Unified diff: `step5_pi_checkpoint` *vs.* `step6_pi_mpi`

```
Files step5_pi_checkpoint/CMakeLists.txt and step6_pi_mpi/CMakeLists.txt are identical
```

```
diff -N -u -r -b -s step5_pi_checkpoint/main.cpp step6_pi_mpi/main.cpp
--- step5_pi_checkpoint/main.cpp        2016-06-15 14:12:52.874919385 -0400
+++ step6_pi_mpi/main.cpp       2016-06-15 13:43:10.271002476 -0400
@@ -1,29 +1,39 @@
 #include <iostream>
 #include <alps/mc/stop_callback.hpp>
+#include <alps/mc/mpiadapter.hpp>
+
 #include "simulation.hpp"

 int main(int argc, char** argv)
 {
     // Define shorthand:
-    typedef MySimulation mysim_type;
+    typedef alps::mcmpiadapter<MySimulation> mysim_type;
+    
+    // init MPI, obtain communicator
+    alps::mpi::environment mpi_env(argc, argv);
+    alps::mpi::communicator comm;
+    // remember the rank
+    const int rank=comm.rank();
+    const bool is_master=(0==rank);

     // Parse the parameters
-    alps::params p(argc, (const char**)argv);
+    alps::params p(argc, (const char**)argv, comm);

     // Define the parameters
-    mysim_type::define_parameters(p)
-        .define<std::size_t>("timelimit", 5, "Time limit for the computation");
+    mysim_type::define_parameters(p);

     if (p.help_requested(std::cerr) || p.has_missing(std::cerr))
         return 1;

     std::cout << "Creating simulation"
+              << " on rank " << rank
               << std::endl;

-    mysim_type mysim(p);
+    mysim_type mysim(p,comm);

     // If needed, restore the last checkpoint
     std::string checkpoint_file = p["checkpoint"].as<std::string>();
+    if (!is_master) checkpoint_file += "."+boost::lexical_cast<std::string>(rank);

     if (p.is_restored()) {
         std::cout << "Restoring checkpoint from " << checkpoint_file
@@ -32,23 +42,29 @@
     }

     std::cout << "Starting simulation"
+              << " on rank " << rank
               << std::endl;

     mysim.run(alps::stop_callback(std::size_t(p["timelimit"])));

     std::cout << "Simulation finished"
+              << " on rank " << rank
               << std::endl;

     std::cout << "Saving to checkpoint " << checkpoint_file
+              << " on rank " << rank
               << std::endl;

     mysim.save(checkpoint_file);

     std::cout << "Collecting results..."
+              << " on rank " << rank
               << std::endl;

     alps::accumulators::result_set results=mysim.collect_results();

+    // Do printing only on master:
+    if (is_master) {
     // Print all results:
     std::cout << "All results:\n" << results << std::endl;

@@ -78,6 +94,7 @@
     alps::hdf5::archive ar(output_file, "w");
     ar["/parameters"] << p;
     ar["/simulation/results"] << results;
+    }

     return 0;
 }

Files step5_pi_checkpoint/simulation.cpp and step6_pi_mpi/simulation.cpp are identical

Files step5_pi_checkpoint/simulation.hpp and step6_pi_mpi/simulation.hpp are identical
```