Weather forecasts for long periods of time have emerged as increasingly important. The global concern with the consequences of climate change has stimulated research to determine the climate in coming decades. At the same time the steps needed to define better modeling and simulation of climate/weather are far from the accurate desired. Upscaling the land surface and consequently increase the number of points used in climate modeling is a goal that conflicts with the performance of numerical applications. Applications that include the interaction of long periods of time and involve a large number of operations become the expectation for results inviable in today computer architectures.

To overcome this situation, a climatic model can take different levels of refinement of the earth's surface. This is the case of Ocean-Land-Atmosphere Model (OLAM), which allows the static refinement of a particular region in the early execution of the code. However, a dynamic refinement would allow a better understanding of the specific climatic conditions of any region of the earth's surface that had an interest, especially over long periods of simulation, where there would be no need to restart the application execution. A mesh refinement in execution time increases the processing load and therefore it is necessary to apply load balance among parallel processing units. This task becomes more complicated when different levels of parallelism are found in current architectures.

Thus, this presentation discusses how to explore parallelism at different levels for climatological models, like OLAM, using classical parallel programming interfaces. We use the notion of parallel tasks as a way of abstract the parallel granularity (processes, threads) for a concurrent application in order to offer load balancing to a mesh refined at runtime. Thus, we expect to provide higher performance for the execution of the code and clarity in the specification of parallelism.