Gestão de Configuração e Mudanças de Software

Allan Lima

Arquitetura, Design e Implementação de Sistemas para Internet
Pós Graduação
Faculdade 7 de Setembro

arglbr@gmail.com

nov/2015

Ferramentas de Build

Índice

- Histórico
- Automação avançada
- Vantagens
- Tipos
- Script de build

Histórico

- Automação de Build é o ato de criar scripts ou automatizar uma grande variedade de tarefas que os desenvolvedores de software fazem no seu dia-a-dia, tais como:
  - Compilação de código fonte em código binário
  - Empacotamento do código binário
  - Executar Testes
  - Implantação de sistemas em produção
  - Criação de documentação e/ou notas de lançamento

Histórico

- Historicamente, os desenvolvedores chamam os compiladores e linkers de dentro de um script de construção ao invés de chamar o compilador pela linha de comando
- É simples usar a linha de comando para compilar um único arquivo fonte e depois chamar um linker para criar o objeto final, no entanto, ao tentar compilar e ligar muitos arquivos de código fonte, em uma ordem específica, usar o processo de linha de comando não parece ser uma solução razoável

Histórico

- A linguagem de script chamada `make` ofereceu uma alternativa melhor
  - Permite construir um script com os comandos necessários para compilar um aplicativo
  - O GNU make também ofereceu recursos adicionais, tais como `makedepend`, bem como a builds incrementais
- Este foi o início da automação de builds, onde seu foco principal foi automatizar as chamadas para os compiladores e linkers.

Histórico

- A medida que o processo de build tornou-se mais complexo, os desenvolvedores começaram a adicionar ações pré-compilação e pós-compilação, como o `check-out` do controle de versão para uma cópia de trabalho.

Automação Avançada

- O termo *builds distribuídas* significa que as chamadas reais para o compilador e linkers podem ser executadas em vários servidores para melhorar a velocidade da compilação
- Este termo é muitas vezes confundido com *processamento distribuído*
  - Processamento distribuído significa que cada etapa de um processo ou fluxo de trabalho pode ser enviado para uma máquina diferente para a execução.
  - Por exemplo, um dos passos da build pode exigir a execução de scripts de testes em várias máquinas

