

Clustering the large VizieR catalogues, the CoCat experience

François Ochsenbein, Sébastien Derriere, Sébastien Nicaisse, André Schaaff

Centre de Données astronomiques de Strasbourg (CDS), Observatoire de Strasbourg, UMR 7550, 11 rue de l'Université, 67000 Strasbourg, France

Abstract. VizieR is a database containing about 4000 astronomical catalogues with homogeneous descriptions. The major part of the catalogues is stored in a relational database but the large catalogues containing over 10 millions rows are stored as compressed binary files and have dedicated query programs for very fast access by celestial coordinates. The CoCat (Co-processor Catalogue) project main goal is to parallelize the VizieR large catalogue treatments (data extraction, cross-matching) for reducing the response time.

1. Introduction

The VizieR catalogue service (Ochsenbein et al., 2000) is currently implemented on a Sun 4-processor server. In the recent years the competitiveness of PCs dramatically increased, with very high performances and ever decreasing costs, and in many circumstances, clusters of Linux PCs are replacing the big standalone servers. In the VizieR case, the current load is high and it was urgent to choose between a complete replacement or an additional server.

2. Organisation of the Catalogues

VizieR catalogues are divided into two categories: *standard* and *large* catalogues, where large catalogues are defined, somewhat arbitrarily, as having more than 10^7 rows. Catalogues with up to a few million records are managed by a standard relational DBMS, while each of the larger catalogs has a dedicated query program which retrieves the records corresponding to a some circular or rectangular region around a position in the sky. Some details about the methods used to store the large catalogues and their performances, in terms of speed and disk usage, are given in Derriere et al. (2000); the current list of these large catalogues is given in Fig.1. It should be noted that both “standard” and “large” catalogues share the same metadata descriptions — the VizieR interface simply translates the user’s requests either into SQL queries, or into some customized set of parameters interpreted by the dedicated query program.

Acronym	Rows $\times 10^6$	Title of the Catalogue	Size Gbytes
GSC-1.1	† 25	The HST Guide Star Catalog, Version 1.1 (Lasker+ 1992)	0.3
GSC-1.2	25	The HST Guide Star Catalog, Version 1.2 (Lasker+ 1996)	0.3
GSC-ACT	25	The HST Guide Star Catalog, Version GSC-ACT (Lasker+ 1996-99)	0.3
USNO-A1.0	† 488	The PMM USNO-A1.0 Catalogue (Monet 1997)	3.3
USNO-A2.0	526	The USNO-A2.0 Catalogue (Monet+ 1998)	3.5
USNO-B1.0	1046	The USNO-B1.0 Catalog (Monet+ 2003)	39.4
GSC2.2	456	The Guide Star Catalog, Version 2.2 (STScI, 2001)	43.3*
APM-North	166	The APM-North Catalogue (McMahon+, 2000)	10.1
UCAC1	† 27	The UCAC1 Catalogue (Zacharias+ 2000)	0.4
UCAC2	48	The UCAC2 Catalogue (Zacharias+ 2003)	1.6
2MASSIpsc	† 162	The 2MASS Catalog Intermediate Data Release (IPAC/UMass, 2000)	12.1
2MASS-PSC	741	The 2MASS All-Sky Catalog of Point Sources (Cutri+ 2003)	40.8
DENIS-P	† 17	The DENIS database first release (Epchtein+, 1999)	3.4
DENIS-2	195	The DENIS database (DENIS Consortium, 2003)	14.2

† obsolete version of the catalog
* no attempt was made to compress the GSC2.2 catalog

Figure 1. The large catalogues in the cluster (*version October 2003*)

3. Which architecture?

As the Sun server is becoming overloaded we decided to move the set of large catalogues to a Linux cluster (the *CoCat* cluster). It then becomes easy to increase the computing power or the storage capability at a very low cost; it represents also a flexible solution for the future evolutions.

A wide range of free or commercial clustering tools is available. We started with a new free clustering tool package, CLIC¹ (Cluster Linux pour le Calcul) which makes use of the MPI library (Message Passing Interface) and is based on the Mandrake Linux 9.0 distribution. The CoCat cluster involves one master node and five slave nodes (Fig 2).

