### Atitit 提升性能 Cache  局部性原理

提高Cache命中率，也就是充分发挥局部性原理。

局部性包括时间局部性、空间局部性。时间局部性：对于同一数据可能被多次使用，自第一次加载到Cache Line后，后面的访问就可以多次从Cache Line中命中，从而提高读取速度（而不是从下层缓存读取）。空间局部性：一个Cache Line有64字节块，我们可以充分利用一次加载64字节的空间，把程序后续会访问的数据，一次性全部加载进来，从而提高Cache Line命中率（而不是重新去寻址读取）。

### Padding 方式

正确的方式应该将该对象属性分组，将一起变化的放在一组，与其他属性无关的属性放到一组，将不变的属性放到一组。这样当每次对象变化时，不会带动所有的属性重新加载缓存，提升了读取效率。在JDK1.8以前，我们一般是在属性间增加长整型变量来分隔每一组属性。被操作的每一组属性占的字节数加上前后填充属性所占的字节数，不小于一个cache line的字节数就可

通过填充变量，使不相关的变量分开

### Contended注解方式

在JDK1.8中，新增了一种注解@sun.misc.Contended，来使各个变量在Cache line中分隔开。注意，jvm需要添加参数-XX:-RestrictContended才能开启此功能   
用时，可以在类前或属性前加上此注释：