DiskSeen预取算法的分析及优化研究

刘燕；朱春节

华中科技大学，武汉光电国家实验室

前言

CPU和磁盘之间的速度差异使得存储性能成为计算机系统的瓶颈环节。预取技术是缓解I/O瓶颈问题的重要技术手段。DiskSeen预取算法主要存在两点不足之处。其一，该算法在实施预取的时候对预取窗口设置了固定的min和max，如果单次请求超过max，剩下的块没有预取，那么这次请求仍然要读盘，并没有带来性能的提高。其二，该算法在进行历史感知预取时一旦找到了与当前路径相匹配的历史路径就会立刻激活预取，降低了预取的有效性。

DiskSeen算法优化方法

针对上述的第一点问题，优化后的算法可以根据激活预取请求的长度，动态地设置min和max的大小，以保证每一次的预取都能够充分发挥预取的作用。针对上述的第二点问题，优化后的算法在历史感知预取中采用了二次匹配激活历史预取的方法，在匹配序列第一次出现的时候只是把它记录下来并不激活预取，当该匹配序列第二次出现且正确时才会激活预取。这样可以进一步提高预取命中率，减少预取浪费。

预取实现

DiskSeen算法主要采用了两种预取方式，顺序预取和历史感知预取，后者的优先级更高。顺序预取的激活条件是找到了一条顺序访问序列，历史感知预取的激活条件是找到了与当前路径相匹配的历史路径。预取的时候会创建两个窗口，分别是当前窗口和预读窗口，它们的大小都是min。图1为预取窗口滑动的流程图，假设f为在当前窗口命中的块数，则新的预取窗口的大小为2*f。图2为寻找与当前路径相匹配的历史路径的流程图，图3为历史路径扩展的流程图，它们是历史感知预取的关键部分。

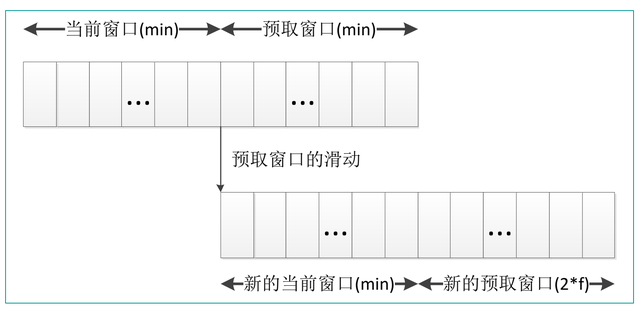

图1 预取窗口滑动

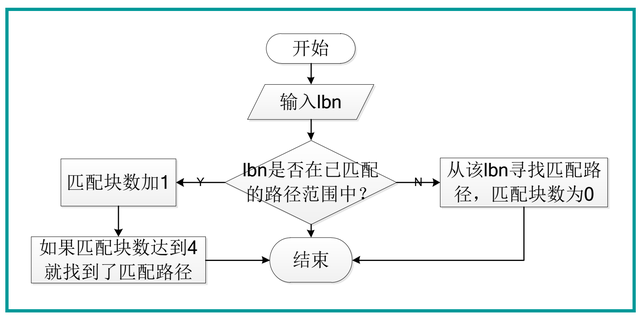

图2 寻找匹配路径

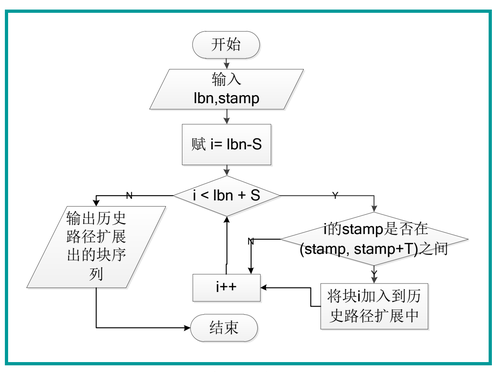

图3 历史路径扩展

实验环境

如图4所示是模拟的实验环境构成图。为了能够真实地模拟I/O层的功能，模拟的实验环境由以下几个部分组成：（1）用来保存预取数据和原始缓存数据的数据结构；（2）LRU（最近最久未被访问）缓存替换策略；（3）预取算法，包括DiskSeen算法以及优化后的DiskSeen算法；（4）Disksim（磁盘模拟器），可以真实地再现磁盘访问。

