Synchronized Project Filtering, Environment Management, and Multi-Machine Builds

Jay Alameda

Eclipse Parallel Tools Developer Workshop

19 September 2012

Agenda

- Define problem, look for solution
  - Constrain scope as much as possible
- Synchronized Project Filtering
- Environment Management
- Multi-Machine Builds

Synchronized Project Filtering

- Problem: synchronizing “large” files causes problems (very slow, or never completes)
- Questions:
  - Is there an identifiable size threshold?
  - Is this only binary, or text as well, or collections of files?
- Issues:
  - How does filtering work today (regex-based)
  - Is size based feasible? Why? Why not?
  - What are the scalability issues associated with filtering?

Environment Management

- Problem Statement:
  - Include paths are not (often) properly set automatically
    - Manual methods do exist, but can be troublesome
  - 3 views of environment are not in sync
    - "Eclipse view" – what Eclipse thinks the environment is
    - Remote (synchronized) build environment
    - Remote (run) environment
      - Batch
      - Interactive
      - Interactive batch

Environment Management (2)

- Questions: for build environment
  - What is the baseline condition for environment for local, remote, and remote synchronized projects?
  - Include Paths: what is the baseline condition for environment for include paths for local, remote, and remote synchronized projects?
  - How do other factors (remote or local toolchains, environment management system) impact either environment or include paths?

Environment Management (3)

- Questions: for run environment
  - What is the baseline condition for environment for *local, remote interactive, remote batch interactive and remote batch runs*?
  - Include Paths: not relevant for run environment?
  - How do other factors (remote or local toolchains, environment management system) impact environment for the four cases above?

Environment Management (4)

- Issues
  - Environment control mechanisms: modules, toolchains, (direct environment setting)
  - Relationship of control mechanisms with scanner discovery – how can we consistently ensure scanner/ discovery sets correct environment in Eclipse
  - Include files: set correctly when scanner/discovery handed good information, need to set for MPI/ openMP (MPI passes this in mpicc wrapper scripts)

Environment Management

- Issues: how to make progress
  - Toolchains: what do these give us, what do they set?
    - Should MPI flavors be implemented as a toolchain?
  - Modules
    - Need to set order of module loading?
    - Link modules at build time with modules at run time?
  - What relationship should there be between toolchains and modules?
  - Where do we need to establish environment (see 4 flavors of runtime?

Configuration Management

- Problem: several things (project, build configuration, machine selection, environment management) tied together in fashion that makes it difficult to manage the following use 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

Configuration Management (4)

