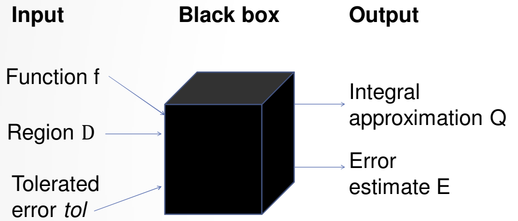
Adaptive Task Partitioning in *ParInt*

Omofolakunmi Olagbemi (with Dr. Elise de Doncker)
College of Engineering and Applied Sciences, Western Michigan University, Kalamazoo

Introduction

ParInt is a software for parallel multivariate integration, and was developed by a group of Computer Science (CS) Faculty members and PhD students led by Dr. Elise de Doncker. The project was funded by the National Science Foundation (NSF) through a series of grants.

Problem definition



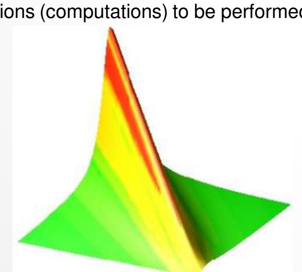
Compute: $Q \approx I = \int_D f(x)\,dx, \;\; E \approx |Q - I|$

such that

$$|Q - I| \leq E \leq tol$$

ParInt (represented by the black box in the diagram above) includes an **adaptive algorithm** in integrating functions specified by the user, and repeatedly evaluates the results till the user's requested accuracy is attained (that is, tolerated error is not exceeded) or till it has reached or exceeded a pre-determined limit on number of evaluations (computations) to be performed.

This diagram shows a plot of the integrand function below for ∈ = 0.1



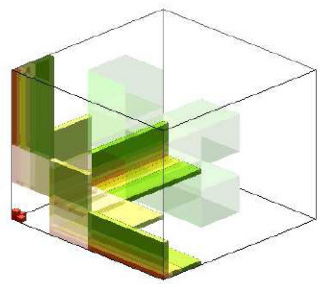
$$\int_0^1 \int_0^1 dxdy \frac{2 \in y}{(x + y - 1)^2 + \in^2}$$

Adaptive integration meta-algorithm

Evaluate initial region and update results
Initialize priority queue with initial region
while (evaluation limit not reached and estimated error too large)
  Retrieve region from priority queue
  Split region
  Evaluate new sub-region and update results
  Insert new regions into priority queue

We consider the function 1 below:

$$f(x,y,z) = x^{-0.2} y^{-0.2} z^{-0.2} (x+y+z)^{-0.2} \qquad (1)$$



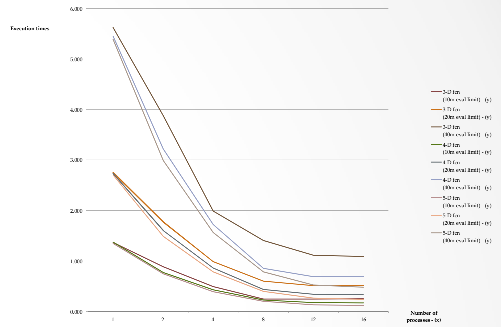
Snapshot of computations (sub-region evaluations) for the function above

The adaptive integration algorithm outlined above makes use of a user-specified number of processes in evaluating the sub-regions of region D. The given region (D) is partitioned among the processes. Each process performs a sequence of subdivision steps as follows:

- it selects the next sub-region to subdivide (the regions with the highest local error estimates are selected for subdivision – these regions are colored red in the diagram above),
- it subdivides and evaluates the selected sub-region,
- it sends intermittent messages to the 'controller processor' reporting changes in its integral (Q) and changes in the error estimate (E).

Some processors initially receive difficult portions of the integration region and so spend more time on their evaluations while other processors may become idle. There are various methods used to balance the load for greater efficiency. In *ParInt*, a scheduler-based load balancing method is used. In this method, the controller processor maintains a list of idle and non-idle processors. Each update message sent to this processor contains information about the idle status of the sending processor. The controller then directs an idle processor to request work from a non-idle one.

Results



The chart above shows the execution times when computing the integrand for function (1) as well as its 4- and 5-dimensional versions. Evaluation limits of 10million, 20million and 40million were used. These were executed on the clusters in the High Performance Computational Sciences (HPCS) Laboratory in the Department of Computer Science (CEAS). From the results above, using an increasingly higher number of processes resulted in progressively lower execution times, until excessive overheads prevent further decrease of execution time.

Applications

1. Statistical analysis of finite elements (for example, automotive simulations),
2. Computational finance (financial derivatives, for example, collateral mortgage obligations, mortgage backed security problems),
3. High energy physics (modeling of particle interactions),
4. Computational chemistry (for representing the electronic structure of an atom or molecule),
5. Computational geometry (tessellations),
6. Biometrics (for example, for analysis of taste-testing trials)
7. Computation of radiation dose for cancer treatment.

Future work

1. Extensions for a hybrid environment (with distributed computing, multi-core, many-core on Graphical Processing Units - GPUs).
2. Extensions with respect to numerical methods.
3. Extensions with respect to parallel strategies and load balancing using hybrid architectures.

References

1. Shujun Li; *Online Support for Multivariate Integration;* PhD Thesis; Western Michigan University; 2005.
2. E. de Doncker, K. Kaugars, L. Cucos, R. Zanny; *Current Status of the ParInt package for Parallel multivariate Integration;* In Proc. Computational Particle Physics Symposium (CPP01), 110 – 119; 2002.