Automação Avançada

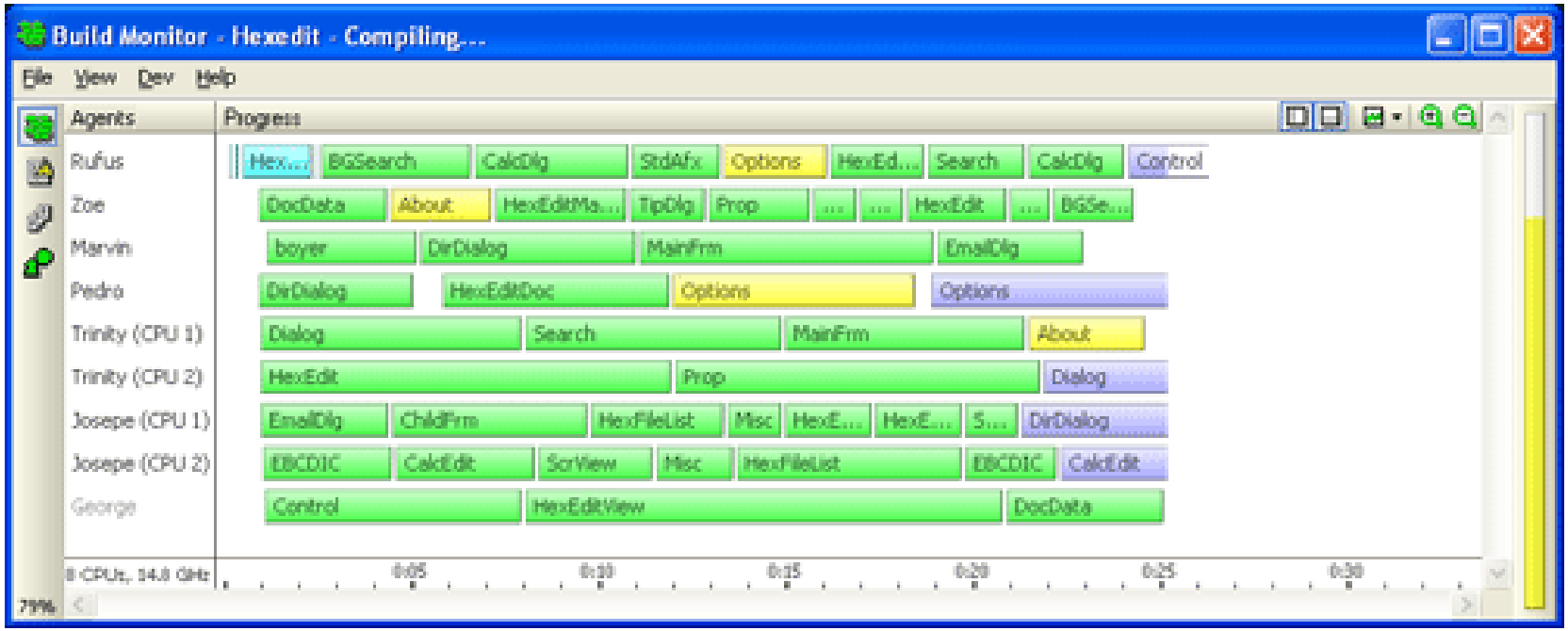
- O processo de build distribuída deve ter inteligência suficiente para compreender as dependências de código-fonte, a fim de enviar as diferentes etapas de compilação para máquinas diferentes
  - Deve ser capaz de gerenciar essas dependências, a fim de realizar compilações distribuídas
    - Executar a compilação de modo paralelo
    - Compilador pode ser chamado em modo *multi-threads* usando uma máquina que possui mais de um núcleo

Automação Avançada

- Nem todas as ferramentas de automação podem executar build distribuída
  - A maioria apenas fornece suporte ao processamento distribuído
  - A maioria das soluções dão suporte apenas a C/C++.
- Um exemplo de solução de build distribuída é o IncrediBuild Xoreax para a plataforma Microsoft Visual Studio

Automação Avançada

- IncrediBuild Xoreax:



Vantagens

- Melhora a qualidade do produto
  - Acelerar o processo de compilação e link
  - Eliminar tarefas redundantes
  - Minimiza as *bad build*
  - Elimina a dependência de pessoa-chave
  - Possui histórico de versões e *releases* para investigar problemas
  - Poupa tempo e dinheiro (por causa das razões listadas acima)

Vantagens

- **On-Demand:** um usuário executa um script na linha de comando
- **Automação agendadas:** um servidor de integração contínua executa uma `nightly build`
- **Automação por evento:** um servidor de integração contínua executando uma build a cada commit de um sistema de controle de versão

Script de Build

- Uma forma específica de automação de build é a criação de scripts de build (Makefiles). Isto é conseguido através de ferramentas como:
  - GNU Automake
  - Cmake
  - Imake
  - Qmake
  - Apache Ant
  - Apache Maven
  - OpenMake Meister

Script de Build

- Requisitos **básicos** de um sistema de build
  - Processo de compilação incremental.
  - Build frequente durante a noite para detectar os problemas mais cedo.
  - Suporte à gerenciamento de dependência de código fonte.
  - Relatar que arquivos fonte deram origem à um determinado executável.
  - Construção rápida.
  - Relatórios sobre a construção, compilação e link.

Script de Build

- Requisitos **opcionais** de um sistema de build
  - Gerar notas de lançamento e outros documentos, como páginas de ajuda.
  - Construir relatórios de status.
  - Relatórios de aprovação ou reprovação de teste.
  - Resumo dos recursos adicionados / modificados / excluídos a cada nova compilação.