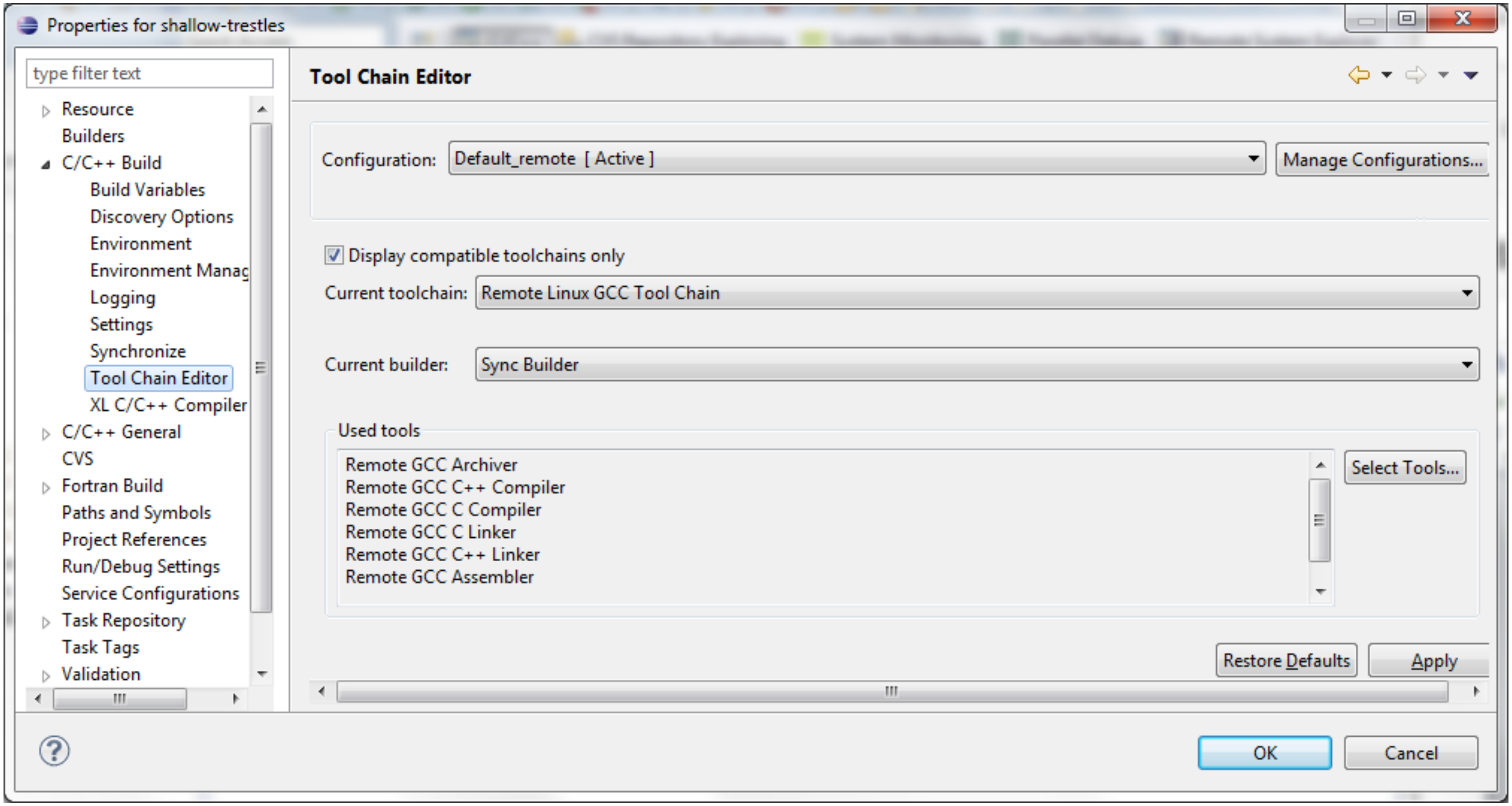
- Makefile excerpt:

```
# 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
```

Configuration Management (2)

- Questions:
  - Can we determine a way to pull these pieces apart?
    - Project, build configuration, machine selection, environment management
  - If so, can we then figure out a way to put the pieces together to solve the above problem (and other use cases) effectively?
- Issues:
  - What should the core component functionality be?
  - How close/what changes need to be made to achieve the core component functionality desired?
  - What changes are *feasible*, given the backdrop of functionality we are inheriting

Toolchain interface



Properties for shallow-trestles

type filter text

- Resource
- Builders
- C/C++ Build
  - Build Variables
  - Discovery Options
  - Environment
  - 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
- Validation

Tool Chain Editor

Configuration: Default_remote [Active]

Manage Configurations...

