系统环境：CentOS6.4  linux内核版本2.6

**一、 必要工具软件安装**

软件版本：gcc (GCC) 4.4.7 ，GNU Make 3.81

重装系统后需要安装GCC开发工具，否则下述软件将无法安装。推荐使用yum进行安装。

安装gcc：yum -y install gcc

安装g++：yum -y install gcc-c++

安装make：yum install make

另外，对于Dell PowerEdge服务器启动有时会显示“SATA Port X device not found. Strike the F1 key to continue,F2 to run the System Setup Program!!”。出现这种情况是由于服务器开启了X硬盘或软盘，但是机器没有相应的硬件。在BIOS中关闭相应的Port即可。按F2进入BIOS，按↑，↓键选择“SATA Settings”，“回车”进入后选择到相应的Port，按“空格”将Port的状态改为“Off”。

**二、 Log4CXX安装**

软件版本：apache-log4cxx-0.10.0.tar.gz，apr-1.4.8.tar.gz，apr-util-1.5.2.tar.gz

*原文链接*[*http://yaocoder.blog.51cto.com/2668309/980276*](http://yaocoder.blog.51cto.com/2668309/980276)

**下载解压**  
<http://logging.apache.org/log4cxx/>， <http://apr.apache.org/>  
apr-1.4.8.tar.gz,  apr-util-1.5.2.tar.gz,  apache-log4cxx-0.10.0.tar.gz  
  
1.tar zxvf apr-1.4.8.tar.gz  
2.tar zxvf apr-util-1.5.2.tar.gz  
3.tar zxvf apache-log4cxx-0.10.0.tar.gz  
  
  
**编译安装**  
**首先安装apr-1.4.8**，切换cd apr-1.4.8，配置./configure --prefix=/usr/local/apr，接着make， make install  
  
**接着安装apr-util-1.5.2**，切换至cd ../apr-util-1.5.2， ./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr，接着make，make install；  
  
**最后安装apache-log4cxx-0.10.0**，切换cd ../apache-log4cxx-0.10.0，配置./configure --prefix=/usr/local/ --with-apr=/usr/local/apr --with-apr-util=/usr/local/apr-util  
**注意配置前需进行以下操作：**  
1.vim src/main/cpp/inputstreamreader.cpp  
增加#include <string.h>；  
#include <log4cxx/logstring.h>  
#include <log4cxx/helpers/inputstreamreader.h>  
#include <log4cxx/helpers/exception.h>  
#include <log4cxx/helpers/pool.h>  
#include <log4cxx/helpers/bytebuffer.h>  
+  
#include <string.h>  
+  
否则会出现inputstreamreader.cpp:66: error: 'memmove' was not declared in this scope  
make[3]: \*\*\* [inputstreamreader.lo] 错误 1  
  
2.vim src/main/cpp/socketoutputstream.cpp  
增加#include <string.h>；  
#include <log4cxx/logstring.h>  
#include <log4cxx/helpers/socketoutputstream.h>  
#include <log4cxx/helpers/socket.h>  
#include <log4cxx/helpers/bytebuffer.h>  
+  
#include <string.h>  
+  
否则会出现socketoutputstream.cpp:52: error: 'memcpy' was not declared in this scope  
  
  
3.vim src/examples/cpp/console.cpp  
增加#include <string.h>，#include <stdio.h>；  
+  
#include <stdio.h>  
+  
#include <stdlib.h>  
+  
#include <string.h>  
+  
#include <log4cxx/logger.h>  
#include <log4cxx/consoleappender.h>  
#include <log4cxx/simplelayout.h>  
#include <log4cxx/logmanager.h>  
#include <iostream>  
#include <locale.h>  
否则会出现  
console.cpp: In function ‘int main(int, char\*\*)’:  
console.cpp:58: 错误：‘puts’在此作用域中尚未声明

