# 熔毁bug和KPTI补丁：它如何影响ML性能？



2018年初，互联网获悉两起新的严重漏洞攻击事件影响了主要处理器供应商，分别被称为和。这些漏洞是处理器推测性执行中的漏洞，使得攻击者能够读取（并可能执行）各自进程之外的内存位置，这意味着程序可以读取其他软件内存中的敏感数据。

为了修复这个bug，Linux内核合并了一个名为KAISER或PTI（页表隔离）的补丁，有效地修复了崩溃攻击。然而，这个补丁导致了性能的下降，人们报告说，CPU性能的全面下降（和）从5%到5%。

然而，PTI性能问题在很大程度上取决于手头的任务 - 很可能只有在综合基准（如FSMark）中才会出现如此大的下降。所以问题是：在机器学习应用程序中，我们将看到什么样的性能冲击？

#### 设置

为了比较PTI和不PTI的性能，我安装了一台新的Ubuntu16.04机器，安装了intel微码，并将自动安装在Ubuntu16.04上的最新内核（4.10.0–42 generic）与安装了PTI补丁的最新主线内核版本（4.15.0–041500rc6 generic）进行了比较。我使用Anaconda和Python 3.6（以及pip的额外包）来执行测试。

我用来测试的设备包括一个英特尔酷睿i7-5820K（Haswell-E，库存时钟）和64GB的DDR4@2400MHz。如果人们感兴趣，我可以让我的手更多的英特尔CPU测试跨更广泛的世代。值得注意的是，AMD处理器没有启用PTI补丁，因为它们不受熔毁攻击的影响，所以如果您使用AMD，性能不会受到影响。

### 结果

首先，我们看到整体性能略有下降，但对具有卷积层的模型的推理却大幅度下降。特别是在亚历克斯内特身上，前传速度慢了5%左右，但后传速度几乎相同，这就是为什么训练的表现命中率大约是推理的一半。

就使用Keras的原始操作而言，完全连接层和LSTM层几乎不影响性能，但卷积有10%的大幅度下降。

对于Alexnet和MNIST基准，我使用while，对于Keras，我使用了一个随机初始化的模型，其中有几个层有问题，并测量了随机数据的推理速度。值得注意的是，这些基准测试完全在CPU上运行。

我在这里使用Scikit learn来衡量“经典”ML和数据科学算法的性能。与NNs相比，这里我们看到了更大的性能下降，PCA和线性/LogisticRegression受到的影响最大。这种下降的原因可能是一些数学受到了非常严重的影响，如下面的NumPy基准所讨论的。

有趣的是，kNearestNeighbour完全不受PTI的影响，而且在新内核上的性能似乎稍好一些。这可能只是在误差范围内，但可能是其他一些内核改进帮助它稍微加快了速度。

我还输入了一个pandas的基准测试，从缓存在内存中的文件中读取csv（），看看PTI - 大约6%读取（2GB，1M行，1K列，float，80%丢失），csv解析速度会降低多少。

所有scikit学习基准也都是在Bosch数据集上计算的，我发现它通常适合ML基准，因为它有大量的、标准化的和格式良好的数据（尽管kNN和Kmeans是在一个子集上计算的，因为完整的数据会花费太长时间）。

这些基准测试可能是这里最综合的，只测试一个scipy操作的速度。然而，这些结果表明，PTI的性能命中率是非常依赖于任务的。在这里我们可以看到，大多数操作只受到轻微的影响，dot积和FFT对性能的影响很小。

当启用PTI时，SVD、LU分解和QR分解都会对性能产生巨大影响，其中QR分解从190GFLOPS减少到110GFLOPS，减少了37%。这可能有助于解释PCA（严重依赖SVD）和线性回归（严重依赖QR分解）的性能下降。

这些基准测试使用的是英特尔自己的软件包 - 只是使用Anaconda，而不是英特尔的python发行版。

XGBoost给了我们一些有趣的结果。在大多数情况下，当使用较少的线程时，XGBoost的性能会比PTI有一个可忽略的降低，无论使用的是慢速精确方法还是快速直方图方法。

然而，当使用非常多的线程时，CPU同时处理更多的列，使用PTI的速度就快到了极点。

这并不是XGBoost在大量内核上的完美表现（因为它在12个逻辑内核上运行40个线程），但它表明当CPU同时处理许多事情时，PTI会产生更大的影响。不幸的是，我无法访问任何可以修改内核的高核数服务器，因此无法获得更深入的结果。

与scikit learn一样，这些基准测试是在Bosch数据集上进行的。

### 结论

最主要的是，PTI的性能影响很大程度上取决于任务，有些任务不受影响，有些任务的性能降低了40%。总的来说，我认为影响比我预期的要小，因为只有少数应用程序受到严重影响。

谢谢你的阅读！我希望这有助于了解内核更新到达时的预期情况。😊如果您有任何问题、需要运行更多基准测试的请求或希望看到在不同CPU上重复的测试，请告诉我！