[✓] Display compatible toolchains only

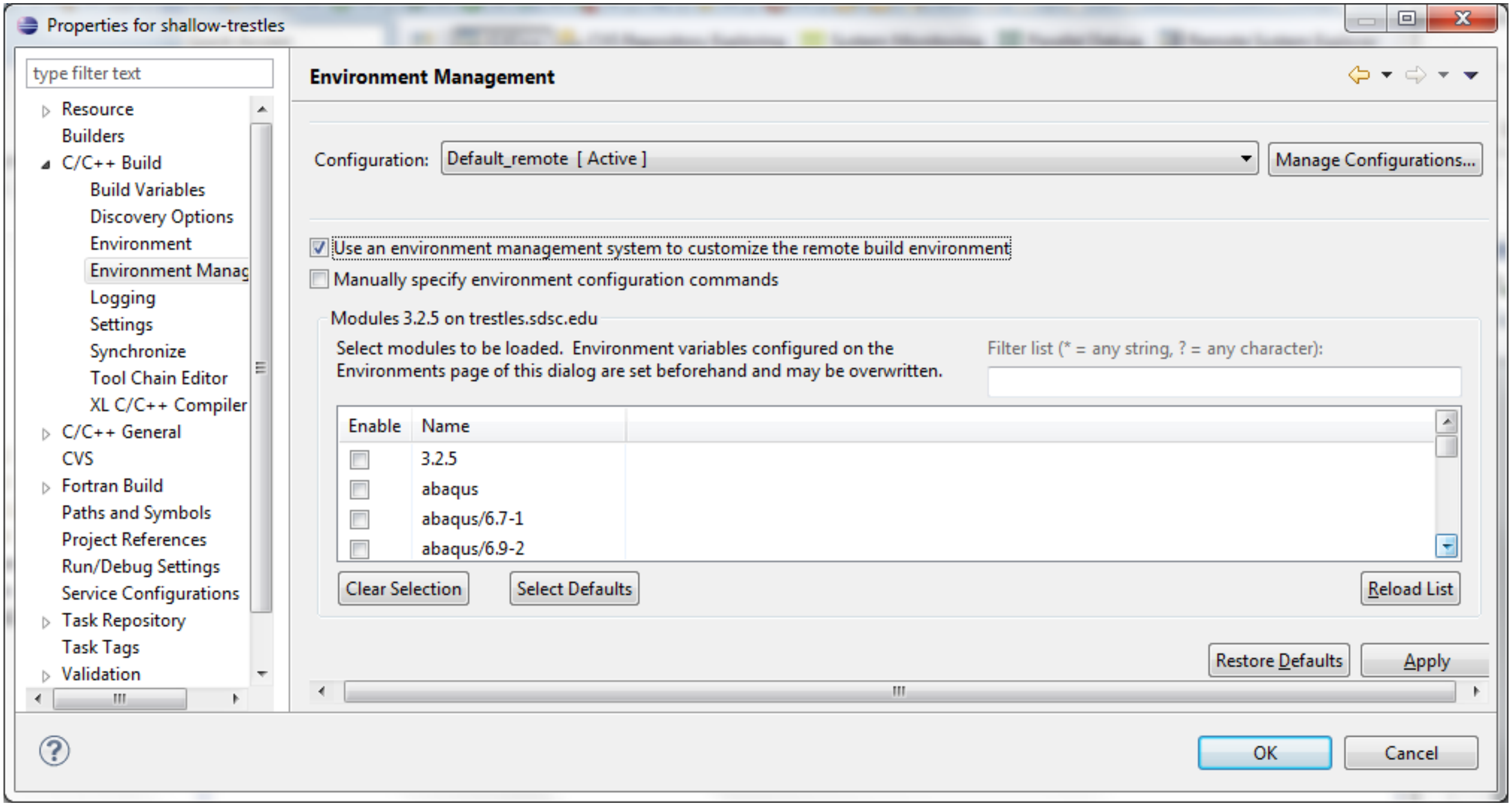
Current toolchain: Remote Linux GCC Tool Chain

Current builder: Sync Builder

Used tools

Remote GCC Archiver
Remote GCC C++ Compiler
Remote GCC C Compiler
Remote GCC C Linker
Remote GCC C++ Linker
Remote GCC Assembler

Modules…



Properties for shallow-trestles

type filter text

- Resource
- Builders
- C/C++ Build
  - Build Variables
  - Discovery Options
  - Environment
  - 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
- Validation

Environment Management

Configuration: Default_remote [Active] Manage Configurations...

[✓] Use an environment management system to customize the remote build environment

[ ] Manually specify environment configuration commands

Modules 3.2.5 on trestles.sdsc.edu

Select modules to be loaded. Environment variables configured on the Environments page of this dialog are set beforehand and may be overwritten.

