Mem. S.A.It. Suppl. Vol. 1, 221

Memorie della
Supplementi

A task of advanced computing: how to process large quantities of data?

F. Pasian

INAF-Osservatorio Astronomico di Trieste,Via Tiepolo n.11, 34131 Trieste, Italy
e-mail: `pasian@ts.astro.it`

Abstract. In the round table we have had the chance to widen the scope of the discussion to include "advanced computing" in astrophysics, i.e. the union of computation, data, networks, etc., in other words data-intensive science. Here I wish to stress that there is an additional topic which is worth mentioning in this forum: which kind of supercomputing is necessary to tackle the processing and analysis of large quantities of data? By discussing this technical item we may find general solutions which are applicable to theoreticians as well.

1. The difficult task of processing large quantities of astronomical data

There are two general cases which can be considered.

1. Processing of huge quantities of data (large detectors, mosaics, images with high time resolution) is typical of the optical and solar communities. Let's consider the example of an optical survey performed by LBC@LBT: in principle, the instrument will produce every 3 minutes for each of the 2 channels six CCD images, all of which need to be routinely corrected for defects, and astrometrically and photometrically calibrated before a mosaic of the four scientific frames is actually built. The amount of computation is impressive but, since based on local operators, "embarassigly parallel", and the level of parallelism is coarse-grained. This allows to subdivide data to fit the RAM available to the CPUs and have each CPU to perform a single processing unit (e.g. one of the calibration steps on a single chip of the mosaic).

2. In some other case, we may have a smaller quantity of data, but a higher level of computational complexity. A good example is Planck, which will implement a full-sky survey built from the set of maximal circles in the sky obtained by spacecraft rotation in the anti-solar direction. The amount of data is not that dramatic (2 Gpixels/day) but there is a number of computational challenges. One of these is related to map-making: there is the need to invert the NxN noise covariance matrix, where N is the total number of mesurements in one frequency channel. A brute force approach would lead to a computation time of 10^9 years on a 1 GHz CPU. Several assumptions can be made to simplify the computation and make it feasible, but in any case the speed of the algo-

Send offprint requests to: F. Pasian
Correspondence to: Oss. Astron. di Trieste, via Tiepolo n.11, 34131 Trieste

rithms strictly depends on the amount of data which may be kept in RAM.

It is clear that the cases discussed depend on the "memory footprint" of the applications. If we can sub-divide the data to be processed, a local cluster of PCs such as a Beowulf machine, or even a grid of distributed PCs, can be a low-cost but effective solutions to the problem. In the case we have to run applications needing data to be shared among all CPUs involved in the computation, clearly parallel HPC is the ideal solution. However, ad-hoc hybrid solutions may be found, and they may be different depending on the specificities of the problem. As an example, "one-shot" computations can be treated differently than applications needing continuous processing "operations".

In any case, before using expensive HPC machines for data processing (but also theoretical!) problems, both an accurate analysis of the "memory footprint" and a careful speed-vs.-RAM-consumption trade-off study must be done for every application. To solve our problems, machines which are cheaper or easier to use (e.g. our institute's LAN during the night) may be just as appropriate.