Parallel computing and model coupling for data assimilation

Nils van Velzen

JONSMOD
May 11 2010
Overview

- OpenDA in a nutshell
- Object oriented design and model component
- Automatic parallelization in OpenDA
- Coupling with Parallel models in OpenDA
- Conclusions
OpenDA in a nutshell
Object oriented design and model component

- Object oriented design
  - Classes, software building blocks
  - State (variables) of a class is NOT accessible from outside
  - State can be indirectly accessed/changed using the methods from the interface
Object oriented design and model component

- Model in OpenDA (formal)

\[
\frac{dx(t)}{dt} = M\left(x(t), p, u(t), w(t)\right)
\]

- State of a model instance \( x, u, p, w, t \)

- Methods to get or change the model-state
Object oriented design and model component

- Propagate the model state-vector
  \[ x(t) = \int_{t}^{t+\Delta t} M(x(t), p, u(t), w(t)) dt \]

- Get, set, axpy for \( x, u, p, w, t \)

- GetObsValues: \( y(t) = H(x(t)) \)
Automatic parallelization in OpenDA

- Note:
  - Multiple model instances for multiple states
  - State of model is NOT directly accessible
  - Propagating of state is *NON-BLOCKING*

- Propagating multiple modes can be done in parallel

- Interface of all models is the same in OpenDA
  - One generic way to support parallelism for all models
Automatic parallelization in OpenDA

- Relevant for many algorithms
  - EnKF
  - RRSQRT
  - Ensrf
  - Finite difference gradients
- Often propagating states takes the most time
Automatic parallelization in OpenDA
Automatic parallelization in OpenDA

Diagram showing the interaction between a master process and worker processes, detailing the communication and model instances involved.
Automatic parallelization in OpenDA

- Testcase with LOTOS-EUROS air quality model
- Compare various DA methods
- Investigate impact of automatic parallelism
- Set up an ozone test-case
- 38 observation stations
Automatic parallelization in OpenDA
Automatic parallelization in OpenDA
Automatic parallelization in OpenDA
Coupling with Parallel models in OpenDA

- Various forms of parallel computing
  - parallelized using threads
  - Multiple processes
    - master-worker programming model
      - Master represents the whole model
    - Worker-worker programming model
      - The model is a concatenation of sub (worker) models
Coupling with Parallel models in OpenDA
Automatic parallelization in OpenDA
Automatic parallelization in OpenDA

- Proof of concept: WAQUA/TRIWAQ

DDHOR (parallel) CZUNO model
Automatic parallelization in OpenDA
Automatic parallelization in OpenDA
Conclusions

- OpenDA is a flexible framework for data assimilation and model calibration
- Easy to experiment with various DA-methods
- Automatic parallelization to improve performance
- Parallel models can be used in OpenDA as well
- Illustrated using real operational models
Questions ?!?