OpenSees Profiling Activities

Rationale

One of QuakeCoRE's strategical objectives is to be able to perform OpenSees simulations at an HPC scale. However, OpenSees does not seem to present a good scaling on bigger machines (http://opensees.berkeley.edu/OpenSees/workshops/parallel/ParallelOpenSees.pdf). A profiling exercise can provide more insight on the reasons behind the lack of good speedup of OpenSees.

Profiling

There are two flavors of parallel OpenSees:

- OpenSeesSP: a single interpreter running on a processor 0. This processor acts as a master and distributes the workload to the other n-1 worker
 processors. The changes required to convert a normal Tcl script to run with this interpreter are minimal and the parallelization is hidden from the
 user
- OpenSeesMP: we run n interpreters, one on each processor. Domain decomposition and communications amongst the processors must be specified by the user.

In a first stage, we are going to focus on the profiling of OpenSeesSP, as it seems to be more transparent for the user.

Current activities

Running a decently sized job on Pan to investigate scaling and using Mumps compiled with different compiler and backend (SCOTH, Imkl).

Next steps

Contact NeSI expert on Pan profiling to see if he can help with the issues I encountered with the Intel profiler.

History

- OpenSees Initial Profiling
- OpenSees and Allinea
- OpenSees and GPU
- OpenSees on Pan