

# Matlab 之 parfor 并行编程

作者: [zdlgv5](#)

原文链接: <https://ld246.com/article/1477610180618>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# Matlab的parfor并行编程

- 

- 通常消耗最多计算资源的程序往往是循环。

把循环并行化。或者优化循环体中的代码是最经常使用的加快程序执行速度的思路。

- 
- Matlab提供了parforkeyword，能够非常方便的在多核机器或集群上实现并行计算。



## parforkeyword的使用

- 

- 由forkeyword引导的循环通常为串行运行。假设改为parfor则能够由多个worker以并行方式运行。
- parfor能够将n次循环分解为独立不相关的m部分，然后将各部分分别交给一个worker运行。
- 循环运行的结果应该与n次循环运行的顺序无关。



## parfor中的变量类型

### 简约变量

- 

- 一般parfor中各次循环相应的运算应该相互独立，但简约操作能够在多次循环内同一时候对一个量操作。这样的变量称为简约变量。

比例如以下方代码中a就是简约变量。

```
a = 0;
for i = 1:1000
    a = a+i;
end
```

- 
- 简约操作包含`+ - * & | [,] {} min max union intersect`。
- 同一个parfor循环对简约变量的操作必须一致。即必须是同一种简约操作符。并且与操作符的位置也必须一致。
- 
- 简约变量赋值表达式应该满足结合律和交换律。

`* [] {}`底层有特殊处理保证结果的正确性。

- 



### 切片变量

- 

- parfor中可能须要读取或写入parfor之外的矩阵，读取写入位置与循环变量相关。这样就须要向worker传输大量的数据。
- 
- 矩阵假设被Matlab识别为切片变量，则数据能够分段传输到各worker，提高传输效率。
- 
- 切片变量矩阵的大小是不可在parfor中改变的。且为了保证Matlab识别正确。每次循环中仅仅能取由同一个索引值索引的切片。如`a[i] a[i+1]`同一时候出现则a不被识别为切片变量

</ul>

### <h3 id="-">循环变量</h3>

<ul>

<li>如上例中的i, 表示当前循环的id。

<p>&nbsp;  </p>

<p>&nbsp;  </p>

</li>

</ul>

### <h3 id="-">广播变量</h3>

<ul>

<li>在parfor之前赋值, 在parfor内仅仅进行读取操作。

<p>&nbsp;  </p>

<p>&nbsp;  </p>

</li>

</ul>

### <h3 id="-">暂时变量</h3>

<ul>

<li>作用域局限于parfor内。parfor结束后不存在。

<p>&nbsp;  </p>

<p>不影响parfor之前声明的同名变量。</p>

</li>

</ul>

### <h3 id="-">各种变量区分的样例</h3>

<ul>

<li>下例中, parfor中的tmp是暂时变量, parfor结束后tmp的值依旧是5, 不受暂时变量的影响。</li>

<li>broadcast是广播变量。每次循环中的值不变。</li>

<li>reduced是简约变量。Matlab对其的值将分段由各worker计算后送回主进程处理。</li>

<li>sliced为切片变量。传输数据有优化提升。</li>

<li>i为循环变量。

<p>&nbsp;  </p>

```
<pre><code><span>tmp = 5;
```

```
broadcast = 1;
```

```
reduced = 0;
```

```
sliced = ones(1, 10);
```

```
parfor i = 1:10
```

```
    tmp = i;
```

```
    reduced = reduced + i + broadcast;
```

```
    sliced(i) = sliced(i) * i;
```

```
end</span></code></pre>
```

</li>

</ul>

## <h2 id="worker-">worker配置</h2>

<ul>

<li>在执行程序之前。须要配置worker。否则如前文所说, parfor循环将以普通for循环的形式执行无法并行。</li>

</ul>

### <h3 id="-">单机配置</h3>

<ul>

<li>使用matlabpool命令能够开启关闭本机的并行计算池。</li>

<li><code>matlabpool n</code>命令能够打开n个worker。</li>

<li><code>matlabpool open configname</code>依照指定配置打开, 默认配置为<code>local</code>。

<p>&nbsp;  </p>

&nbsp;

</li>

<li>程序执行结束后，应该使用<code>matlabpool close</code>关闭worker。</li>

<li>配置项的改动能够通过<code>Parallel -&gt; Manage Cluster Profile</code>完毕。

&nbsp;

&nbsp;

</li>

<li>n的选择：假设有c个cpu核心，通常能够设置为c。假设是远程server，为防止server响应卡顿，够设置为<code>c-1</code>。

&nbsp;

<p>对于计算密集型程序，超线程带来的性能提升差点儿为0，能够设置为核心数，而不是线程数。</p>

</li>

</ul>

## 注意事项

<ul>

<li>循环次数n最好能整除以worker个数m，否则部分worker会分配较多的循环，造成一部分worker闲置一段时间，减少了并行性。</li>

<li>并行执行时各个worker之间会进行通信。要注意大量传输数据带来的性能下降。

&nbsp;

<p>尤其对于广播变量。假设较大可尝试变为切片变量。</p>

<p>转载请注明作者：Focustc，博客地址为<a href="http://blog.csdn.net/caozhk" target="\_blank">http://blog.csdn.net/caozhk</a>。原文链接为<a href="http://blog.csdn.net/caozhk/article/details/38234293" target="\_blank">点我</a></p>

</li>

</ul>