

# Environment Management and Multi-Machine Builds

# Environment and Build Management

- Consider this case:
  - Real synchronized project, with makefile
  - Would like to flip back and forth between compilers (eg Cray or PGI) or same compiler (openACC or straight CPU code) and stay with same code base
  - Currently: 1 version synchronized (cray compiler)
    - Need to login and do PGI builds manually/requires different modules

# Environment and Build Management (2)

- Makefile excerpt:

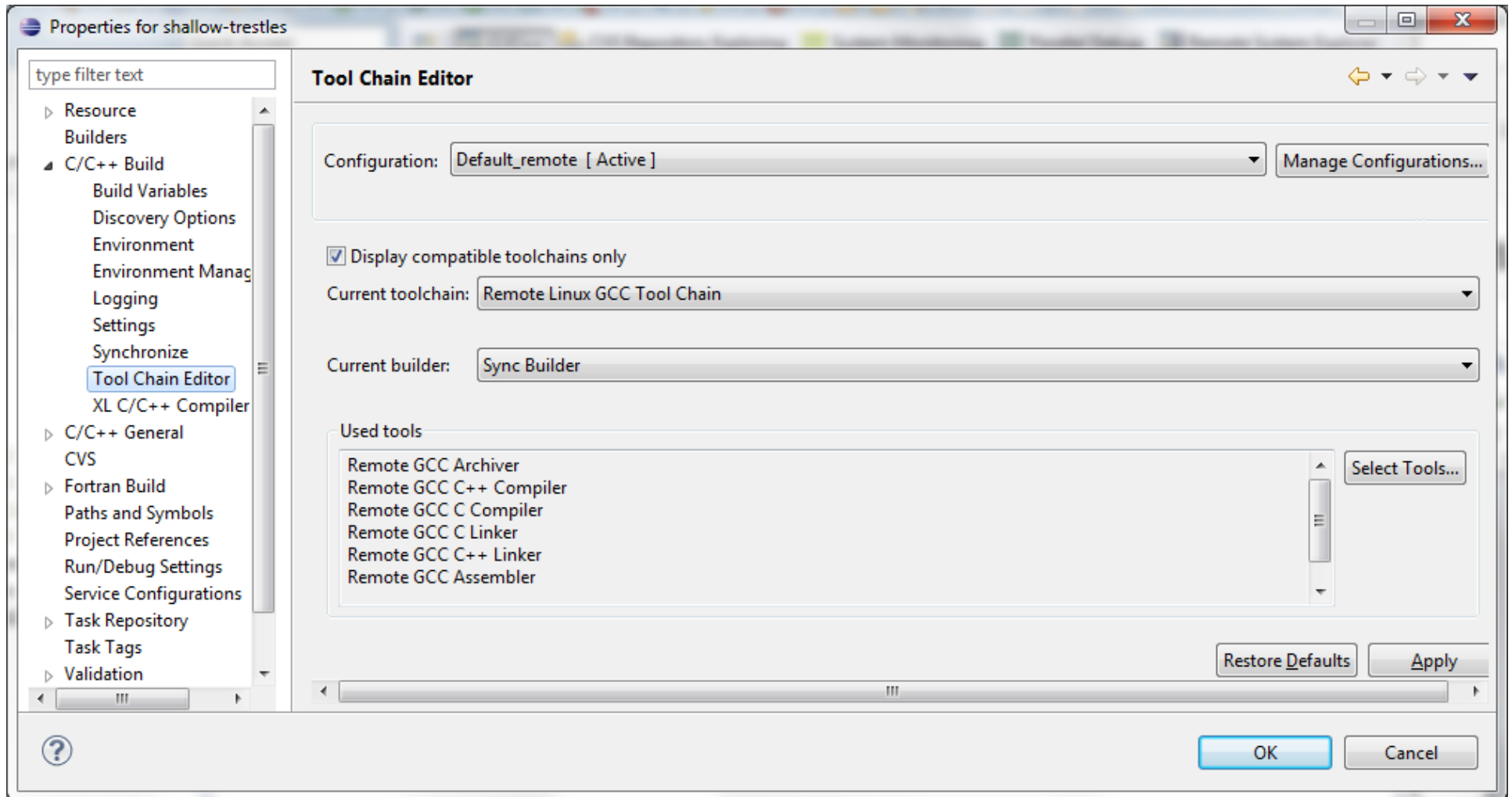
```
# cray ftn
#FFLAGS = -rm -s real64 -O3 -O fp3,cache3,scalar3,vector3 -h acc
# pgi ?
#FFLAGS = -acc -O2 -g -r8 -Minfo=acc,ccff
FFLAGS = -O2 -g -r8 -Minfo=acc,ccff
```

Side Note on synchronized projects: The sync. filtering probably needs tweaking because any large file beyond a few megabytes seems to break synchronization (making it take a very loooong time to synchronize). `core.*` should be added to the defaults and I'd like to see an option for excluding by size (ignore any file > 1M regardless of file type).

