Parallel computing and model coupling in OpenDA

Nils van Velzen

Deltares DA-meeting
Feb 18 2010
Overview

- Background and motivation for OpenDA
- Object oriented design and model component
- Automatic parallelization in OpenDA
- Coupling with Parallel models in OpenDA
- Conclusions
Background and motivation for OpenDA

- Simulation models
  - Weather forecast
  - Air quality
  - Shallow water
  - Ocean modeling

- Predictions are not perfect!
Background and motivation for OpenDA

- Measurements:
  - Satellites
  - Buoys
  - Radar
  - Weather stations

- Measurements are not perfect!
Background and motivation for OpenDA

- Make model perform better using the available observations:
  - Data assimilation: combine the model forecast and observations into a mixed forecast
  - Calibration: change model parameters in order to reduce the difference between the model predictions and observations
Background and motivation for OpenDA

- Simulation software can be (extremely) complex
- More complex is combined with Data assimilation and or Calibration methods
- Expensive to develop and maintain
Background and motivation for OpenDA

- Split complex software systems up in (well defined!) parts
  - Re-use of code
  - Less errors in software
  - Less complex building blocks
  - Possible to replace parts of the system
  - Cooperation between researchers and institutes
Background and motivation for OpenDA
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

- Recall:
  - 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

costawb-process 1 (Master process)

Data assimilation method

Model Instances

Parallel interface model instance

Number of worker process
Model instance ID

Parallel interface model instance

Number of worker process
Model instance ID

Parallel interface model instance

Number of worker process
Model instance ID

messages

costawb-process 2 (Worker process 1)

Worker

Model Instances

Dynamic model instance

$x, u, p, w$

costawb-process 3 (Worker process 2)

Worker

Model Instances

Dynamic model instance

$x, u, p, w$
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
Automatic parallelization in OpenDA

![Graph showing computation time vs. number of processors]
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 ?!?