

Asynchrones Restart-Modul: Kurzdokumentation

Jörg Benkenstein, Ernst Basler + Partner GmbH

August 2, 2013

1 Aufgabenstellung für das asynchrone Schreiben der Restart-Dateien

ICON besitzt die Möglichkeit, für lange Simulationszeiträume und / oder zur Sicherung von Zwischenergebnissen im Falle von Hardwaredefekten während der Vorhersagerechnung Restart-Dateien zu schreiben, um an den entsprechenden vordefinierten Zeitpunkten die Simulation wieder aufnehmen zu können. Das Schreiben der Restart-Dateien erfolgt derzeit synchron durch den Standard-Output-Prozessor (PE 0), d.h. alle anderen Prozessoren befinden sich während dieser Zeit im Wartezustand.

Dieses Problem ist durch die Implementierung einer Option für asynchrones Schreiben der Restart-Dateien auf dedizierten Prozessoren zu beheben. Diese sollen nicht zwingend identisch mit den Prozessoren für den asynchronen Regel-Output sein.

2 Implementierte Lösung für das asynchrone Schreiben der Restart-Dateien

Das asynchrone Schreiben der Restart-Dateien kann mit einem oder mehreren Restart-Prozessoren ausgeführt werden. Die Anzahl der verwendeten Prozessoren wird per Konfiguration im Bereich "parallel_nml" durch den Wert der Variablen "num_restart_procs" gesteuert, z.B.:

```
num_restart_procs = 1
```

Der Default-Wert der Variable ist 0. In diesem Fall wird kein asynchroner Restart durchgeführt, sondern der Restart wird vom Compute-Prozessor 0 geschrieben. Bei einem Wert größer 0 werden entsprechend viele Restart-Prozessoren erzeugt. Die Restart-Prozessoren schreiben die für einen Restart des Modells benötigten Variablen und Listen in ein oder mehrere Restart-Dateien im NetCDF-Format. Die Restart-Prozessoren führen während des gesamten Modelllaufs keine weiteren Aufgaben aus.

Der asynchrone Restart ist auf der Basis von MPI-Kommunikation zwischen Compute- und Restart-Prozessoren implementiert worden und kann folglich nur genutzt werden, wenn die ausführbare Anwendung des Modells entsprechend compiliert worden ist.

Ist "num_restart_procs" auf einen Wert größer 0 gesetzt und die Anwendung mit "-DNOMPI" compiliert worden, wird der Modelllauf mit einer Fehlermeldung abgebrochen.

Jeder Restart-Prozessor erhält von den Compute-Prozessoren die Daten eines ICON-Patches. Diese werden in eine Restart-Datei ausgegeben.

3 Ausblick

Die Schreibfunktionalität wurde bereits vorbereitet, dass bei paralleler Ausgabe mehrerer ICON-Patches durch mehrere Restart-Prozessoren die erzeugten Restart-Dateien im Namen um den Wert eines Zählers erweitert werden. Um diese Funktionalität verwenden zu können, muss aber noch die Import-Seite codemäßig