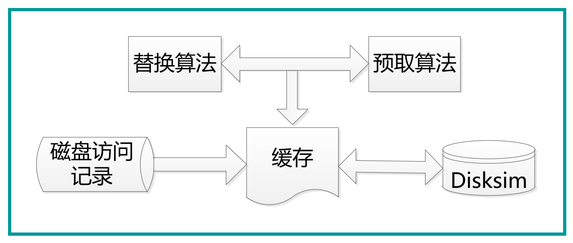

图4 模拟实验环境

实验结果

1、图5和图6分别为三种情况，即无预取、DiskSeen和改进的DiskSeen，运行五个trace后的命中率和平均响应时间的比较。从中可以得出，DiskSeen算法与无预取环境相比能够明显提高缓存命中率和降低平均响应时间，而优化后的DiskSeen算法与优化前的DiskSeen算法相比最多可以将命中率提高25%、将平均响应时间降低4.35%。

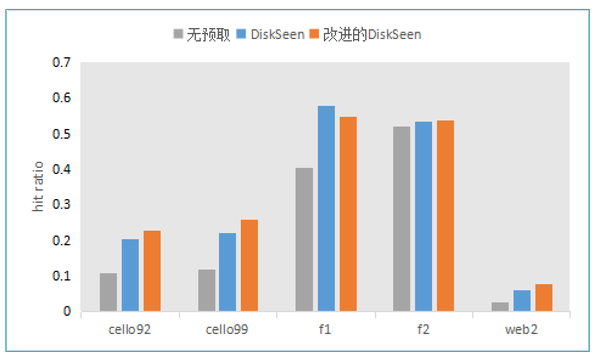

图5 命中率的比较

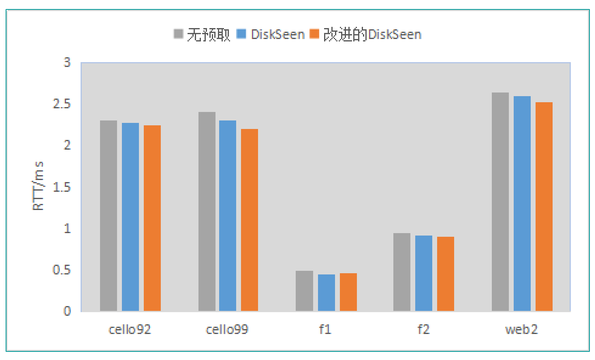

图6 平均响应时间的比较

2、如图7所示，为用优化后的DiskSeen算法连续运行两次的缓存命中率的对比。我们可以看到第二次运行的命中率有了明显的提高。连续运行性能的提高在现实中会很有意义，这是因为用户通常都只是会改变一些输入的数据，而不会改变数据集和访问模式。

3、表1显示了交替运行两个不同的trace的命中率的对比。实验表明只要块表中还存在着一条匹配的历史路径，那么噪声对历史感知预取的干扰只是微乎其微的。

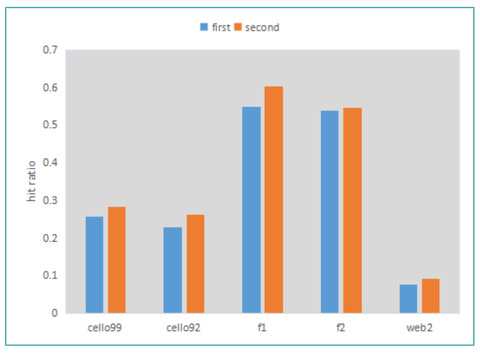

图7 连续运行两次的命中率的比较

表1 交替运行两种trace的对比

实验	命中率（hit ratio）			
无预取	f1	0.404	cello-99	0.118
1	f1	cello-99	f1	cello-99
	0.548	0.253	0.581	0.278
2	cello-99	f1	cello-99	f1
	0.256	0.571	0.278	0.581