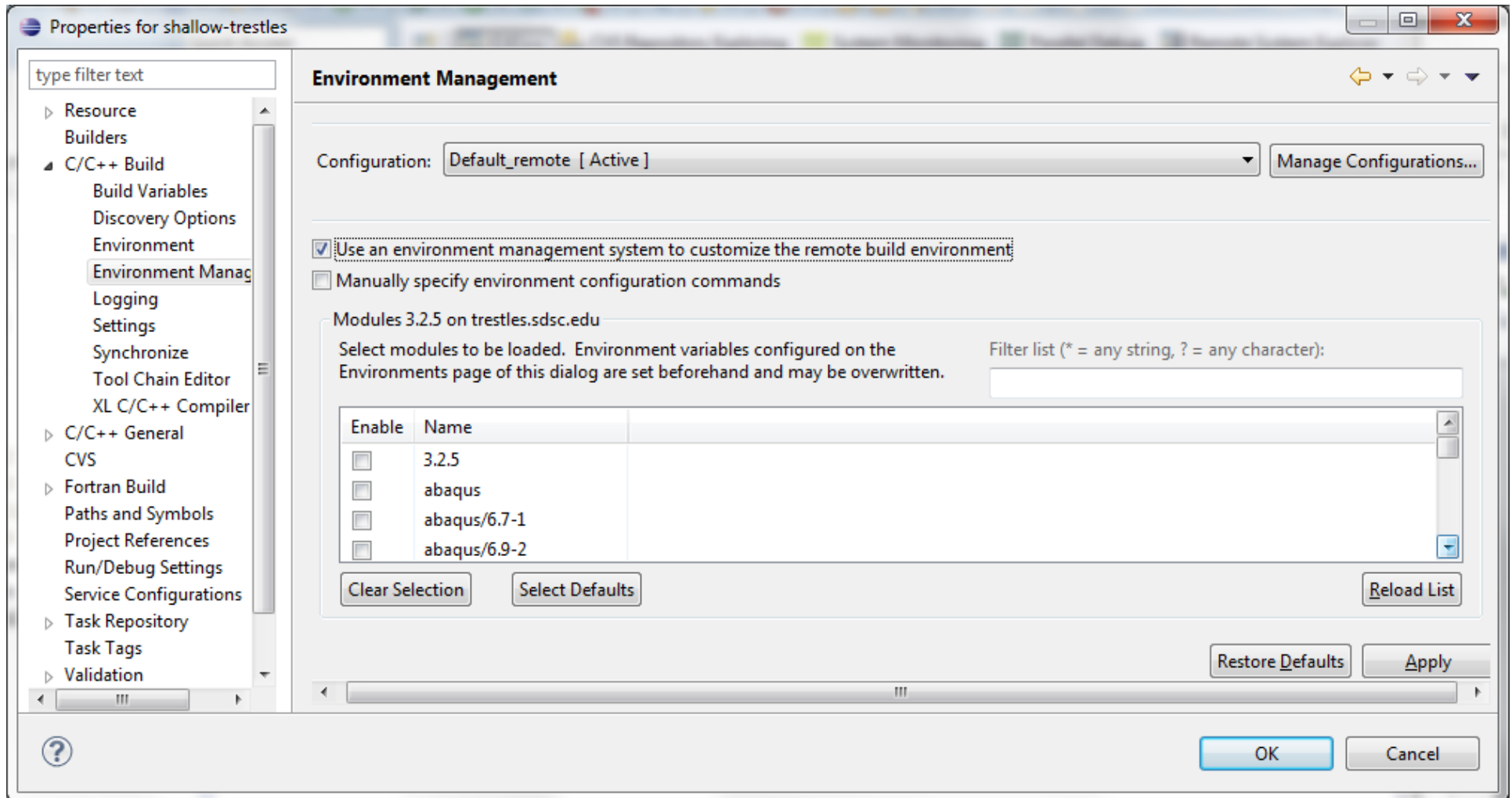
# Environment Management

- Current situation:
  - Multiple mechanisms for managing env
  - Local vs remote versus synchronized projects
  - Toolchain contribution
    - MPI include paths *in wrapper script (-I /path/to/include)*
  - Modules
    - Importance of ordering modules!
- Mileage does vary

# Toolchain interface



# Modules...



# Environment variables (populated by?)

The screenshot shows the 'Properties for shallow-trestles' dialog box, specifically the 'Environment' tab. The dialog is titled 'Properties for shallow-trestles' and has a search bar at the top left with the text 'type filter text'. The left sidebar contains a tree view with the following items: Resource, Builders, C/C++ Build (expanded), Build Variables, Discovery Options, Environment (selected), Environment Manag..., Logging, Settings, Synchronize, Tool Chain Editor, XL C/C++ Compiler, C/C++ General, CVS, Fortran Build, Paths and Symbols, Project References, Run/Debug Settings, Service Configurations, Task Repository, Task Tags, and Validation. The main area is titled 'Environment' and contains a 'Configuration:' dropdown menu set to 'Default\_remote [ Active ]' and a 'Manage Configurations...' button. Below this is a table titled 'Environment variables to set' with columns 'Variable', 'Value', and 'Origin'. The table contains the following data:

Variable	Value	Origin
CWD	C:\Users\jalameda\wor...	BUILD SYSTEM
HOME	/home/ux400689	BUILD SYSTEM
LM_LICENSE_FILE	/home/diag/pgi/license...	BUILD SYSTEM
LOGNAME	ux400689	BUILD SYSTEM
MAIL	/var/mail/ux400689	BUILD SYSTEM
PATH	/home/diag/opt/pgi/o...	BUILD SYSTEM
PWD	C:\Users\jalameda\wor...	BUILD SYSTEM
SHELL	/bin/bash	BUILD SYSTEM

Below the table are two radio buttons: 'Append variables to native environment' (unselected) and 'Replace native environment with specified one' (selected). To the right of the table are buttons for 'Add...', 'Select...', 'Edit...', 'Delete', and 'Undefine'. At the bottom right are 'Restore Defaults' and 'Apply' buttons. At the very bottom are 'OK' and 'Cancel' buttons.

# Managing Multiple Machine Builds

- Using CDT Build Configurations
- How to switch machines
- What works, what doesn't work
- How modules confounds the situation
- What abstraction makes sense to you

