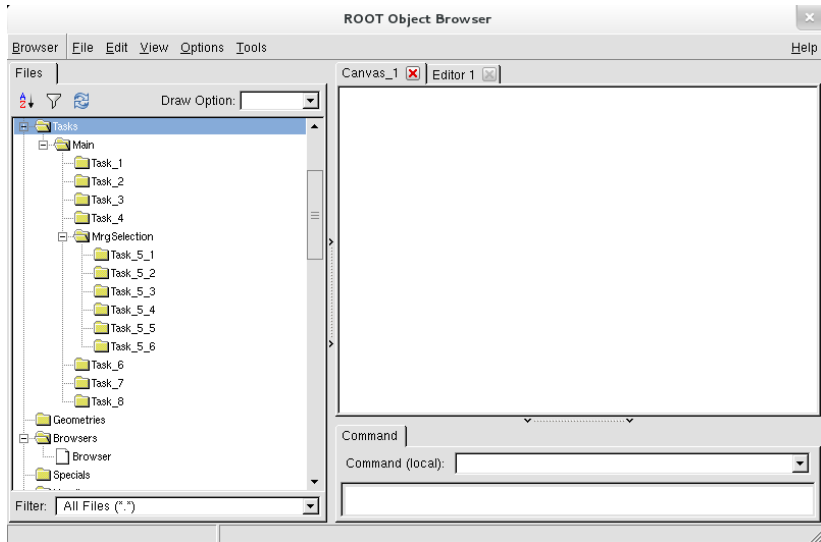


New ROOT concurrent classes

Martin Vala

June 6, 2012

Parallel tasks



- Both classes are TTask
- TParallelTaskManager
 - Mangment of sub tasks and sub managers
 - It can access to parent manager
 - subTask->SetActive(...)
 - Dependencies (TODO)
- TParallelTask
 - `Int_t TParallelTask::ThreadStart()`
 - `Int_t TParallelTask::ThreadStop()`
 - `void *TParallelTask::Thread0(void *arg)`

Example and demo [1]

```
{
  TParallelTaskManager *mgrMain = new TParallelTaskManager("Main", "Main
Manager", runParallel);
// mgr->SetMaxThreads(2);
  TParallelTaskManager *mgrSel = new TParallelTaskManager("MgrSelection", "Selection
Manager", runParallel);
// mgrSel->SetActive(kFALSE);
  AddParallelTask(mgrMain, "Task_1", "Task 1", 2.0);
  AddParallelTask(mgrMain, "Task_2", "Task 2", 1.0);
  AddParallelTask(mgrMain, "Task_3", "Task 3", 1.0);
  AddParallelTask(mgrMain, "Task_4", "Task 4", 1.5);
  mgrMain->Add(mgrSel);
  if (mgrSel->IsActive()) {
    AddParallelTask(mgrSel, "Task_5_1", "Task 5 1", 2.0);
    AddParallelTask(mgrSel, "Task_5_2", "Task 5 2", 1.0);
    AddParallelTask(mgrSel, "Task_5_3", "Task 5 3", 1.5);
    AddParallelTask(mgrSel, "Task_5_4", "Task 5 4", 1.0);
    AddParallelTask(mgrSel, "Task_5_5", "Task 5 5", 2.0);
    AddParallelTask(mgrSel, "Task_5_6", "Task 5 6", 1.0);
  }
  AddParallelTask(mgrMain, "Task_6", "Task 6", 2.0);
  AddParallelTask(mgrMain, "Task_7", "Task 7", 2.5);
  AddParallelTask(mgrMain, "Task_8", "Task 8", 2.5);
  mgrMain->ExecuteParallel("");
}
```

Example and demo [2]

```
TParallelTask *AddParallelTask(TParallelTaskManager *mgr, const char
*name, const char *title, Long64_t masxum=1, Bool_t isActive=kTRUE) {
    if (!mgr) {
        Printf("Task %s was NOT added. mgr is null !!!", name);
        return 0;
    }
    TParallelTaskTest *t= new TParallelTaskTest(name, title);
    t->SetMaxNum(masxum);
    mgr->Add(t);
    return t;
}
```

Example and demo [3]

```
void TParallelTaskTest::Exec(Option_t *option)
{
    TParallelTask::Exec(option);
    TStopwatch timer;
    timer.Start();
    Long64_t base=1e8;
    Int_t numThreads = TThread::Exists();
    Printf("Running %s::Exec(\"%s\") num=%lld [START] %d",GetName(),option,fMaxNum,
numThreads);
    Long64_t sum = 0;
    TH1D h("h", "Hist", 100, -10, 10);
    h.SetDirectory(0);
    h.FillRandom("gaus", fMaxNum*base);
    numThreads = TThread::Exists();
    if (numThreads<0) numThreads=0;
    Printf("Running %s::Exec(\"%s\") num=%lld [DONE] %d",GetName(),option,fMaxNum,
numThreads);
    timer.Stop();
}
```

- <https://github.com/mvala/sunrise>
- `git clone git://github.com/mvala/sunrise.git`
- `cd sunrise/scripts`
- `# setup ROOT env`
- `./make.sh`
- `# 1 core`
 - `./runParallelTasks.sh 0`
- `# num of cores on PC`
 - `./runParallelTasks.sh 1`
- `# 2 cores`
 - `./runParallelTasks.sh 1 2`