**CUDADATRIE说明文档**

李雨前 yuqianfly@gmail.com

华中科技大学计算机学院 智能与分布式计算实验室

**目录：**

**0 申明**

**1 CUDADATRIE简介**

**2 CUDADATRIE算法说明**

**3 CUDADATRIE 代码结构**

**4 CUDADATRIE 使用向导**

**5 CUDADATRIE扩展应用**

**6 参考**

**7 致谢**

**0申明：**

**本作品系原创，知识产权归华中科技大学计算机学院“智能与分布式计算实验室”所有。除华中科技大学计算机学院“智能与分布式计算实验室”外，任何个人、团队、研究结构、商业单位等不能对本并行分词系统进行专利申请或者撰写本并行分词系统算法的论文。任何个人、团队、研究结构、商业结构可以对本算法进行使用、修改、扩展、传播。**

**使用本算法不当而造成的损失，责任有使用者自行负责。**

1. **CUDADATRIE简介**

基于CUDA平台并行中文分词，简称CUDADATRIE。作品词库大小：5.1MB，共计24万词条，词典加载后占内存约3MB，实际有效空间利用率约75%。作品采取基于词典的分词思路，借助双数组检索树来组织词条，适合海量文本快速分词、词频统计、文本聚类等自然语言和数据挖掘应用，具有很强的扩展性。

1. **CUDADATRIE算法说明**

作品中非常重要的一个数据结构就是Trie树，又名检索树，实质是有限状态机。如下图1所示。



图1 中文trie示意图

图1是由“共和国”、“共和党”、“华人”、“华裔”、“人民共和国”、“人民共和党”、“人民共产党”、“万岁”、“万家灯火”、“中华人民共和国”构成的一颗trie树。绿色的节点表明是一个词条的结束。比如，输入“中华人民共和国”，只需扫描一篇待分词字符串，就可以得到分词：“中华”“中华人民”“中华人民共和国”，复杂度为O(n)，n为待分词字符串长度。

图2是基于CUDA平台的并行分词系统框架图



图2 并行分词系统框架图

并行分词采取基于词典的分词思路，使用Trie树结构组织词条。真正移植词典到GPU内存，以支持在GPU特定硬件和软件环境下并行分词，需要将图1所示的树进行双数组化[1~2]. 这是并行分词的前提，同时对中文词条构建有效的双数组，需要深入研究中文词库特征。下面简要描述作品是如何实现并行分词的。

1 中文词典构造：读取词库词条（24万），构建双数组Trie树（空间利用率75.4%）

2 加载双数组检索树到GPU显存

3加载一批文档到CPU内存

4 每个文档对应一个block进行分词

5 每个线程独立保存其分词结果

6 分词结果写回磁盘

对应图2，Base、Check、Status构成双数组树，Term0 、Term1等数组对应每个线程分词的结果。每篇文档对应一个block，文档平均分配到每个线程。线性数量、文档数量、文档大小、硬件条件等对分词吞吐率都有影响。

作品中，实现了三种分词方法：正向最大匹配、正向最小匹配、全匹配。例如：中华人民共和国万岁数。正向最大匹配分词结果：中华人民共和国、万岁；正向最小匹配分词结果：中华、人民、共和国、万岁；全匹配分词:中华、中华人民、中华人民共和国、华人、人民、人民共和国、共和国、共和国万岁、万岁。详细算法请看源代码。

1. **CUDADATRIE 代码结构**

作品代码主要包括：词典加载、词典拷贝、批量文档加载、批量文档拷贝、正向最大、最小、全匹配分词算法、批量分词结果写磁盘。总代码量约50k。另外，inputFiles是待分词文档，outputFiles是分词输出和控制台日志信息，Library是词库。CUDAdaTrie.cu CUDAdaTrieOpt.cu分别是优化前和优化后的内核代码。

作品代码没有公开词典构造的详细代码，相关算法可以参考[1~3], 随作品还附带一个已建好的词典和小规模测试文本，直接加载和使用。关于代码的其它任何疑问请联系作者。

