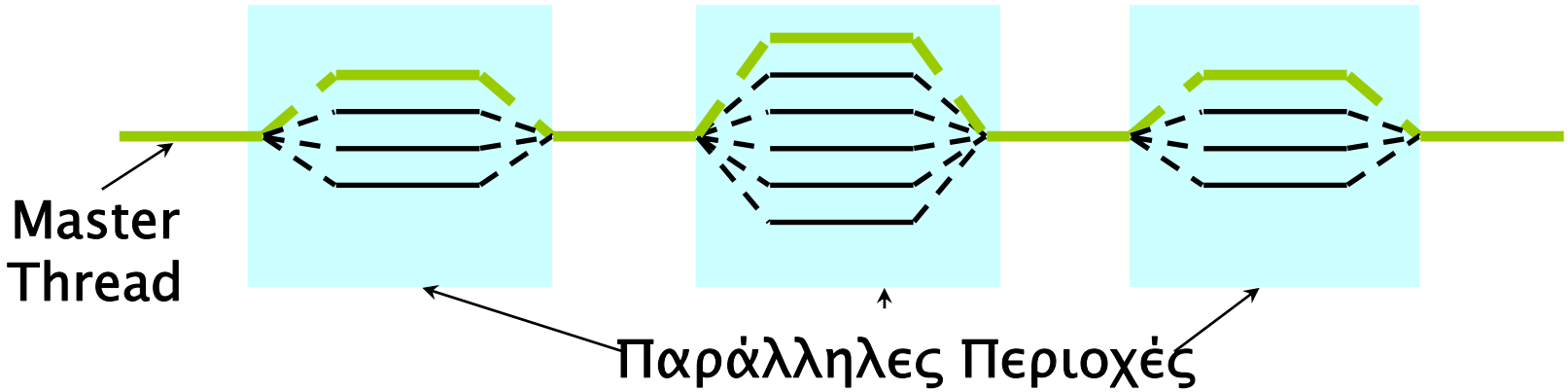
- Ύπαρξη ενός κοινού προτύπου για πλειάδα αρχιτεκτονικών κοινής μνήμης
- Καθορισμός ενός απλού και περιορισμένου συνόλου οδηγιών για των προγραμματισμό παραλλήλων μηχανών κοινής μνήμης
  - Γενικά μπορούμε να πετύχουμε την παραλληλοποίηση χρησιμοποιώντας ένα σύνολο 3-4 κοινών οδηγιών
- Δυνατότητα «σταδιακής» παραλληλοποίησης, αντίθετα με τις βιβλιοθήκες περάσματος μηνυμάτων όπου ακολουθείται προσέγγιση καθολικού ή καθόλου παραλληλισμού
- Δυνατότητα υποστήριξης ταυτόχρονα λεπτά και αδρά καταμερισμένου παραλληλισμού
- Υποστήριξη Fortran (77, 90 και 95), C και C++

Προγραμματιστικό Μοντέλο

- Μοντέλο προγραμματισμού κοινής μνήμης
- Παραλληλισμός βασισμένος σε νήματα. Μια διεργασία κοινής μνήμης μπορεί να αποτελείται από ένα ή περισσότερα νήματα. Το OpenMP βασίζεται στη χρήση πολλών νημάτων τα οποία μοιράζονται τον ίδιο χώρο μνήμης
- Σαφώς ορισμένος παραλληλισμός : Το OpenMP δεν είναι μοντέλο αυτόματης παραλληλοποίησης. Προσφέρεται δηλαδή, σε μεγάλο βαθμό, έλεγχος στον προγραμματιστή

Προγραμματιστικό Μοντέλο

- **Fork – Join μοντέλο** :Τα OpenMP προγράμματα ξεκινούν σαν 1 διεργασία, το master thread. Το master thread εκτελεί ακολουθιακά μέχρι να συναντήσει το 1ο παράλληλο τμήμα
- **Fork** : Το master thread δημιουργεί μια ομάδα νημάτων
- Οι εντολές που εμπεριέχονται στο παράλληλο τμήμα εκτελούνται παράλληλα από τα πολλαπλά νήματα
- **Join** : Όταν τα νήματα της ομάδας ολοκληρώσουν τις εντολές του παράλληλου τμήματος συγχρονίζονται και τερματίζουν(*). Απομένει και πάλι μόνο το master thread



(*) Στην πραγματικότητα περιμένουν να επαναχρησιμοποιηθούν σε επόμενη παράλληλη περιοχή

Προγραμματιστικό Μοντέλο

- **Βασισμένο σε οδηγίες στο μεταγλωττιστή**:
  Σχεδόν όλη η παραλληλοποίηση γίνεται με οδηγίες προς τον μεταγλωττιστή οι οποίες ενσωματώνονται στο πρόγραμμα Fortran ή C

- **Υποστήριξη πολυεπίπεδου παραλληλισμού**:
  Το πρότυπο υποστηρίζει την τοποθέτηση παράλληλων τμημάτων μέσα σε άλλα παράλληλα τμήματα. Ωστόσο οι υλοποιήσεις δεν προσφέρουν πάντα αυτή τη λειτουργικότητα

