



★ Interconnection & Interoperability of Grids between Europe & China ★

1.1. CONDOR'S SKECTH

The goal of the Condor® Project is to develop, implement, deploy and evaluate mechanisms and policies supporting [High Throughput Computing \(HTC\)](#) on large collections of distributively-owned computing resources.

1.1.1. What is High throughput computing (HTC)

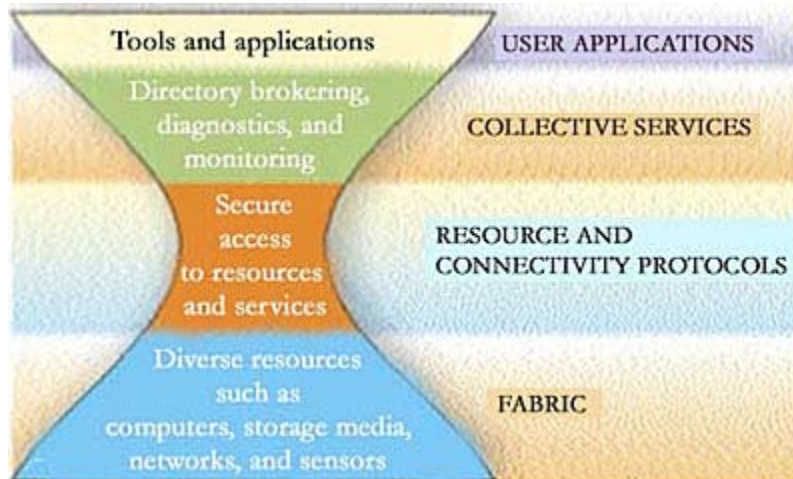
For many experimental scientists, scientific progress and quality of research are strongly linked to computing throughput. In other words, most scientists are concerned with how many floating point operation per month or per year they can extract from their computing environment rather than the number of such operations the environment can provide them per second or minute. Floating point operations per second (FLOPS) has been the yardstick used by most High Performance Computing (HPC) efforts to evaluate their systems. Little attention has been devoted by the computing community to environments that can deliver large amounts of processing capacity over long periods of time. We refer to such environments as High Throughput Computing (HTC) environments.

For more than a decade, the Condor team at the Computer Sciences Department at the University of Wisconsin-Madison has been developing and evaluating mechanisms and policies that support HTC on large collections of distributively owned heterogeneous computing resources. At the first time was introduced the distinction between High Performance Computing (HPC) and High Throughput Computing (HTC) in a seminar at the NASA Goddard Flight Center in July of 1996 and, a month later, at the European Laboratory for Particle Physics (CERN). In June of 1997 HPCWire published an [interview](#) on High Throughput Computing.

The key to HTC is effective management and exploitation of all available computing resources. Since the computing needs of most scientists can be satisfied these days by commodity CPUs and memory, high efficiency is not playing a major role in a HTC environment. *The main challenge a typical HTC environment faces is how to maximize the amount of resources accessible to its customers.* Distributed ownership of computing resources is the major obstacle such an environment has to overcome in order to expand the pool of resources it can draw from. Recent trends in the cost/performance ratio of computer hardware have placed the control (ownership) over powerful computing resources in the hands of individuals and small groups. These distributed owners will be willing to include their resources in a HTC environment only after they are convinced that their needs will be addressed and their rights protected.

1.1.2. GRID Architecture in a nutshell

One of the most important "lessons learned" from Grid application projects is that trying to force homogeneity on distributed groups of collaborators is futile. Each segment of the group will have its own system administrators, its own business office, its own set of rules and regulations that get in the way of homogeneous practices. The collaborators themselves will have their own preferences, which sometimes rise almost to the level of dogma, resisting change or conformity with other ideas. Forcing people to give up their local ways of doing things requires more energy than most projects have at their disposal.



The Open Grid Services Architecture extends the IP network services upward from physical transport of messages to a variety of services that have proven common in Grid applications. OGSA defines a service-oriented architecture, which can be regarded as the key to effective virtualization of physical resources. OGSA services enable applications to address common Grid requirements for on-demand availability, system management, collaborative computing, and so on. The OGSA builds on existing Web service standards and extends these standards when needed.