Architecture Recovery from Fortran Code with Kieker

Symposium on Software Performance

Reiner Jung, Henning Schnoor,
Sven Gundlach, Wilhelm Hasselbring

8th November 2022

OceanDSL
Digital Twin

- a model of the object
- an evolving set of data relating to the object
- a means of dynamically updating or adjusting the model in accordance with the data

[Wright and Davidson 2020]
Models we analyzed so far (1/2)

**UVic (University of Victoria, ESM)**

- Models the complete earth
- Includes atmosphere, oceans, ice, land, bio-geo-chemical processes in the ocean
- Written in Fortran 77 and Fortran 90
- No VCS, no centralized infrastructure
- Self-made configuration and build system

**MITgcm (MIT General Circulation Model, ESM)**

- Versatile model, can model the complete earth system
- Provides wide range on example models also used for testing
- Written in Fortran 77 and 90
- Uses git
- Feature model
- Self-made configuration and build system


Models we analyzed so far (2/2)

Shallow-Water-Model

- Written in Fortran 95 with modules
- Uses git
- Standard Makefile, uses autoconf, automake

[Claus 2016]
Upcoming Models

ICON (Icosahedral Nonhydrostatic Model)
- Global weather model
- Deutscher Wetter Dienst (DWD)

ECHAM5
- Atmospheric general circulation model
- MPI for meteorology

Metos3D
- Marine/Ocean ecosystem model toolkit
- Partly in Python and Fortran
Overview Process

1. Understand Build Process
2. Configure Model and Setup Parameters
3. Instrument Scientific Model
4. Execute Model
5. Recover Architecture
6. Recover Interfaces
7. Inspect Recovered Architecture

Diagram:
- Understand Build Process → Configure Model and Setup Parameters → Instrument Scientific Model
- Execute Model → Recover Architecture
- Recover Interface → Inspect Recovered Architecture
Log events

- Kieker collector
- Netcat & Split
  ```bash
  nc -l 5678 | split -b 102400000 - log-
  ```

What is logged?

- Subroutine calls
- Function calls
- Procedure calls
Recover Architecture

Reconstruction

• Resolve operation and file names with addr2line
• Construct operation uses and calls from Kieker events
• Create type, assembly and deployment model based on operations
• Add and aggregate call information to the Kieker execution model

Component Identification

1. Names of the files
2. Directory names of files
3. Fully qualified operation signatures in o-files, e.g., _module_MOD_operation_

⇒ We can apply all methods in combination
Approaches

1. Large interface per component to component connection
   - few interfaces
   - different provided interfaces may share operations

2. One provided interface per component, multiple required interfaces
   - fewest interfaces
   - each component has only one provided interface
   - can lead to wide interfaces

3. Provided interfaces have operations that are required by the same requiring components
   - May create too many interfaces when different subsets of operations are used
Current Discovery Strategy

Provided Interfaces

- Identify for each operation all caller components
- Group all callees that have the same set of caller component
- Create a provided interface for each callee group

Required Interfaces

- Create one for each used provided interface by a component
- Link all caller that call callees of a provided interface to the corresponding required interface
Energy-Moisture Balance Model (EMBM)
Lessons Learned

Fortran

- F77 uses global symbols, no name spaces
- F95 can use modules, o-file symbols use FQN _module_MOD_operation_
- Names are case insensitive, o-file symbols are lower case with ’_’ as prefix

Scientific Models

- Every scientific model has its own build system or use of build tools
- High degree of interconnection between components
Conclusions

Advantages of dynamic recovery

- Object files are sufficient
- Debugging symbols are helpful
- Understanding code assembly and build procedure not necessary
- Can show the number of calls in an interface to rank functions
- Not limited to Fortran
- Fast setup

Disadvantages

- Requires a running executable
- Can result in a lot of monitoring data
- Cannot cover dataflow (but our static recovery does)

