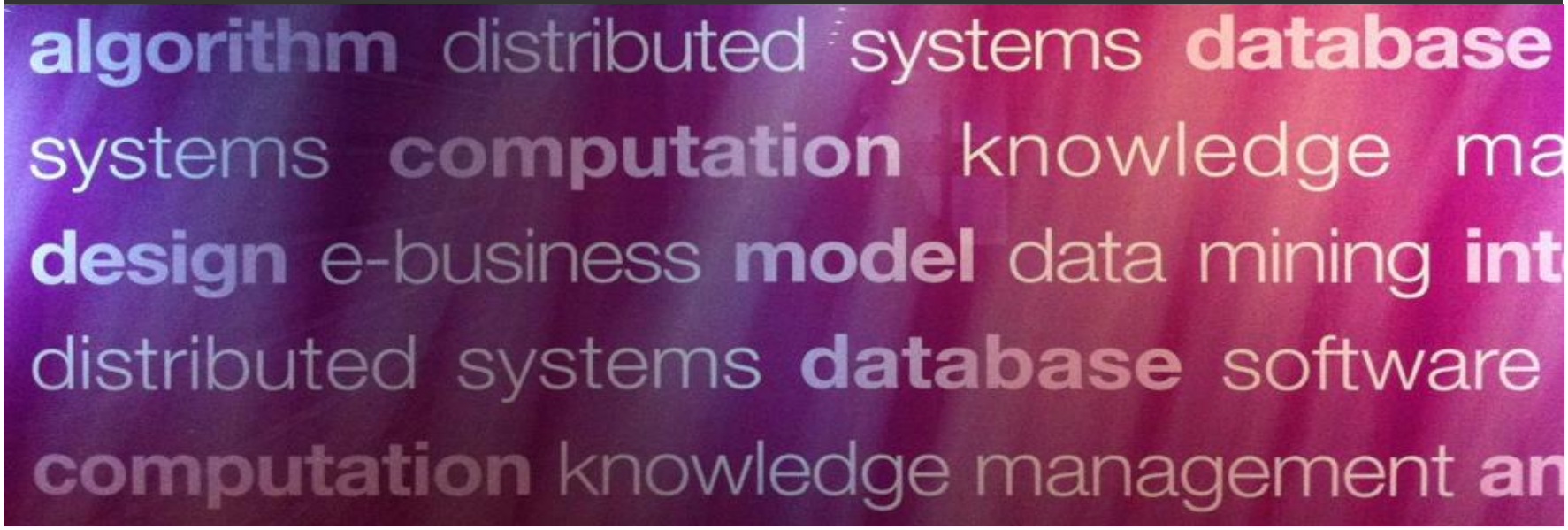
MONASH University

Information Technology

FIT3143 - LECTURE WEEK 7b

PARALLEL ALGORITHM DESIGN - SIMPLE PARTITIONING



Overview

- Example of a parallel partitioning algorithm for numerical computation.
- Note: This lecture include hands-on coding demonstration. Hence, although the content of the notes may be simple, additional code and drawing resources are enclosed with this lecture notes in Moodle.

Learning outcome(s) related to this topic

- Design and develop parallel algorithms for various parallel computing architectures (LO3)

Numerical Integration

- A general divide-and-conquer technique divides the region continually into parts and lets some optimization function decide when certain regions are sufficiently divided. Let us take an example, i.e., numerical integration.

