**伤害数字转String产生大量GC问题**

解决方法:请参考如下处理函数：

static char [] s\_charIndex = new char[10];

public static void InitCharIndex()

{

char index = '0';

for (int i = 0; i < 10; ++i)

{

s\_charIndex[i] = index;

index++;

}

}

public static int GetIntLength(int value)

{

int count = 1;

value = value / 10;

while ( value > 10)

{

count++;

value = value / 10;

}

if(value > 0)

count++;

return count;

}

public static void ChangeStringContent(string strDest, int number)

{

if (strDest == null)

return;

int length = GetIntLength(number);

unsafe

{

fixed (char\* aTemp = strDest)

{

for (int i = length - 1; i >= 0; --i)

{

\*(aTemp + i) = s\_charIndex[number % 10];

number = number / 10;

}

}

}

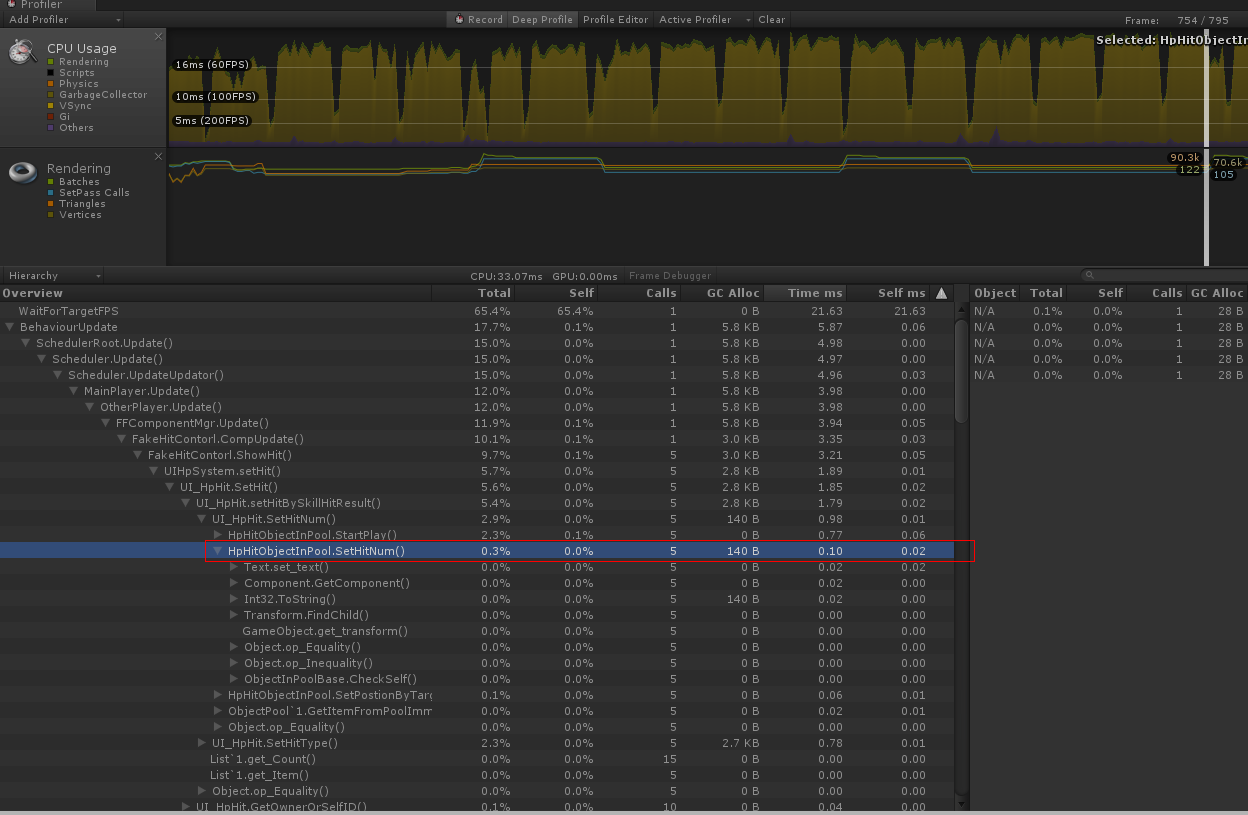
}

验证上述方法能否解决伤害数字tostring的GC问题。

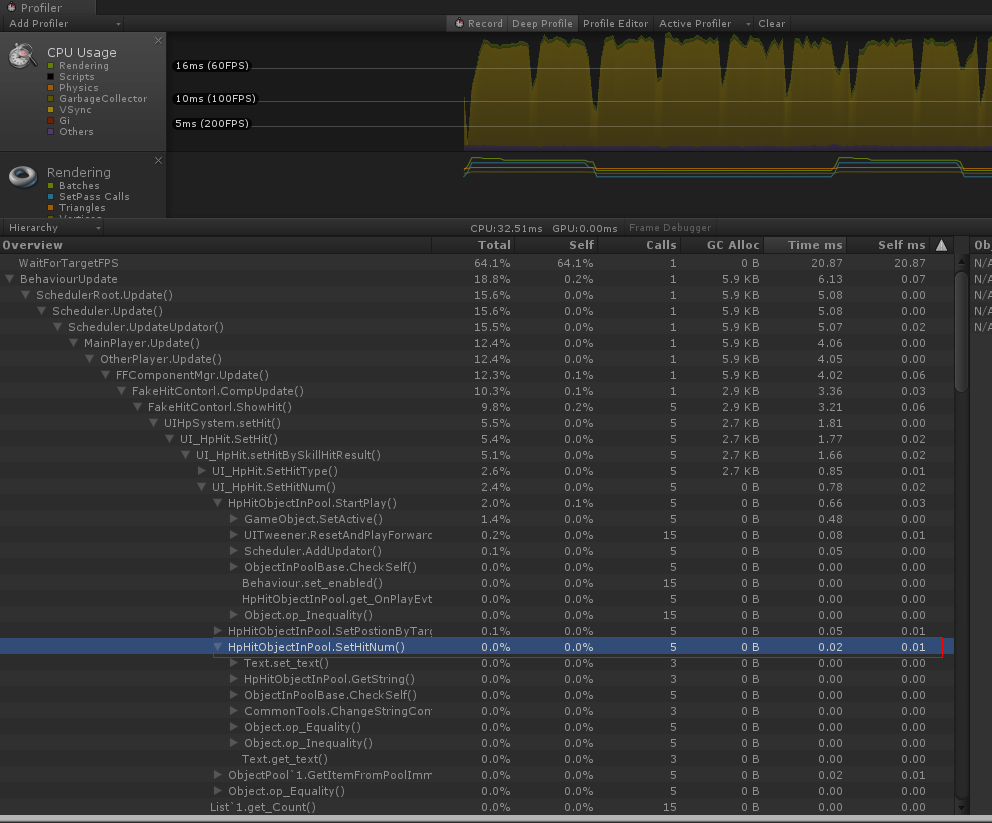
Windows 端：

使用上述方法替换int.tostring()方法性能测试。如图：

替换前：



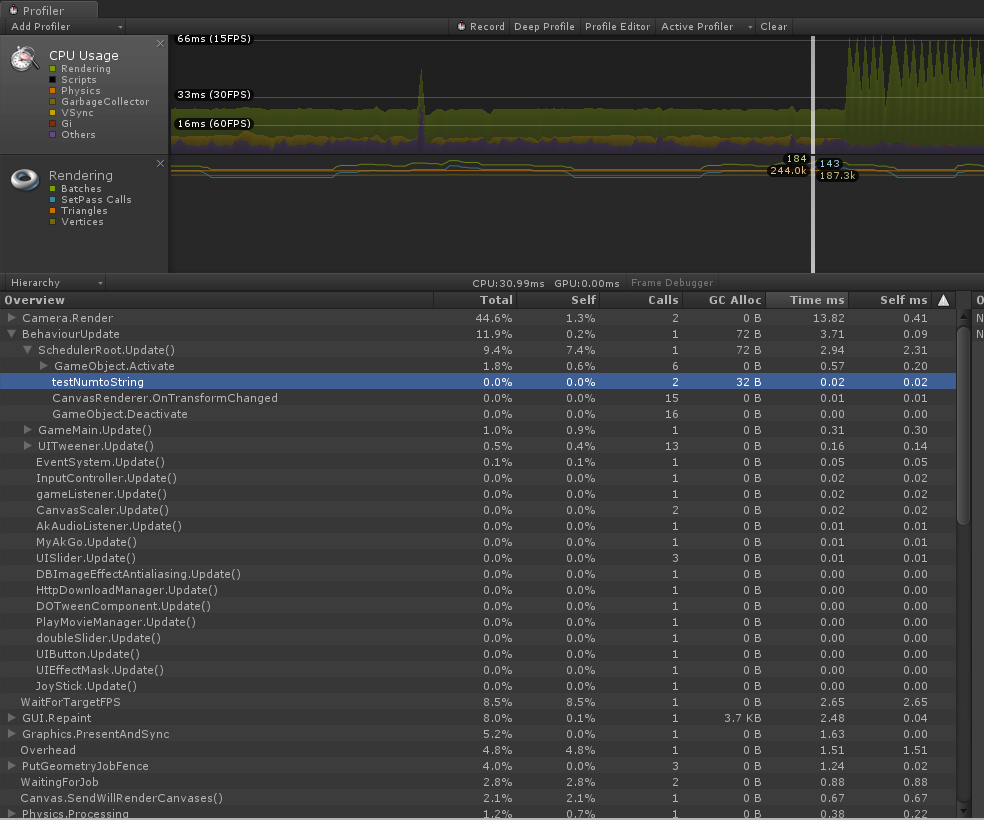
替换后：



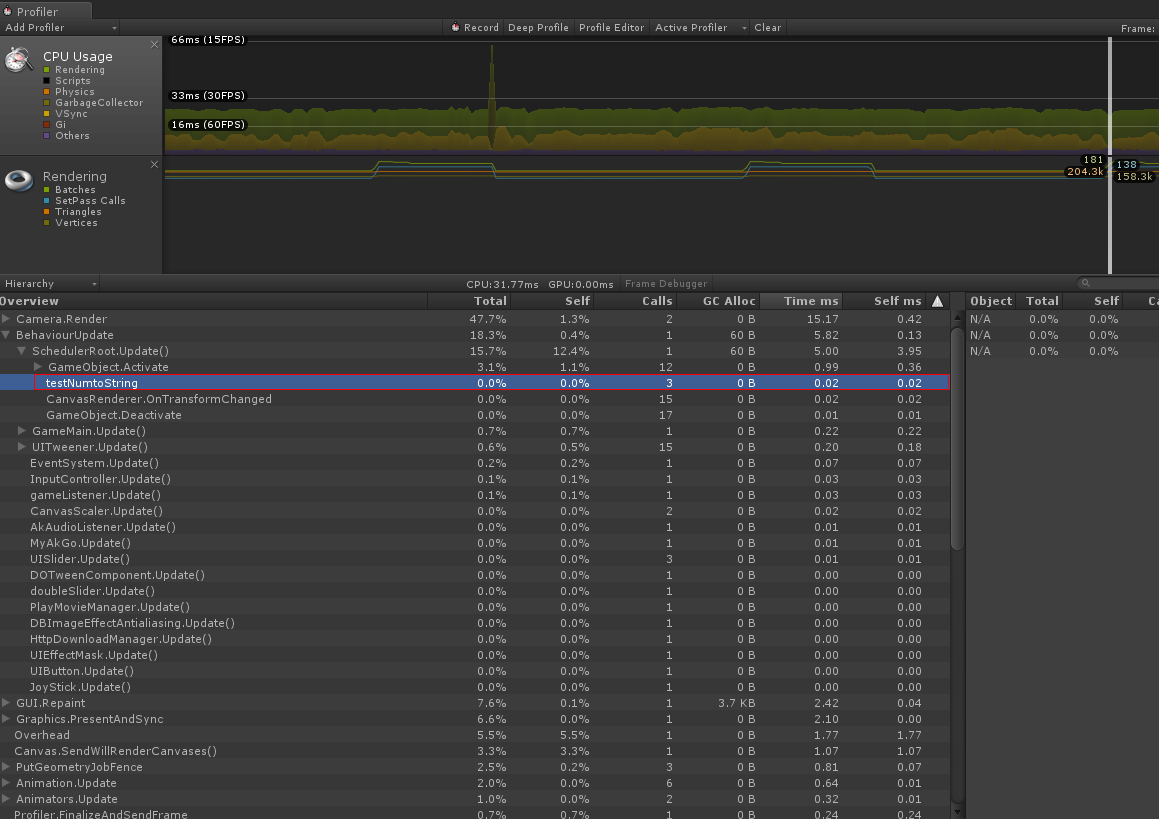
如上图对比：SetHitNum()方法不再产生GC

Android：采样标题”testNumtoString”