1. **CUDADATRIE 使用向导**

作品在windows VS2008上编辑编译调试进行，GPU卡型号：GTS250 ，1G显存，128 core

在编译时需要加大堆栈空间支持，建议为10MB。Vs向导堆栈空间修改方法：右键点击工程，选择“属性”，弹出对话框，找到左边树形菜单的“链接”项，选择“系统”，对应右边“堆栈保留空间”，输入10240000

运行程序中，需要library、inputFiles、outputFiles文件支持，其中library是待加载的词库，inputFiles是待分词的批量文本，目前设置为96篇文档，应根据硬件条件调节批量文档数目。OutputFiles是保存分词结果和控制台输出信息。可以修改inputFiles里面的文档，但只支持txt文档，常规编码，单文档大小限制在12K以内。对应单篇大文档，需根据需求进行参数的调整。

随文档附带的ext文件如果运行，请确保对应的library 、inputFiles路径正确。如果无法运行，可以直接将文档中的源代码重新编译。在重新编译过程中，CUDAdaTrie.cu CUDAdaTrieOpt.cu只能使用其中一个，后者是优化后的版本。其它代码不需改动。

由于词典和测试文档加起来超过4MB，所以没有将整个代码工程一起打包。其它疑问请与作者联系。

**注意：**

1本作品公开的分词算法在精度上还没有进行优化，其精度完全在于词典本身。所以，请不要用于商业系统，任何使用不当造成的损失，作者和CUDA大赛方概不负责，责任由使用者自负。

2本作品并行分词是文档平均分配到线程，会导致边界词条异常，分词准确率会在数据结尾有异常。block线程少，出错的概率也少。有兴趣的读者，只稍加修改原程序就可以避免边界错误。作品中对停用词的处理，也是简单的，进一步的优化工作读者可自行完善。

3 本作品主要针对中文分词，但数据结构适合英文应用，特别是英文纠错、词频统计等。建议运行时，使用规范的中文文档测试，后期应用据具体需求做优化扩展。

4 任何个人和研究者可以对作品进程修改、完善、传播，商业使用者请注明引用来源。

1. **CUDADATRIE扩展应用**

CUDADATRIE扩展应用有：

海量文本处理：例如：英文词频统计、英文文档聚类、中文文档聚类、词频统计、信息检索等.

自然语言处理：例如，机器翻译等。

CUDA平台树形结构支持：其它树形数据结构，仿双数组结构进行扁平化处理，即可实现CUDA上树的递归操作。

任何对以上扩展应用的深入研究有兴趣者，请与作者联系。欢迎一并研究和共同进步。

1. **参考**

[1 ]<http://linux.thai.net/~thep/datrie/datrie.html> (double-array trie )

[2][Jun-ichi Aoe , Katsushi Morimoto , Takashi Sato, An efficient implementation of trie structures, Software—Practice & Experience, v.22 n.9, p.695-721, Sept. 1992](http://portal.acm.org/citation.cfm?id=146691&dl=GUIDE&coll=GUIDE&CFID=99323075&CFTOKEN=89137056)

[3]王思力,张华平,王斌. 双数组Trie优化算法及其应用研究 中文信息学报, 2006 , 20(5) :24 - 30

[4] <http://developer.nvidia.com/object/gpu_programming_guide.html>

[5] http://blog.csdn.net/OpenHero/category/379389.aspx

[6] http://cuda.itpub.net/list.html

[7]张舒 禇艳利 赵开勇 张钰勃 GPU高性能运算之CUDA 中国水利水电出版社2009

1. **致谢**

感谢CUDA大赛主办半方给予这个平台，让更多的研究作品得到分享和继承！感谢实验室老师给予的指导！感谢QQ群：48077444！最后特别感谢多次给我打电话，询问我的作品进展情况的那位CUDA大赛方的美妹，使得我对参赛的激情保持到现在。

最后感谢大家！祝大家身体健康、万事顺利！

2010-9-10