掌握這個小技巧，讓你的C++ 編譯速度提升50 倍!

跟田老師學C++ 昨天

以下文章來源於高效程序員 ， 作者Waleon

高效程序員

聚焦程序人生，踐行終身成長。專注分享IT 技術「C++/Python/Linux/Qt 等」、學習資料、職場經驗、熱點資訊，有趣、好玩、靠譜！（...

隨著C++ 項目的持續擴大，編譯效率越來越是一個問題了。想一想你每天花在這上面的時間，再乘以團隊成員的個數，是不是成本很高?

那有沒有什麼辦法，在不需要修改源碼，也不更換硬件的情況下提升效率呢？一起來看看下面這幾個方法，足以讓你的編譯速度飛起來。

先隨便下載一個第三方源碼，例如：spdlog，我們來測試一下整個編譯需要多久：

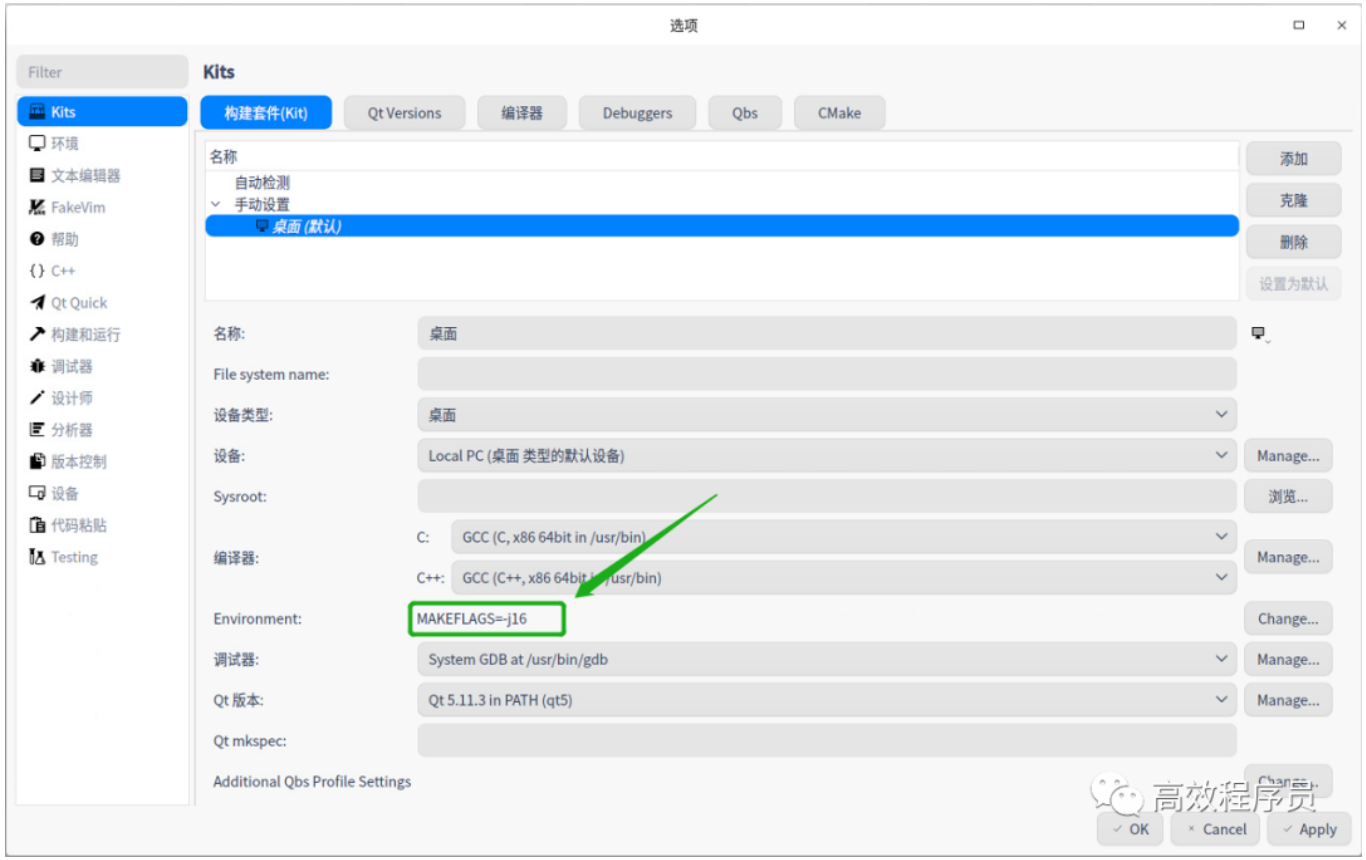
源碼很少，但卻用了00:48，速度著實不給力，來優化一下吧!