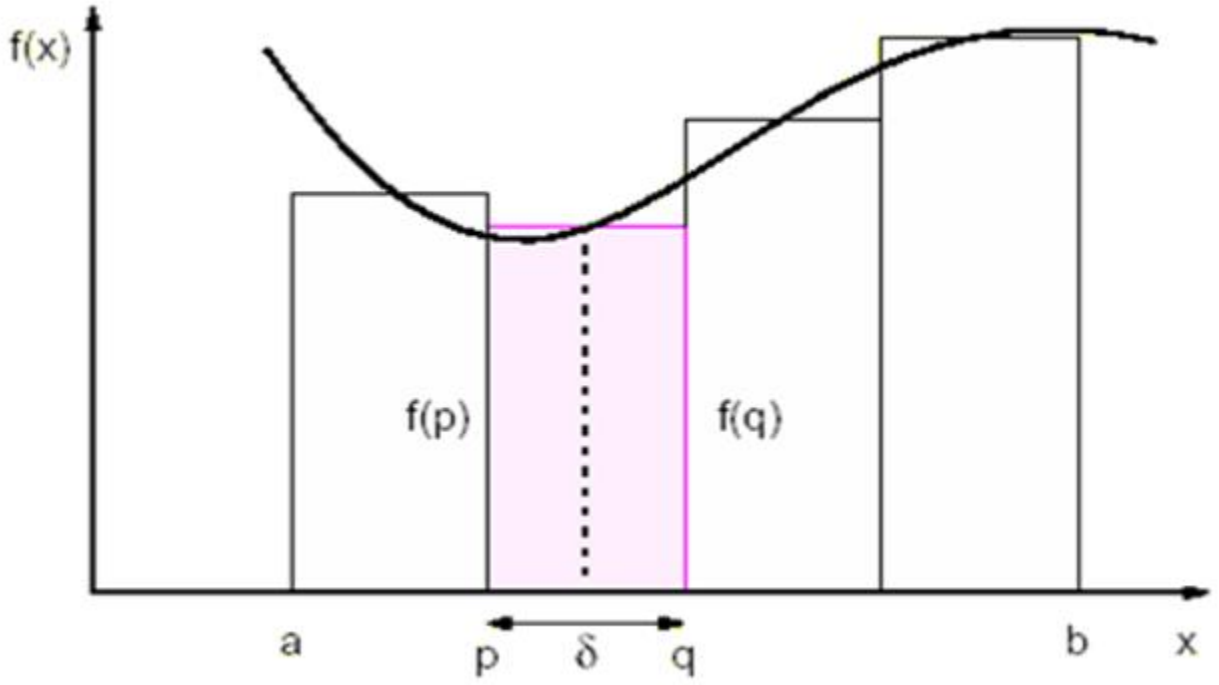
Numerical Integration

$$I = \int_a^b f(x)dx$$

- To integrate the function (i.e., to compute the 'area under the curve'), we can divide the area into separate parts,each of which can be calculated by a separate process.
- Each region could be calculated using an approximation given by rectangle, where f(p) and f(q) are the heights of the two edges of the rectangular region and ∂ is the width (interval).
- The complete integration can be approximated by the summation of rectangular regions from a to b by aligning the rectangles so that the upper midpoint each rectangle intersect with the function. This has the advantage that errors on each side of midpoint end tend to cancel.

Numerical Integration (Rectangles)

- Each region under the curve is calculates using an approximation given by rectangles which are aligned.