**三、 Libevent安装**

软件版本：libevent-2.0.21-stable.tar.gz

1.解压libevent-2.0.21-stable.tar.gz

tar –zxv –f libevent-2.0.21-stable.tar.gz

2.编译安装

cd libevent-2.0.21-stable

./configure --prefix=/usr/local

其中—prefix指定安装路径，指定为/usr/local后则将头文件安装到/usr/local/include下，库文件安装到/usr/local/lib下。

make

make install

**四、 Libjson安装**

软件版本：libjson\_7.6.1.zip

1.解压libjson\_7.6.1.zip

unzip libjson\_7.6.1

2.编译安装

make

编译完成后将libjson.a复制到/usr/local/lib下

cp libjson.a /usr/local/lib

将libjson.h和JSONOptions.h 复制到/usr/local/include/libjson/下

cp libjson.h JSONOptions.h /usr/local/include/libjson/

将\_internal/Source复制到/usr/local/include/libjson/下

cp -r Source/ /usr/local/include/libjson/

**五、 mysql-connector/C++安装**

软件版本：mysql-connector-c++-1.0.5.tar.gz

推荐使用源代码编译。使用发布的二进制库不稳定。

1.解压mysql-connector-c++-1.0.5.tar.gz

tar zxv –f mysql-connector-c++-1.0.5.tar.gz

2.编译安装

编译mysql-connector-c++需要cmake，如果系统没有安装cmake则安装：

yum install cmake

cd mysql-connector-c++-1.0.5

生成Makefile：

cmake -DCMAKE\_REQUIRED\_FLAGS=-xO4 -DCMAKE\_INSTALL\_PREFIX=/usr/local

此时执行make会出现错误。

1.对于nt8\_t、uint8\_t、int16\_t、uint16\_t,等等未定义的情况，解决方法是在cppconn/config.h中把

#ifdef HAVE\_STDINT\_H  
#include <stdint.h>  
#endif

改为：

//#ifdef HAVE\_STDINT\_H  
#include <stdint.h>  
//#endif

2.对于strtoll、strtoull未定义的情况，解决方法是在cppconn/config.h中加入

#define HAVE\_FUNCTION\_STRTOLL  
#define HAVE\_FUNCTION\_STRTOULL

3.mysql\_art\_resultset.cpp等出现sprintf未定义等错误，解决方法是在相应错误的cpp文件中加入：

#include <cstdio>

进行编译安装：

make

make install

进行安装。

**六、 boost安装**

软件版本：boost\_1\_42\_0.tar.gz

1.解压boost\_1\_42\_0.tar.gz

tar zxv –f boost\_1\_42\_0.tar.gz

2.编译安装

cd boost\_1\_42

sh bootstrap.sh

执行脚本后将生产bjam程序。

./bjam release install

系统默认会将include拷贝到/usr/local/include/boost/中，将lib拷贝到/usr/local/lib下。

**七、 Redis及hiredis安装**

软件版本：redis-2.4.18.tar.gz

安装Redis

1.解压redis-2.4.18.tar.gz

tar –zxv –f redis-2.4.18.tar.gz

2.编译安装

cd redis-2.4.18

make

make test

如果make test提示没有tcl错误，则安装tcl

yum install tcl

make install

安装hiredis

1.解压

tar –zxv –f hiredis.tar.gz

2.编译安装

cd hiredis-master

make clean

make

make intall

**八、 服务器的配置**

**1. 配置iptables**

对于业务服务器，需要配置iptables，开启相应的端口。

目前需要开启的端口有ssh(22), ftp(21), mysql(3306), logicDispatch进程(12002), imServer进程(12004),imDevServer进程(12008), mailSync进程(6001)。

在/etc/sysconfig/iptables文件中添加如下规则。

#ssh

-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT

# ftp

-A INPUT -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT

#mysql

-A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT

#[myapp:logicDispatch](http://myapplogicdispatch/)

-A INPUT -m state --state NEW -m tcp -p tcp --dport 12002 -j ACCEPT

#[myapp:imServer](http://myappimserver/) and imDevServer

-A INPUT -m state --state NEW -m tcp -p tcp --dport 12004 -j ACCEPT

-A INPUT -m state --state NEW -m tcp -p tcp --dport 12008 -j ACCEPT

#[myapp:mailSync](http://myappmailsync/)

-A INPUT -m state --state NEW -m tcp -p tcp --dport 6001 -j ACCEPT

添加完后重启iptables服务

service iptables restart

使之生效。

**2. 开启core dump**

打开/etc/profile文件。

vi /etc/profile

在文件最后添加

#open core dump

ulimit -S -c unlimited > /dev/null 2>&1

表示打开core dump，对dump文件的大小无限制。

然后source /etc/profile 使之生效。

**3. 配置最大文件打开数**

1. 查看服务器的系统级打开文件数限制。

cat /proc/sys/fs/file-max

当前业务服务器该数值为384469。

这表明这台Linux系统最多允许同时打开(即包含所有用户打开文件数总和) 384469个文件，是Linux系统级硬限制，所有用户级的打开文件数限制都不应超过这个数值。通常这个系统级硬限制是Linux系统在启动时根据系统硬件资源状况计算出来的最佳的最大同时打开文件数限制，如果没有特殊需要，不应该修改此限制。

1. 修改用户级最大打开文件数

使用如下命令查看最大打开文件数。

ulimit –Hn    (硬限制)

ulimit –Sn   （软限制）

在profile文件中添加ulimit修改。

vi /etc/profile

在文件最后添加

#add the max filefds

ulimit -SHn 384469

然后source /etc/profile 使之生效。

**4.****MySQL数据库服务器配置空闲连接超时时间**

（MySQL 8小时问题）MySQL 的默认设置下，当一个连接的空闲时间超过8小时后，MySQL 就会断开该连接。所以此问题会造成业务进程一段时间（超过8小时）没有数据库操作后进程崩溃。将MySQL连接空闲超时时间改大可以避免此问题。

修改my.cnf, 在[mysqld]后面添加上：

# MYSQL 8 Hours Time out Problem

wait\_timeout=2147483

interactive-timeout=2147483

重启MySQL服务。

service mysql restart

登录MySQL，可以查看这两个参数：

mysql> show variables like '%timeout%';

**九、 业务程序编译出现的问题**

1.编译业务进程需要安装MySQL-shared-5.5.34-1.el6.x86\_64.rpm，MySQL-shared-compat-5.5.34-1.el6.x86\_64.rpm，MySQL-devel-5.5.34-1.el6.x86\_64.rpm。

rpm –ivh MySQL-shared-5.5.34-1.el6.x86\_64.rpm

rpm –ivh MySQL-shared-compat-5.5.34-1.el6.x86\_64.rpm

rpm –ivh MySQL-devel-5.5.34-1.el6.x86\_64.rpm

2.编译业务程序时出现“uuid/uuid.h: No such file or directory”，需要安装uuid-devel和libuuid-devel。

yum install uuid-devel  
yum install libuuid-devel