**K-Best 测量方法**

**引自http://www.ibm.com/developerworks/cn/linux/l-cn-optimization/index.html**

在检测程序运行时间这个复杂问题上，采用 Randal E.Bryant 和 David R. O’Hallaron 提出的K次最优测量方法。

假设重复执行一个程序，并纪录该程序的K次最快执行时间，如果发现测量的误差ε很小，那么用测量得到的最快值表示程序的真正执行时间， 这种方法称为“ K 次最优（K-Best）方法”，要求设置三个参数：

（1）k：要求在某个接近最快执行时间值范围内的测量值数量；

（2）ε：测量值必须多大程度地接近，即测量值按照升序标号V1, V2, V3 ,… , Vi，… ，同时必须满足(1+ ε)Vi Vk

（3）M: 在结束测试之前，测量值的最大数量，即最大测量次数

按照升序的方式维护一个由k个最快执行时间组成的数组KTEST，数组KTEST的长度为k，记录了k个测量得到的程序执行时间，且按照升序排列。

在多轮程序执行时间测量过程中，对于每一轮测试得到的每一个新的程序执行时间测量值，如果该值比当前数组KTEST中的最大值KTEST[k]更小，则用该最新测量值替换数组中k处的元素KTEST[k]，然后再按照升序重新排列数组KTEST[k]。

持续不断进行该过程，并满足误差标准，此时就称测量值已经收敛。如果M次后，不能满足误差标准，则称为测量值不能收敛。

在接下来的所有试验中，采用K=10，ε=2%，M=200来获取程序运行时间，同时也对K次最优测量方法进行了改进：不是采用最小值来表示程序执行的时间，而是采用K次测量值的平均值来表示程序的真正运行时间。由于采用的误差ε比较大，在所有试验程序的时间收集过程中，均能收敛，但也能说明问题。

为了可移植性，采用gettimeofday()来获取系统时钟（system clock）时间，可以精确到微秒。