修改前：



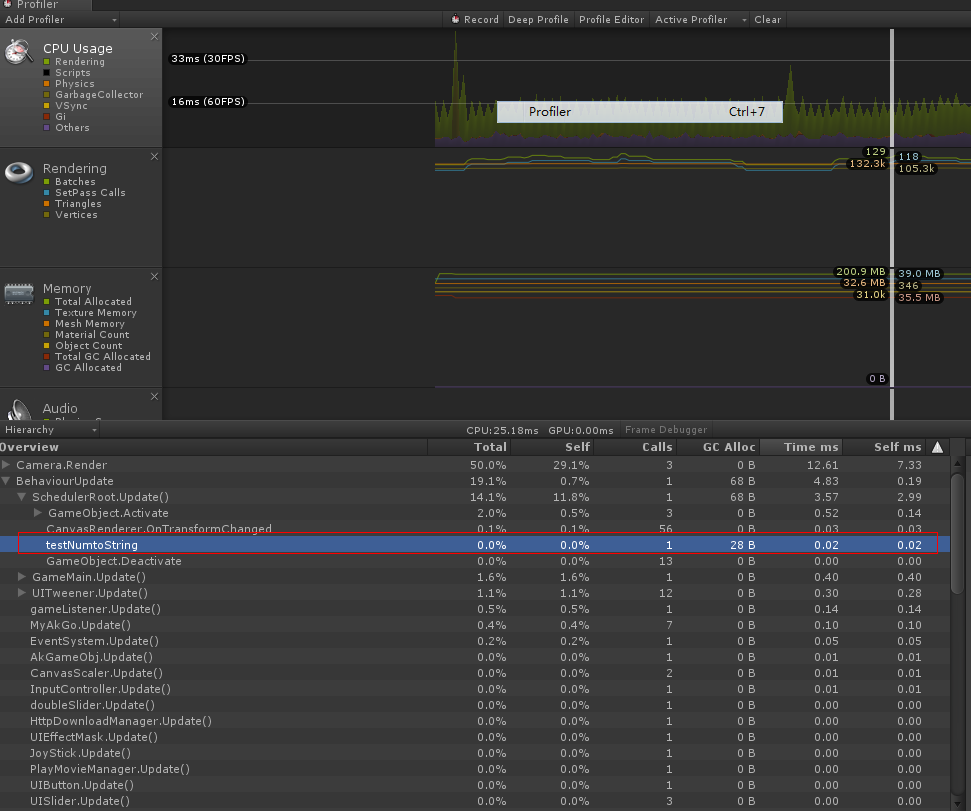
修改后：



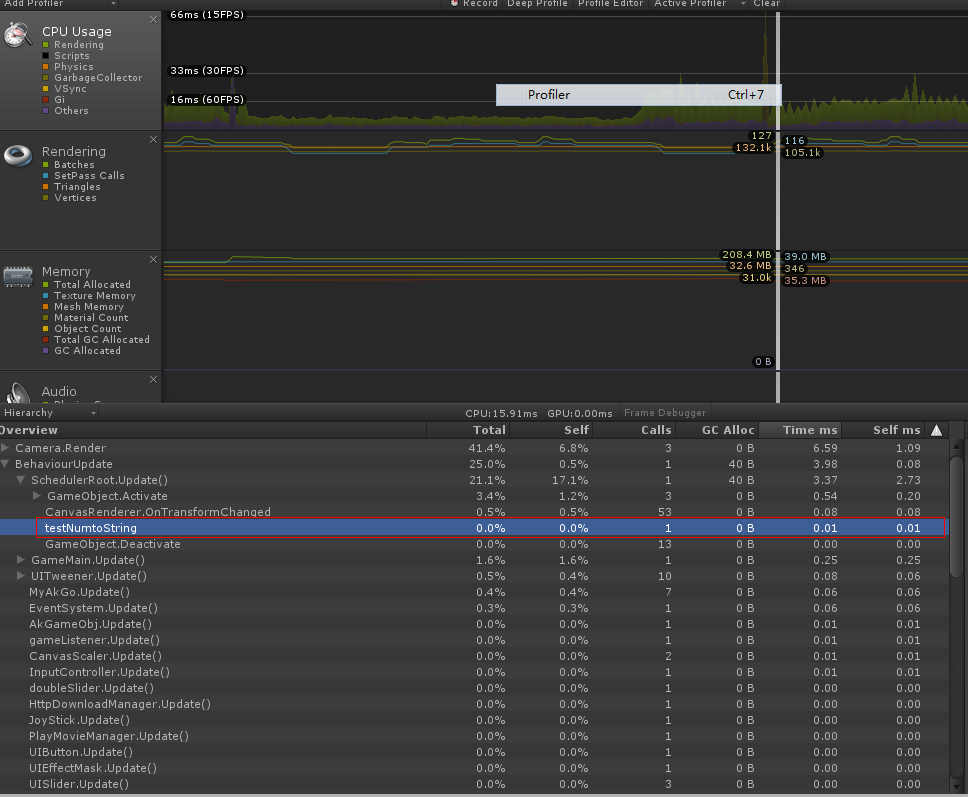
使用Profiler.BeginSample("testNumtoString"); 采样得到tostring和tostring替代方法的性能对比。修改前有32B的GC，修改后没有GC。

IOS：采样标题”testNumtoString”

修改前：



修改后：



使用Profiler.BeginSample("testNumtoString"); 采样得到tostring和tostring替代方法的性能对比。修改前有28B的GC，修改后没有GC。

结论：自定义的方法替换tostring方法可以消除伤害数字中tostring产生的GC。