Filter list (* = any string, ? = any character):

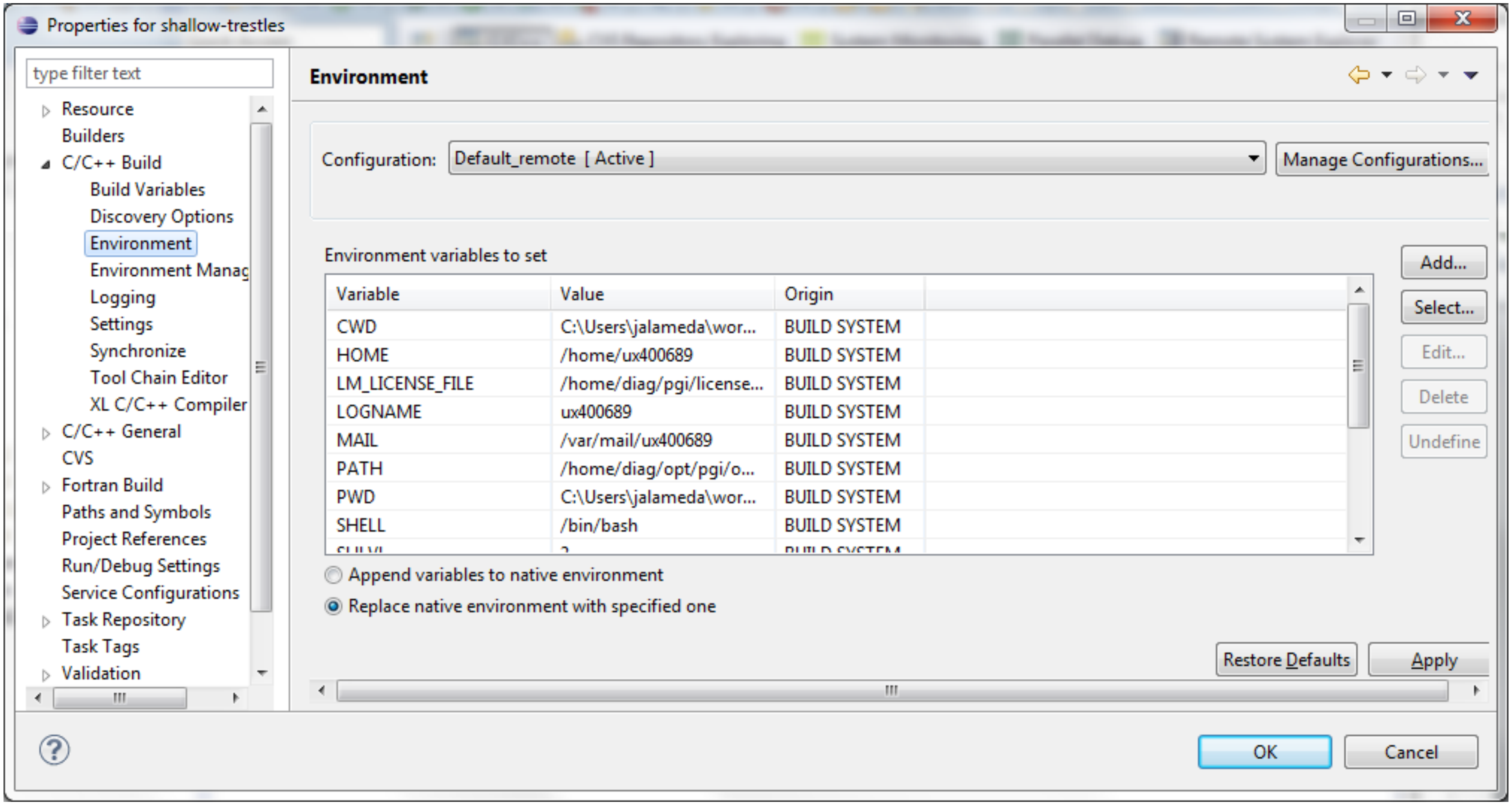
Enable	Name
☐	3.2.5
☐	abaqus
☐	abaqus/6.7-1
☐	abaqus/6.9-2

Clear Selection Select Defaults Reload List

Restore Defaults Apply

OK Cancel

Environment variables (populated by?)



