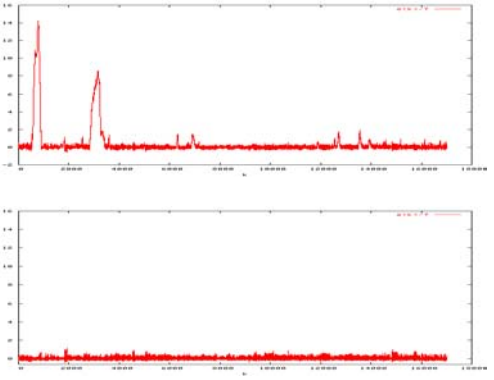
Quality of Service

QMGR – QoS Manager

The problem of finding a correct allocation of bandwidth to the different tasks is still to be considered as a tough problem. This problem is addressed in this software components and it essentially amounts to finding an appropriate allocation of resources to competing tasks. To this regard, a **static allocation** of CPU bandwidth confronts unfavourably to a **dynamic one**. For instance a typical movie alternates ''fast scenes'' (which induce a **heavy computation workload**) to ''slow'' dialogues between the characters (which on the contrary require a **light computation**). An allocation based on fast scenes can lead to resource wastes in certain moments. On the other hand, an allocation based on the average, leads to poor QoS when computational peaks occur. To take advantage of this structure, we propose a **dynamic adjustment of the bandwidth**, in which a software module complementary to the scheduler collects QoS measures for the executing tasks and varies the assigned bandwidth accordingly. A key advantage of this idea is that the system is self-tuning: the user is not required to provide parameters he/she has little knowledge of.

Features

- Permits the use of soft real-time tasks
- Quality of Service of applications
- Dynamic adjustment of the bandwith
- IRIS constant bandwith server



In the picture above we compare the behaviour of the same task (an mpeg player) with two different allocation: static and dynamic. The value misured is the difference between the desired finishing time and the experienced one (compared to the period of the task). As we can see, due to a temporary heavy computation workload, the execution in the former case is delayed for several periods, while in the latter one this does not happens.

With the QoS OCERA components you can run in the same system several applications with different allocation policies: static, dynamic and with no allocation (background). In the picture the two mpeg players are running with dynamic bandwidth allocation, while a never-ending loop that tries to use all cpu-time loop is forced to use only a small fraction of cpu (to avoid interference with mpeg players)