4. The Dispatcher

Tools like MPI are designed to run parallelized CPU-intensive tasks on a cluster, but in the CoCat case it is necessary to dispatch a large number of queries (typically 10^5 – 10^6 daily requests) and their results. The large catalogues being

¹<http://cllc.mandrakesoft.com/>

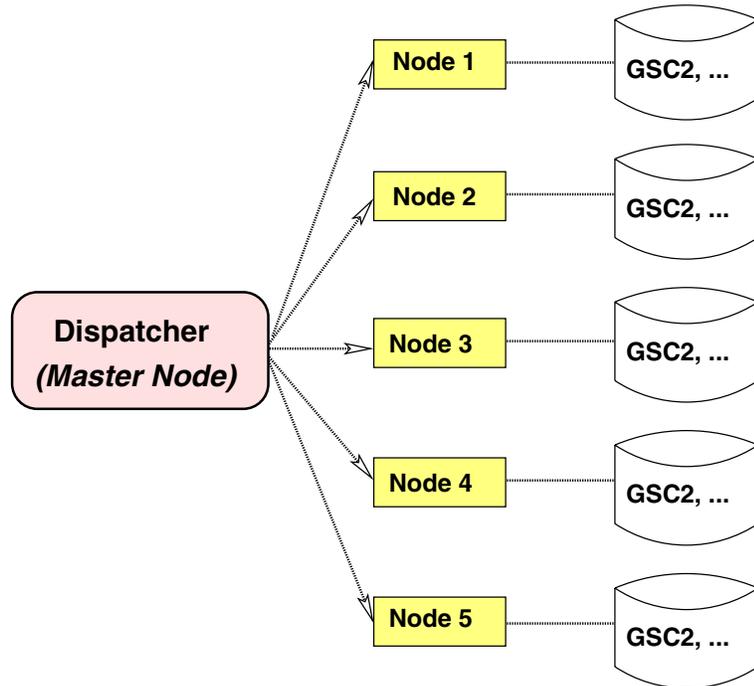


Figure 2. The CoCat cluster.

stored in a compact form, it was possible in a first step to replicate the data (about 200Gbytes) on each node. With the increasing number of increasingly larger catalogues it will be necessary in the near future to distribute the data over several nodes, and it will become mandatory to describe on which engines which part of which catalogue can be accessed: this role is devoted to the Dispatcher, running on the master node, and illustrated in Fig. 3.

5. The first tests

The first tests showed that the performances are not as high as expected: the overhead of the MPI library is large compared to the time required by the actual execution of the requests initiated by the Dispatcher. The CLIC package, while easing up the installation of the system and the applications on the cluster nodes, requires an identical hardware configuration of each node: this introduces a severe lack of flexibility in the management and the evolution of the cluster.

We are currently testing new configurations for a more performant Dispatcher, where each node is considered as an independent resource and where the Dispatcher assigns the tasks according to its knowledge of the current load on each node. Such a method seems to work well in the current situation where all catalogues are present on each node, but in a near future we will have to take some important decisions about:

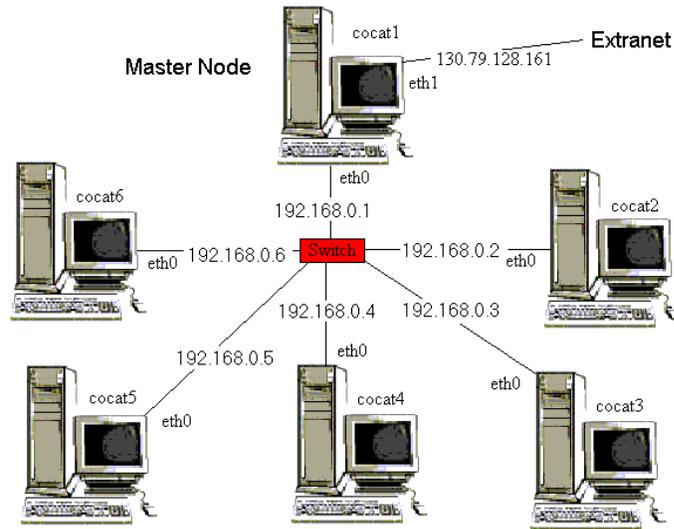


Figure 3. CoCat global architecture

- which strategy to adopt about splitting the very large catalogues and how to distribute catalogue subsets on the different cluster nodes
- whether it would be useful to dedicate one or several nodes to specific tasks (e.g. cross-matching)
- whether it would still be useful to implement a parallel processing (e.g. for cross-matching large catalogues) in the dispatcher.

References

- Derriere, S., Ochsenbein, F., & Egret, D. 2000, in ASP Conf. Ser., Vol. 216, ADASS IX, ed. N. Manset, C. Veillet, & D. Crabtree (San Francisco: ASP), 235
- Ochsenbein, F., Bauer, P., & Marcout, J. 2000, A&AS, 143, 23