A Mathematical Analysis of the Scattered Decomposition

John Salmon

Mail Code 356-48, California Institute of Technology, Pasadena, CA 91125

ABSTRACT

A theoretical basis for the scattered decomposition is worked out in some detail. The basic result has been part of the "folklore" for some time, but has never been proved. The load imbalance expected from a scattered decomposition of a set of computational tasks is proportional $n^{-\frac{1}{2}}\frac{\sigma_{task}}{m_{task}}$, where n is the number of tasks assigned to each processor, m_{task} is the mean time per task and σ_{task} is the root mean square deviation time per task. The constant of proportionality is of O(1), and is a very slowly increasing function of the number of processors.

The scattered decomposition is part of the standard "toolkit" of the parallel programmer [Fox 88]. When confronted with an irregular problem, or one in which correlations exist which would cause a regular decomposition to suffer from load imbalance, the scattered decomposition often provides a solution. In essence, the scattered decomposition takes the computational tasks, and distributes them over the machine without regard to their proximity in some underlying space. The goal is to destroy any correlations that might exist between tasks that are assigned to an individual processor. The question addressed here is just how well does this procedure work?

Consider the example of generating a computer graphics image by ray-tracing [Goldsmith 88]. A tiled decomposition of an image would assign rectangular blocks of pixels to individual processors. This decomposition can lead to significant load imbalance because pixels (and hence blocks of pixels) near the periphery of the image will often be much easier to compute than their counterparts near the center of the image. With a scattered decomposition, each processor is assigned pixels from all parts of the image, and this source of load imbalance is eliminated. Nevertheless, without prior knowledge of exactly which pixels are the most time consuming, the scattered decomposition still suffers from some statistical load imbalance. The magnitude of this residual load imbalance is estimated below.

First, define the load imbalance as

$$l = \frac{t_{slowest} - m_{proc}}{m_{proc}}$$

where m_{proc} is the mean time for a processor to complete its tasks, and consider the load imbalance one can expect from an ensemble of N_{proc} such processors. Let the time for each processor to finish be a random variable with distribution P_{proc} and assume that the processors are independent. Under these conditions, l is also a random variable. It's distribution, denoted by P_{imbal} is obtained by noting that for the load imbalance to be less than l, each of N_{proc} independent processors must finish in time less than $(l+1)m_{proc}$. Thus

$$P_{imbal}(imbal < l) = [P_{proc}(t_{finish} < (l+1)m_{proc})]^{N_{proc}}$$
$$= [1-P_{proc}(t_{finish} > (l+1)m_{proc})]^{N_{proc}}$$
$$\approx 1 - N_{proc}P_{proc}(t_{finish} > (l+1)m_{proc})$$

or

$$P_{imbal}(imbal > l) \approx N_{proc}P_{proc}(t_{finish} > (l+1)m_{proc})$$

The approximation is good as long as the result is small.

Now suppose that the time taken by each processor is the sum of n sub-tasks, and that the time to complete each sub-task is also a random variable with distribution $P_{task}(t)$. The essential feature of the scattered decomposition is to make the sub-tasks assigned to a given processor independent, so that lengthy tasks are not assigned disproportionately to a single processor. If n is large and the distribution P_{task} has finite variance, then the Central Limit Theorem [Lindgren] states that the distribution $P_{proc}(t)$ approaches a Normal Distribution, regardless of the details of $P_{task}(t)$.

$$P_{proc}(t_{finish} < t) \approx \int_{-\infty}^{x} \exp(-(\Delta t)^2/\sigma_{proc}^2)$$

$$= \frac{1}{2}(1+erf(\frac{\Delta t}{\sqrt{2}\sigma_{proc}}))$$

$$P_{proc}(t_{finish} > t) \approx \frac{1}{2}\, erfc(\frac{\Delta t}{\sqrt{2}\sigma_{proc}})$$

where

$$m_{proc} = nt_{task},\ \ \sigma_{proc} = n^{1/2}\sigma_{task},\ \ \Delta t = (t - m_{proc}),$$

and $erf(x)$ and $erfc(x)$ are the error function and the complimentary error function, respectively. When the complimentary error function is small, its value may be approximated by [Abramowitz]:

$$erfc(x) \approx \frac{1}{x\sqrt{\pi}}\exp(-x^2)$$

We are now in a position to estimate $P_{imbal}(imbal > l)$ in terms of m_{task} and σ_{task}. Simple manipulation of the above expressions reveals that:

$$P_{imbal}(imbal > l) \approx \frac{N_{proc}}{\sqrt{2\pi n}}\frac{\sigma_{task}}{l m_{task}}\exp(-\frac{nl^2 m_{task}^2}{2\sigma_{task}^2})$$

The load imbalance l_c corresponding to a a confidence of c is defined to be the value of load imbalance which is exceeded with probability $1-c$. That is:

$$1 - c = P_{imbal}(imbal > l_c)$$

Another simple calculation reveals that

$$l_c \approx \frac{\sigma_{task}}{n^{1/2} m_{task}}\sqrt{2\ln\left[\frac{N_{proc}}{1-c}\frac{\sigma_{task}}{\sqrt{2\pi n}\, l m_{task}}\right]}$$

$$= \frac{\sigma_{task}}{n^{1/2} m_{task}}\sqrt{\ln\left[\frac{N_{proc}^2}{2\pi(1-c)^2}\right]}$$

The constant of proportionality is an extremely slowly growing function of N_{proc}. In the rather extreme case of N_{proc}=1000 and a confidence of 99%, we still have:

$$l_{99\%} \approx 4.6\frac{\sigma_{task}}{n^{1/2} m_{task}}$$

Acknowledgments

This work was supported in part by Department of Energy Grant No. DE–FG03–85ER25009, the Program Manager of the Joint Tactical Fusion Office and the ESD division of the USAF as well as grants from IBM and SANDIA and a Shell Foundation Fellowship. It is my pleasure to acknowledge several illuminating conversations with Jeff Goldsmith, Roy Williams, Mark Muldoon and Jon Flower.

References

Abramowitz, M., Stegun, I., *Handbook of Mathematical Functions*, Dover Publications, New York, 1972

Lindgren, B., *Statistical Theory*, Macmillian, New York, 1976

Fox, G., Johnson, M., Lyzenga, G., Otto, S., Salmon, J., Walker, D., *Solving Problems on Concurrent Processors*, Prentice-Hall, Englewood Cliffs N.J., 1988.

Goldsmith, J., Salmon, J., A Hypercube Raytracer, *Proceedings of the Third Conference on Hypercube Computers and Applications*, ed. G.C. Fox, SIAM, Philadelphia, 1988