Properties for shallow-trestles

Environment

Configuration: Default_remote [Active]

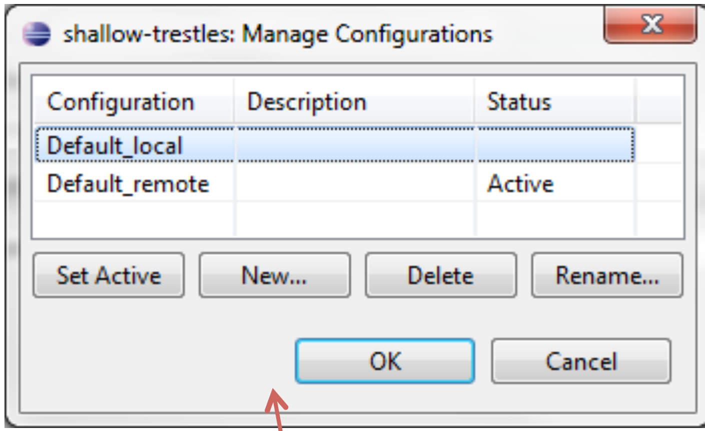
Environment variables to set

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

Append variables to native environment [✓ Replace native environment with specified one]

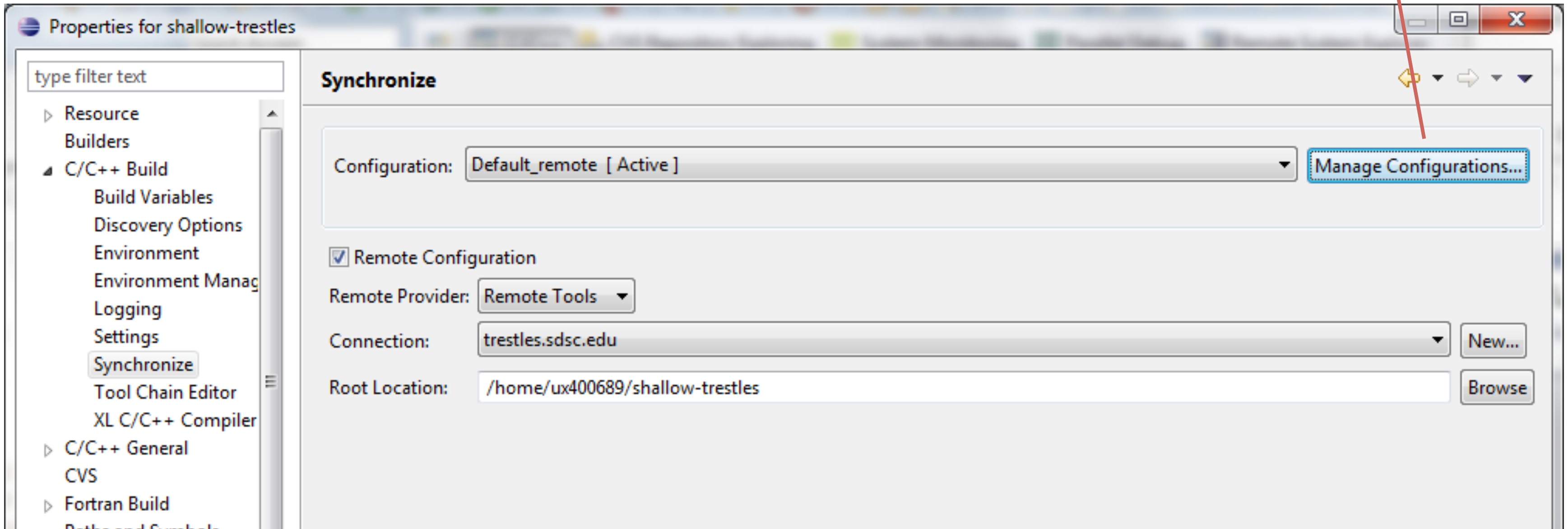
Managing Multiple Machine Builds

- 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



shallow-trestles: Manage Configurations

Configuration	Description	Status
Default_local		
Default_remote		Active



Properties for shallow-trestles

Synchronize

Configuration: Default_remote [Active]

☑ Remote Configuration

Remote Provider: Remote Tools

Connection: trestles.sdsc.edu

Root Location: /home/ux400689/shallow-trestles