●1● 啟用多核編譯

可以開啟多核編譯來提高編譯速度，充分利用機器的性能來優化編譯。

打開Qt Creator，選擇【Kits】->【構建套件（kit）】，在【Environment】處輸入“MAKEFLAGS=-j16”。

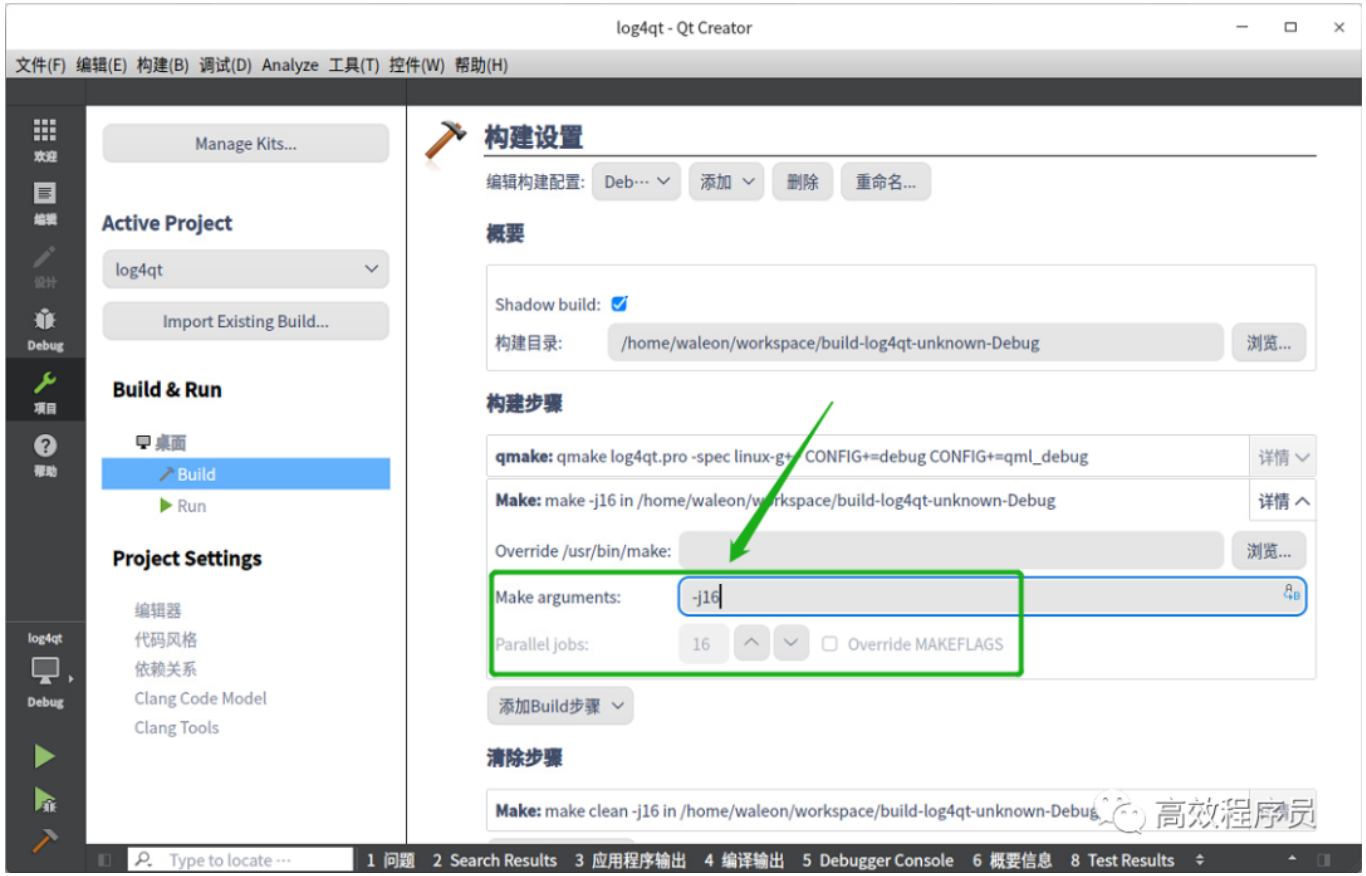
這是最便利的方式，一勞永逸，只需設置一次，後期所有使用make 的構建系統都會自動啟用。



