

Distributed Builds

Definition

We need the ability to distribute work to:

- process multiple independent builds beyond the capacity of a single server's processing power
- execute builds on multiple different platforms
- retain a unified view of all project builds

Requirements

A. Distribution of work

1. Master Agent - controls build via xml-rpc
 - a. slaves need to be added to the master server
 - b. check availability of slaves
 - before build
 - timeout (to handle cases when slave goes down - build error)
 - c. distribute work
 - available slave will get build task from the queue
 - build
 - pass current configuration
 - expect result (master)/return result (slave)
 - attach each result returned by the slave to each project in the group (add to database)
2. Slave Agent
 - configurable in continuum.xml
 - queried for available installations
 - enqueue build, stop build
 - returns results

B. UI/Configuration

1. Aggregated view of distributed projects and projects in queue
2. Slave server list, add, edit, delete (new pages)
3. Slave configuration stored and editable in XML file

Limitations

1. credentials (s.a. svn credentials) are passed along if specified, but if server cache is used it will need to be done individually on the slaves
2. project dependencies are not considered in distribution - interdependent projects should target the same slave server
3. only system administrator can enable/disable distributed builds
4. dispatch project build to any slave available. cannot assign specific slave to a project group.
5. no tracking of scm changes

Future Enhancements

1. Remote builders
 - Builders will can be installed on some remote machines, a continuum manager will send actions to run to builders. An action can be something to run on all builders, on some of them or eventually only to an available builder if we don't want to run more than one build. Actions can be sent with JMS and builders will can apply some filters if they don't want to receive all actions. With that, we'll can do some parallel builds but the dependency tree must be respected for the build order. To work correctly with dependencies, each builders must use a central local repository. Maybe we can use an internal Archiva.
 - With Continuum builders configured to receive all commands, users will can run multi-platform build for each build definition execution
 - With Continuum builders configured to receive only some project types, users will can use a different builder by project group. In this case, the build of all projects will be done quickly because commands are balanced on few servers
 - With Continuum builders configured to build something when it is available, users will can installed few builders on several machine to balance the charge. In this case, it will be possible to run some parallel builds.
 - When the builder work will be done, a message will be sent to the manager to notify the end of the process.
 - With JMS used for the communication, we'll can add some listeners to create reports/statistics, log some informations...
2. Policy-based distribution
 - next available
 - load balanced
 - target environment matching