Concurrent Programming

Session 4: Parallel Programming Styles

Computer Engineering Department
Iran University of Science and Technology
Tehran, Iran

Instructor: Hadi Salimi
Distributed Systems Lab.
Computer Engineering Department,
Iran University of Science and Technology,
hsalimi@iust.ac.ir

Concurrent Programming Paradigms

- Iterative Parallelism
- Recursive Parallelism
- Producers and Consumers (Pipeline)
- Client and Servers
- Interacting Peers

Iterative Parallelism

- An iterative parallel program contains two or more iterative processes.
- Each process computes results for a subset of the data, then the results are combined.

Matrix Multiplication

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

for [i = 0 to n-1] {
  for [j = 0 to n-1] {
    # compute inner product of a[i,*] and 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];
  }
}
```

An embarrassingly parallel application.

Parallelism Condition

- Two operations can be executed in parallel if they are *independent.*
- Two operations are independent if their write sets are disjoint.

Parallel Version

```
co [i = 0 to n-1] {   # compute rows in parallel
  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];
  }
}


co [j = 0 to n-1] {   # compute columns in parallel
  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];
  }
}
```

Another Version

```
co [i = 0 to n-1, j = 0 to n-1] { # all rows and
  c[i,j] = 0.0;                   # all columns
  for [k = 0 to n-1]
    c[i,j] = c[i,j] + a[i,k]*b[k,j];
}
```

Recursive Parallelism

- Some recursive calls can be done recursively if:
  - The procedure does not reference global variables or only reads them
  - Reference and result variables, if any, are distinct

Iterative Version

```
double fleft = f(a), fright, area = 0.0;
double width = (b-a) / INTERVALS;
for [x = (a + width) to b by width] {
  fright = f(x);
  area = area + (fleft + fright) * width / 2;
  fleft = fright;
}
```

Is this program parallelizable?

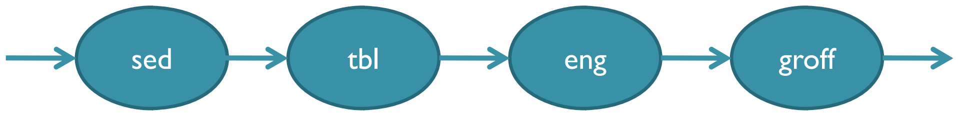
Recursive Version

```
double quad(double left,right,fleft,fright,lrarea) {
  double mid = (left + right) / 2;
  double fmid = f(mid);
  double larea = (fleft+fmid) * (mid-left) / 2;
  double rarea = (fmid+fright) * (right-mid) / 2;
  if (abs((larea+rarea) - lrarea) > EPSILON) {
    # recurse to integrate both halves
    larea = quad(left, mid, fleft, fmid, larea);
    rarea = quad(mid, right, fmid, fright, rarea);
  }
  return (larea + rarea);
}
```

Producers and Consumers

- Consumer processes
- Producer processes
- Unix pipes

Sed –f script $* tbl | eng | groff Maros -



Client and Servers

- Web Servers
- File Servers
- Database Management Servers

Interacting Peers

- Being client or server is just a rule
- In a peer-to-peer system each node may be client or server.
- There may be a data flow among peers.