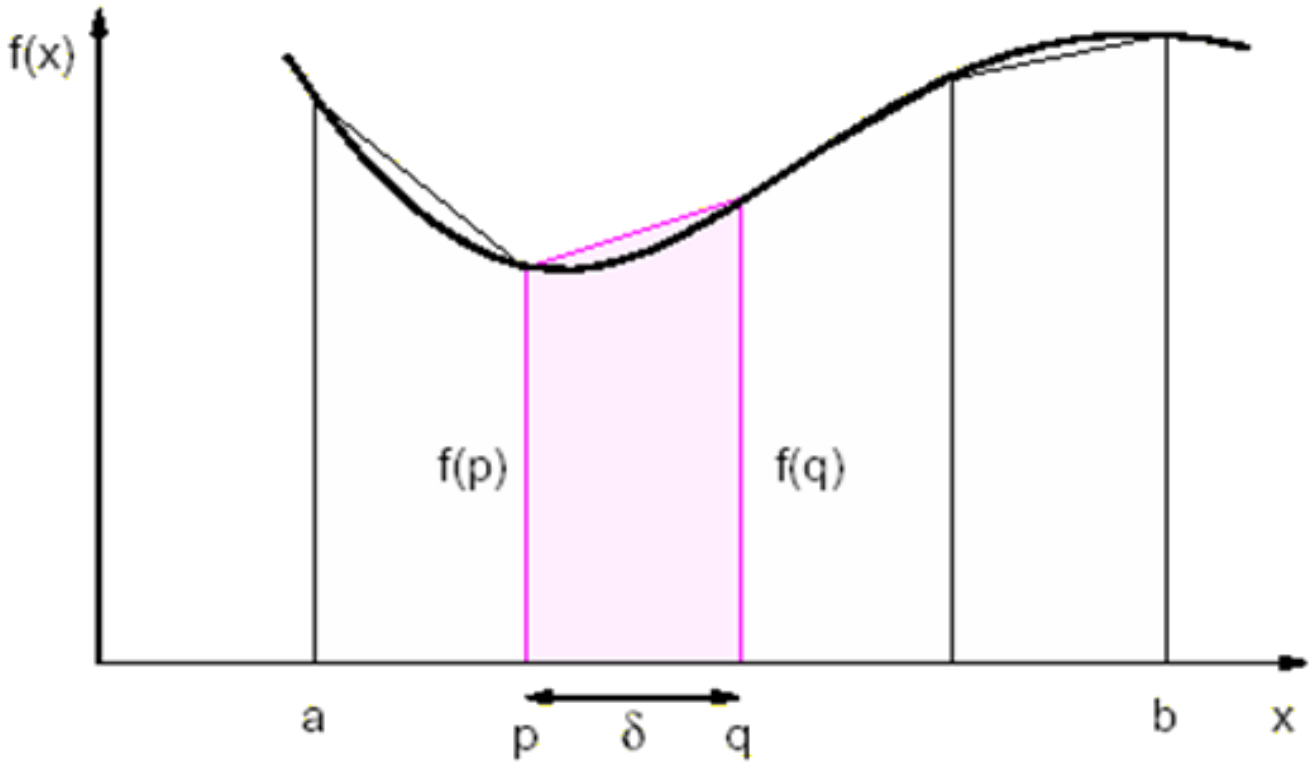


Numerical Integration (Trapezoid)

- Here we take the actual intersections of the lines with the function to create trapezoidal regions as shown in the fig.
- Each region is now calculated as 1/2(f(p) +f (q))∂
- Such approximate numerical methods for computing a definite integral using linear combination of values are called ***quadrature methods***

Numerical Integration (Trapezoid)

Area = ½ (f(p) +f (q)) ∂



Numerical Integration (Trapezoid)

- Let us consider a trapezoidal method.
- Prior to start of the computation, one process is statically assigned to be responsible for computing each region. By making the interval smaller, we come closer to attaining the exact solution.
- Since each calculation is of the same form, the single process multiple data (SPMD) model is appropriate (i.e., data parallelism).
- Suppose we were to sum the area from x=a to x=b using p processes, numbered 0 to p-1.The size of the region for each process is (b-a)/p. To calculate the area in the described manner, a section of SPMD pseudo-code could be as follows

Numerical Integration – Trapezoid (Basic pseudocode)

```
Process P_i
if (i == master) {                          /* read number of intervals required */
        printf("Enter number of intervals ");
        scanf(%d",&n);
}
bcast(&n, P_group);                         /* broadcast interval to all processes */
region = (b - a)/p;                 /* length of region for each process */
start = a + region * i;                     /* starting x coordinate for process */
end = start + region;                       /* ending x coordinate for process */
d = (b - a)/n;                              /* size of interval */
area = 0.0;
for (x = start; x < end; x = x + d)
        area = area + f(x) + f(x+d);
area = 0.5 * area * d;
reduce_add(&integral, &area, P_group);    /* form sum of areas */
```

A reduce operation is used to add the areas computed by the individual processes.

Numerical Integration – Trapezoid (Basic pseudocode)

- Please refer to the enclosed drawings and code with this lecture notes for further explanation on partitioning and parallel code for numerical integration.

References

Wilkinson .B, Allen .M, “Chapter 4: Partitioning and Divide-And-Conquer Strategies”, in *Parallel Programming – Techniques and Applications Using Networked Workstations and Parallel Computers*, Pearson Education Inc., 2005, pp. 122 – 126.