這裡的數字大小，需根據電腦的CPU 核心數和線程數來設置，假如是8 核16 線程，建議設置16。

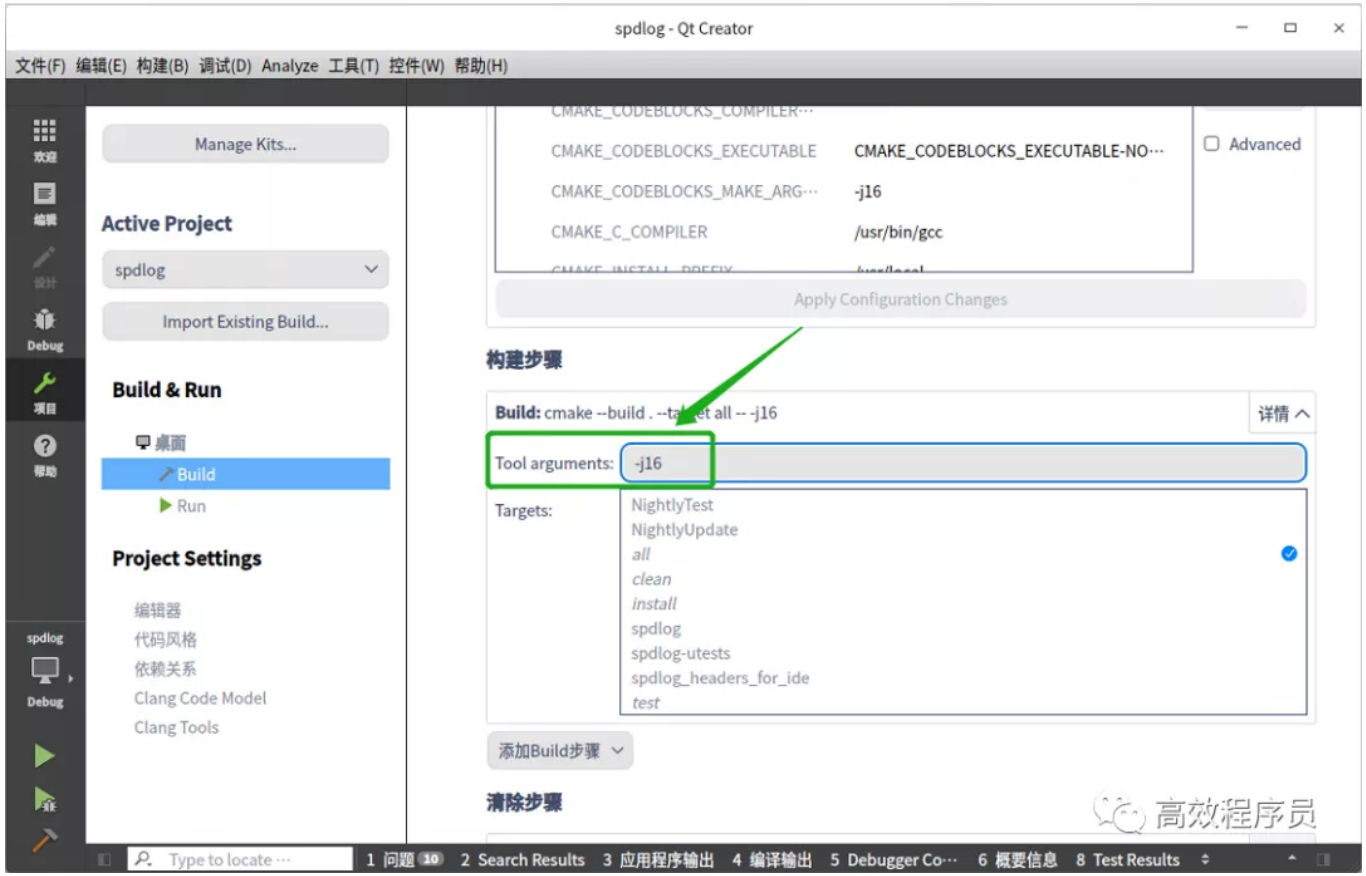
qmake 設置

如果只想应用于某个特定项目，选择【项目】->【构建步骤】->【Make】，点击右侧的【详情】按钮，在【Parallel jobs】或者【Make arguments】处设置并行工作线程的个数，这两个效果是一样的：



cmake 设置

和 qmake 类似，只不过 cmake 的设置在【Tool arguments】处：



效果

设置完成之后，重新编译一下：

优化到了 00:10，怎么样，提升了不少吧。

2

使用 ccache 编译器缓存

ccache（全称：compiler cache）是一个编译器缓存，该工具会高速缓存编译生成的信息，并在编译的特定部分使用高速缓存的信息，比如头文件，这样就节省了通常使用 cpp 解析这些信息所需要的时间。

- ccache 主页：https://ccache.dev/
- 文档地址：https://ccache.dev/documentation.html
- GitHub 源码：https://github.com/ccache/ccache

安装 ccache

要安装 ccache，执行以下命令：

```
$ sudo apt install ccache
```

qmake 设置

打开 .pro，添加以下配置，ccache 就可以工作了：

```