- **Δυναμικά μεταβαλλόμενο πλήθος νημάτων**:
  Το πρότυπο υποστηρίζει τη δυναμική αλλαγή του πλήθους των νημάτων που εκτελούν τα παράλληλα τμήματα. Και αυτή η λειτουργικότητα παρέχεται προαιρετικά από τις υλοποιήσεις

Σύνταξη

- Οι περισσότερες οδηγίες στο OpenMP είναι οδηγίες προς τον compiler και η σύνταξη τους έχει την ακόλουθη μορφή:
  - C/C++

    #pragma omp *directive-name [clauses ...]*
  - Fortran

    C$OMP *directive-name [clauses ...]*
    !$OMP *directive-name [clauses...]*
    *$OMP *directive-name [clauses...]*
- Επομένως είναι δυνατή η αγνόηση τους από τον compiler

Δομημένα Τμήματα Κώδικα

- Οι περισσότερες οδηγίες στο OpenMP εφαρμόζονται σε δομημένα τμήματα κώδικα (structured blocks)

- Αυτό σημαίνει πως τα τμήματα που περικλείονται σε μία οδηγία πρέπει να έχουν ένα μοναδικό σημείο εισόδου στην αρχή τους και ένα μοναδικό σημείο εξόδου στο τέλος τους.

Κατηγορίες Οδηγιών

- Κάθε οδηγία στο OpenMP ανήκει σε μια από τις ακόλουθες 5 κατηγορίες
  - Παράλληλες Περιοχές (Parallel Regions)
  - Διαχείριση Εμβέλειας Δεδομένων
  - Διαμοίραση Έργου (Work-sharing)
  - Συγχρονισμός
  - Συναρτήσεις χρόνου εκτέλεσης/ μεταβλητές περιβάλλοντος

Παράλληλες Περιοχές

- Καθορίζονται από την οδηγία parallel
- Είναι η πιο βασική οδηγία στο OpenMP
- Όταν ένα νήμα (master thread) συναντήσει την οδηγία parallel δημιουργεί επιπλέον νήματα που θα εκτελεστούν παράλληλα.
- Κάθε νήμα συμπεριλαμβανομένου και του master εκτελεί για λογαριασμό του τον κώδικα που περιλαμβάνει η οδηγία parallel
- Στο τέλος της οδηγίας υπονοείται από την υλοποίηση ένα barrier πέραν του οποίου συνεχίζει την εκτέλεση του μόνο το master thread

Hello World!

```
#include <stdio.h>
#include <omp.h>

int main(int argc, char **argv)
{
    int id;

#pragma omp parallel private(id)
      {
          id = omp_get_thread_num();
          printf("Hello World from %d\n", id);
      }

   fprintf(stderr, "Program terminated\n");
   return 0;
}
```

Εμβέλεια Δεδομένων

- Εξ ορισμού οι περισσότερες μεταβλητές είναι κοινές σε μια παράλληλη περιοχή (shared memory model)

- Εξαιρέσεις:
  - Μεταβλητές που αποθηκεύονται στη stack κάθε νήματος (ορίσματα ή τοπικές μεταβλητές σε υπο-ρουτίνες που καλούνται μέσα από το παράλληλο τμήμα)
  - Αυτόματες μεταβλητές μέσα στο παράλληλο τμήμα
  - Index μεταβλητές στα παράλληλα loops

Καθορισμός Εμβέλειας

- Αλλαγή της εμβέλειας μιας μεταβλητής μπορεί να γίνει μέσω των προτάσεων:
  - SHARED (υποδεικνύεται η διαμοίραση της μεταβλητής)
  - PRIVATE (υποδεικνύεται η δημιουργία ιδιωτικού αντιγράφου)
  - FIRSTPRIVATE (επιπλέον αρχικοποίηση με την τιμή πριν την **parallel**)
  - LASTPRIVATE (επιπλέον ανάθεση της τελικής τιμής όπως αυτή προβλέπεται από την ακολουθιακή εκτέλεση)

Παραδείγματα Εμβέλειας

```
      program wrong
      INTEGER X
      X = 0
C$OMP PARALLEL DO PRIVATE(X)
      DO J=1,1000
        X= X + 1
      ENDDO
C$OMP END PARALLEL DO
      PRINT *, X
```

```
      program almost_right
      X = 0
C$OMP PARALLEL DO FIRSTPRIVATE(X)
      DO J=1,1000
        X = X + 1
      ENDDO
C$OMP END PARALLEL DO
PRINT *, X
```

```
      program closer
      X = 0
C$OMP PARALLEL DO FIRSTPRIVATE(X)
C$OMP+ LASTPRIVATE(X)
      DO J=1,1000
        X = X + 1
      ENDDO
C$OMP END PARALLEL DO
      PRINT *, X
```

Διαθέσιμο Υλικό

- Current state of the art
  Έκδοση πρωτοκόλλου 3.0 (ενοποιημένη για C/C++ και Fortran)

- Πληροφορίες και διαθέσιμο υλικό
  http://openmp.org/

  Κατεβάστε την περιγραφή (specification) για το API.
  Βρείτε λύσεις μέσω του forum
  Τσεκάρετε tutorials και παραδείγματα

- Βιβλίο:
  Using OpenMP, by Chapman, Jost, and Van Der Pas

- *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/*