Dependencias de software e hardware

Dependências de hardware

Na issue OPENBUS-31 reportamos os testes realizados em diferentes plataformas para definir sobre quais o Openbus consegue executar.

Atualmente o Openbus já foi testado em:

1. Linux24g3
2. Linux26g4
3. Linux26g4_64
4. Linux26_64
5. Linux26_ia64
6. SunOS510x86
7. SunOS510
8. SunOS58

Os serviços básicos do Openbus não requisitam grande poder computacional. Mesmo máquinas antigas podem abrigar um barramento. Um dos pontos de maior atenção deve ser a latência de rede, visto que o barramento se constitui por serviços remotos que, provavelmente, serão a base para a interação entre muitas aplicações.

No Openbus, não existe nenhuma dependência intencional sobre recursos de *hardware*, entretanto o Openbus depende de bibliotecas fornecidas por terceiros. Essas bibliotecas, por sua vez, podem não ser compatíveis com algumas plataformas. Veja Dependências de software para mais detalhes.

Dependências de software

As dependências podem variar em duas categorias: execução e desenvolvimento. Na primeira, lista-se os *softwares* requisitados para se conseguir instalar um barramento em uma nova máquina. Na segunda, aqueles necessários para compilar o barramento e gerar um pacote de instalação.

Execução

1. Variáveis de ambiente do tecmake
   a. No Linux, o Tecmake **depende** do comando **gcc** para identificar sua versão e completar a variável TEC_UNAME !
2. lua 5.1
3. lualdap
   a. openldap
      i. sasl2
      ii. berkeleydb
4. lce
   a. openssl = 0.9.9
5. luuid
   a. uuid
6. luafilesystem
7. oil
8. loop

Alguns dos nossos **perfis** de pacotes incluem essas dependências (exceto: libc, libstdc++, gcc-runtime, tecmake).

Desenvolvimento

O conjunto de pacotes necessários pode variar de acordo com a plataforma e com o contexto do desenvolvedor.

- Utilitários mínimos
  1. Para uso das ferramentas de empacotamento:

        a. Cliente Subversion.
        b. Cliente HTTP (**wget** ou **curl**).
        c. Tecmake.
- Utilitários para desenvolvimento de serviços em C++
    1. ORB Orbix 6.3
    2. Especialmente, no caso do desenvolvimento em máquinas **Linux**:
        - se as libs de runtime do gcc forem <= 3.3, então deve-se instalar a versão **Orbix6.3-SP3gcc32**, mas se o runtime do gcc for >= 3.4 então deve-se instalar a versão **Orbix6.3-SP3gcc34**.
- Utilitários para desenvolvimento de serviços em Java
    1. Sun Java Virtual Machine 1.6
    2. ORB JacORB 2.3.0
- Utilitários para desenvolvimento dos serviços básicos e empacotamento
    1. Autotools (autoconf + automake). Exemplos: autoconf, autotools-dev, automake.
    2. Compiladores C e C++. Exemplos: gcc, g++, gcc-c++.
    3. Cabeçalhos da GNU libc. Exemplos: glibc-devel, libc6-dev.
    4. Cabeçalhos da biblioteca readline. Exemplos: libreadline5-dev, readline-devel.
    5. Cabeçalhos da biblioteca ncurses. Exemplos: libncurses5-dev, ncurses-devel.
    6. Ferramenta de
    7. Outras bibliotecas [1]:
        a. Cabeçalhos LDAP. Exemplos: libldap2-dev, ldap2-devel.
        b. Cabeçalhos SASL2. Exemplos: libsasl2-dev, sasl2-devel.
        c. Cabeçalhos BerkeleyDB. Exemplos: libdb4.6-dev, db4.6-devel.
        d. Cabeçalhos UUID. Exemplos: libuuid1-dev, uuid-devel.
        e. Cabeçalhos OpenSSL na versão 0.9.9.
        f. Cabeçalhos Lua na versão 5.1.

[1] *essas bibliotecas podem ou estar já instaladas na máquina ou serão instaladas durante o empacotamento*

Recomendações diante de problemas relacionados a versões incompatíveis

No ato da instalação, é comum acontecer do pacote (que tentamos instalar) não ser compatível binariamente com as versões das bibliotecas básicas disponíveis no sistema operacional de destino.
Esse problema acontece pois, nem sempre é possível reproduzir fielmente o estado da instalação da máquina destino em laboratório.

Portanto, o questionário abaixo deve ser respondido e enviado a equipe Openbus antes da geração do pacote. Assim reduzimos as chances de falha da instalação.

Informações mínimas

1. Qual o sistema operacional (nome + *release*)? Exemplos: Linux CentOS 5.2, Linux Fedora 9, Solaris 10 3/05 s10_74L2a.
2. Qual a arquitetura da CPU? 32 ou 64 bits? Exemplos: Intel 64 bits, Intel 32 bits, AMD 64 bits, Sparc 64 bits, Sparc 32 bis.
3. Caso a CPU seja 64 bits, o sistema operacional também é 64 bits? Exemplos: Não, a CPU é 64 bits, mas o sistema operacional é 32 bits.
4. Qual a versão do núcleo (*kernel*) do sistema operacional? Exemplos: Linux 2.6.28, SunOS 5.10.
5. Qual a versão da libc instalada? Exemplos: 2.5-24, 2.8-8, 2.9-6.
6. Qual a versão do compilador C e C++ instalados (em caso de mais de um, favor listar)? Exemplos: GNU C/C++ (gcc) 4.1.2, GNU C/C++ 3.4, GNU C/C++ 2.95, Sun C Compiler 5.10.
7. Qual a versão da máquina virtual Java? Exemplos: GNU Java Compiler (gcj) 1.4.2, Sun JavaVM 1.5u13.

Informações adicionais

Em geral, o **empacotador** deverá gerar os pacotes de instalação incluindo todas as bibliotecas das quais o Openbus depende, como OpenLDAP e OpenSSL. Mas também é possível distribuir o **apenas** Openbus, sem suas dependências. Para tanto assumimos que essas dependências estejam instaladas no sistema. Logo, precisamos saber de outras informações, conforme listado abaixo.

1. Qual a versão da biblioteca OpenLDAP instalada? Exemplos: 2.4.11, 2.3.7, 2.1.10.
    a. Possui suporte a LBER?
    b. Possui suporte a conexões seguras por TLS/SSL?
2. Qual a versão da biblioteca OpenSSL instalada? Exemplos: 0.9.7, 0.9.8, 0.9.9, 1.0.0.
3. Qual a versão da bilbioteca SASL2 instalada? Exemplos: 2.1.22 .
4. Qual a versão da bilbioteca BerkeleyDB instalada? Exemplos: 4.6 .