QMAKE_CXX = ccache $$QMAKE_CXX
```

从 Qt 5.9 开始，有一个更简单的方式：

```
load(ccache)
```

cmake 配置

在 CMakeLists.txt 中添加以下配置，将 ccache 作为编译命令和链接命令的启动器：

```
find_program(CCACHE_FOUND ccache)
if(CCACHE_FOUND)
    set_property(GLOBAL PROPERTY RULE_LAUNCH_COMPILE ccache)
    set_property(GLOBAL PROPERTY RULE_LAUNCH_LINK ccache)
endif(CCACHE_FOUND)
```

效果

配置完成之后，再来测试一下：

简直吊炸天 - 00:01，从最初的 48 秒优化到了仅需 1 秒，效率提升了快 50 倍，这速度 6 的飞起！

END

欢迎一起交流C++开发

请扫描下方二维码加田老师为微信好友

欢迎关注“跟田老师学C++”微信公众号

请手指长按下方二维码图片识别，即可关注

欢迎加入“C++/Qt”知识星球

請手指長按下方二維碼圖片識別，即可加入

記得幫我點贊/在看哦

閱讀原文

喜歡此內容的人還喜歡

通過Handle理解V8的代碼設計（基於V0.1.5）

編程雜技

超簡單Python 漢字拼音轉換工具

Python實用寶典

Python 自動識別圖片文字—OCR實戰教程

Python實用寶典