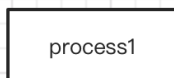


cat-msgid方案优化

当前方案:



msgidx: | sec1 | sec2 | sec3 | .. | sec3600 |

sec1: | port1 | port2 | ... | port10 |

sec2: | port1 | port2 | ... | port10 |

port1: subId

msgIdx = secBase + portBase + subId
(一个baseid最多可以生成 1000000个msgidx)

msgid = domain + ip + hour + msgIdx

1. 将int32的值空间, 等分成3600份, 1s一份
2. 对于每1秒钟的值空间, 等分成10份, 1个端口(port%10)一份;
3. 1个端口内的值空间, 进行subId
4. 总的msgIdx = secBase + portBase + subId

优点:

不需要跨进程间id的协商, 比较独立;

缺点:

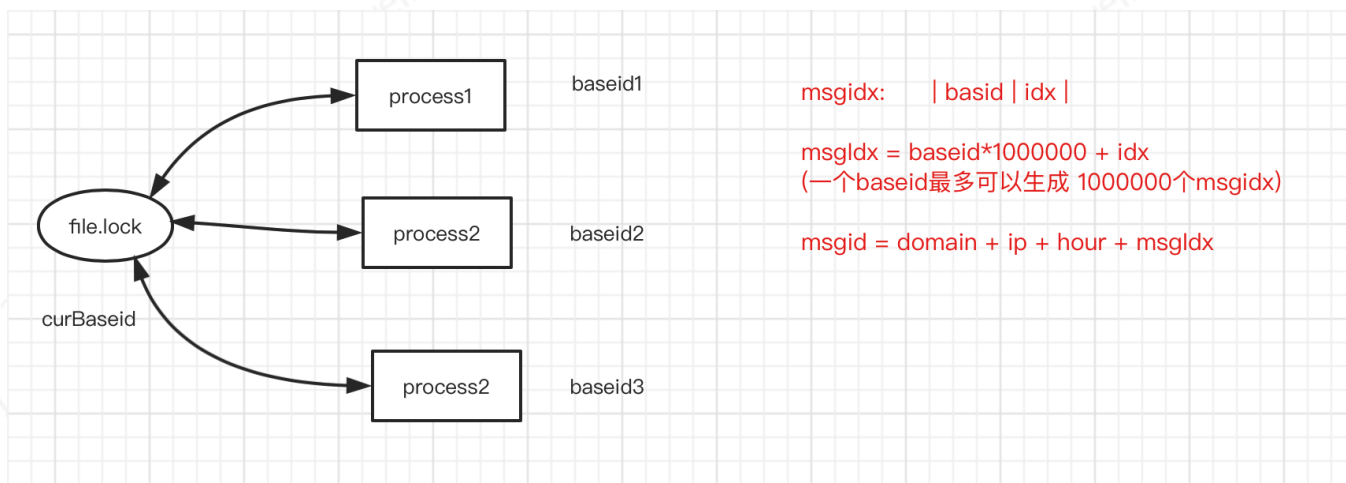
粒度太细, 复杂;

假设条件也许不太合理, 对于1s中, 甚至是1s中每个端口对应的app, 产生的msg数量不能超过 $\text{maxInt32}/3600/10 = 59652$

限制了单机进程/monitor句柄的数量(10),

单机内id冲突概率较大, 由于对port % 10;

最新方案:



1. 对int32的值空间, 进行1000000等分, 等分点记做baseid
2. 每个进程去获取1个baseid, 然后内部细分subId

优点:

方案简单,

假设较少, 除了对整个物理机, 每小时产生的msg数量不能超过maxInt32

缺点:

需要跨进程间通信(文件锁)

对于一台母机中的多个实例, 需要某种全局文件的支持(感觉打破了容器的隔离);