测试数据：

每个角色：120动态骨骼数量  
骨骼链条：15左右  
每个骨骼链条：8左右的动态骨骼数量

**测试机型：红米K40 Pro （高通骁龙888）**

DynamicBone测试结果：

|  |  |  |
| --- | --- | --- |
| 测试角色数（总骨骼数） | 耗时均值/ms | Job额外占用耗时/ms |
| 3（360） | 0.268 | 0.694 |
| 10（1200） | 0.311 | 1.538 |
| 30（3600） | 0.432 | 2.256 |

MagicaCloth测试结果：

|  |  |  |
| --- | --- | --- |
| 测试角色数（总骨骼数） | 每个角色的MagicaCloth脚本数 | 耗时均值/ms |
| 3（360） | 1（每个脚本负责15条链） | 1.73 |
| 5（每个脚本负责3条链） | 2.343 |
| 10（1200） | 1（每个脚本负责15条链） | 3.291 |
| 5（每个脚本负责3条链） | 3.706 |
| 30（3600） | 1（每个脚本负责15条链） | 5.003 |
| 5（每个脚本负责3条链） | 5.126 |

从Profiler里看的话，在总骨骼链数和总骨骼数不变的前提下，增加脚本数增加的耗时在于“分配数据到线程上”，实际“骨骼解算”的耗时并没有怎么增加。因此数据量越大，脚本数量带来的耗时差距就越不明显

**测试机型：OPPO K3（高通骁龙710）**

DynamicBone测试结果：

|  |  |  |
| --- | --- | --- |
| 测试角色数（总骨骼数） | 耗时均值/ms | Job额外占用耗时/ms |
| 3（360） | 0.320 | 0.971 |
| 10（1200） | 0.489 | 2.297 |
| 30（3600） | 1.496 | 4.034 |

MagicaCloth测试结果：

|  |  |  |
| --- | --- | --- |
| 测试角色数（总骨骼数） | 每个角色的MagicaCloth脚本数 | 耗时均值/ms |
| 3（360） | 1（每个脚本负责15条链） | 3.482 |
| 5（每个脚本负责3条链） | 3.757 |
| 10（1200） | 1（每个脚本负责15条链） | 4.289 |
| 5（每个脚本负责3条链） | 4.489 |
| 30（3600） | 1（每个脚本负责15条链） | 11.95 |
| 5（每个脚本负责3条链） | 12.788 |