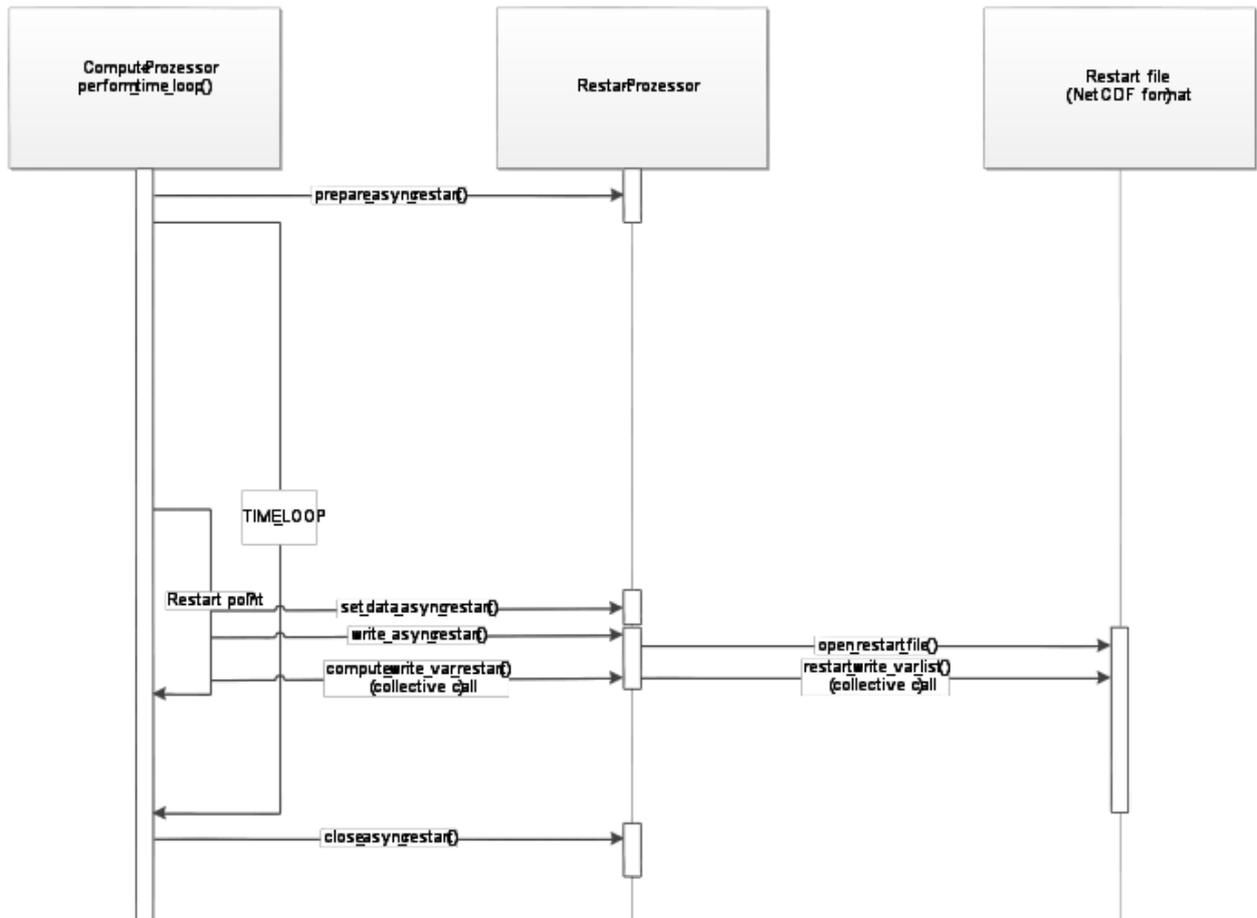


Figure 1: Sequenzdiagramm des asynchronen Restarts

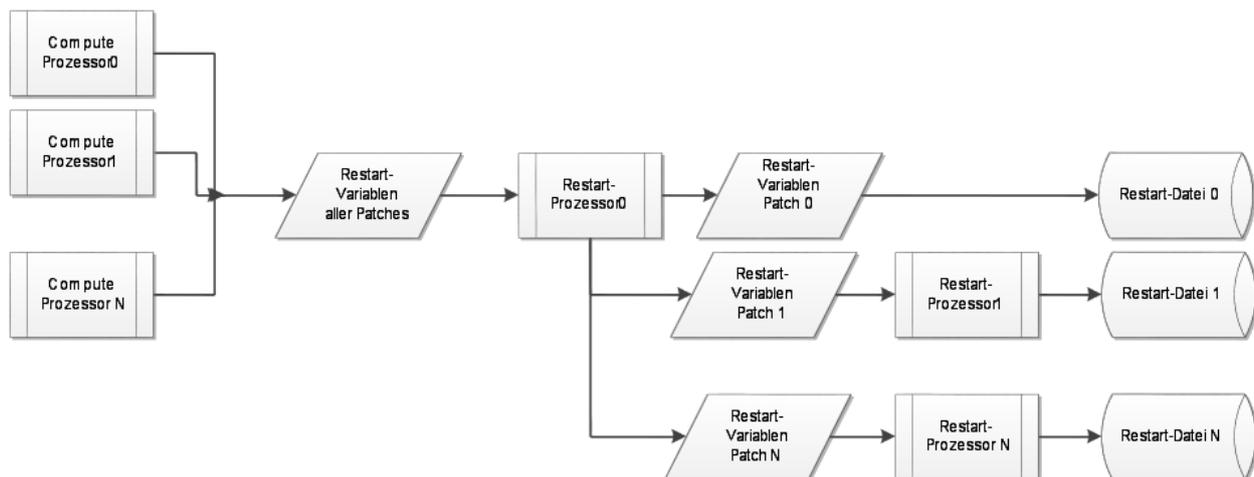


Figure 2: Datenfluss beim asynchronen Schreiben der Restart-Dateien

angepasst werden, damit die erzeugten Restart-Dateien wieder entsprechend eingelesen werden. Zum Zeitpunkt der Entwicklung des asynchronen Restarts wurde durch den Restart-Import nur eine Restart-Datei importiert.

Im Vergleich zum synchronen Restart müssen die Compute-Prozessoren nicht darauf warten bis der zugehörige Restart-Prozessor seine Daten in die Restart-Datei ausgegeben hat, sondern können unmittelbar mit Ihren Berechnungen fortfahren, sobald sie ihre Daten in einem vom Restart-Prozessor eingerichteten Speicher-Bereich geschrieben haben.

Die Restart-Prozessoren lesen die Restart-Variablen aus diesem Speicherbereich und schreiben sie in ein oder mehrere Restart-Dateien, während die Compute-Prozessoren bereits weitere Berechnungen durchführen. Zeitlich begrenzt wird der beschriebene Vorgang nur dadurch, dass die Restart-Prozessoren ihre Arbeit erst beendet haben müssen, bevor sie für eine weitere Restart-Ausgabe zur Verfügung stehen.

Müssen Compute-Prozessoren auf die Beendigung der Restart-Ausgaben warten, bis sie einen weiteren Zeitschritt exportieren können, sollte die Anzahl der verwendeten Restart-Prozessoren erhöht werden.

Bei der Implementierung des asynchronen Restarts wurden folgende Fortran-Dateien geändert bzw. neu erstellt:

1. src/drivers/mo_atmo_model.f90
2. src/atm_dyn_iconam/mo_nh_stepping.f90
3. src/atm_dyn_icoham/mo_ha_stepping.f90
4. src/parallel_infrastructure/mo_mpi.f90
5. src/configure_model/mo_parallel_config.f90
6. src/io/mo_io_restart_namelist.f90
7. src/io/mo_io_restart_async.f90
8. src/namelists/mo_parallel_nml.f90