SSD 动态规划模型

日期：2024 年 3 月 6 日

1 模型

k 为时刻下标，$k = 0, 1, 2, 3, ..., N$.

Δt 为采样间隔.

x_k 为 SSD 当前状态，我们把当前状态简化表示成当前 $[k-1, k]$ 区间的 IOPS (Input/Output Operations Per Second)，即相当于 x_k^{IOPS}. 在 $k-1$ 时刻得到，即未来时段需要执行多少 IO 的值. 定义 y_j 含义相同，但 j 属于另一个时间尺度.

$IO\_R\_N_k$ 为时段 $[k-1, k]$ 的 IO 读序列的数量, 在 k 时刻得到.

$IO\_W\_N_k$ 为时段 $[k-1, k]$ 的 IO 写序列的数量, 在 k 时刻得到.

$GC\_RW\_N_k$ 为时段 $[k-1, k]$ 的前端 GC 需求数量, 在 k 时刻得到.

GC_N_{k-1} 为当前 $[k-1, k]$ 计划执行的 GC 数量，在 $k-1$ 时刻设置.

V_l 为当前时刻 l 剩余的 VPC 数量 (4kb), l 对应的时间间隔为 10ms，空白块的更新间隔为 10ms, l 为时刻下标，$l = 1, 2, ..., L$.

BLK_N_l 为当前时刻 l 空白块数量，在 l 时刻得到，已知 BLK_N_0. 设置 LBB 和 UBB 为空白块数量在任意时刻的下界约束和上界约束.

$GC\_blk\_VPC_l$ 为当前时刻 l 的累计 VPC4KB 个数值，在 l 时刻得到. VPC_l 为当前时刻 VPC4KB 个数值.

FKB_N 回收单元 4KB 数量，已知.

ur_k 和 uw_k 分别为当前时刻 k 采取的动作, 即决定执行 IO 读和写命令个数. 记 $u_k = (ur_k, uw_k)$. $ur_k, uw_k \geq 0 \quad \forall k = 0, 1, 2, ..., N-1$.

定义 f_{SSD} 为预测当前时刻 GC 执行数量函数, 对应 $GC\_N_k = f_{SSD}(ur_k, uw_k)$.

当前时刻 k 的状态转移函数 $x_{k+1} = \frac{ur_k + uw_k}{\Delta t}$.

我们可以假设 SSD 在整个过程中的是静态的，即执行系统平稳运行，没有发生故障，同时假设处理结束时间为时刻 N. 我们给出以下动态规划模型.

由于带宽波动约束与带宽分配模型可能有不同的时间粒度，假设带宽波动约束的时间粒度更宽，是带宽分配模型的 M 倍. 记带宽波动约束的时间下标为 $j = 0, 1, 2, ..., N-1$，对应 IOPS 序列为 y_j，令当前时刻 j 的 IOPS 均值为 $AVG_j = \frac{\sum_{i=1}^{i \leq j} y_i}{j}$，最小值为 $MIN_j = \min\{y_1, y_2, ..., y_j\}$，假设成本函数是:

$$g(u_k)=\begin{cases} -x_{k+1}, & \text{if } y_{j+1}\in[\frac{0.9j\cdot AVG_j}{j+0.1},\frac{-0.9j\cdot AVG_j+(j+1)MIN_j}{0.9}] \\ penalty>0, & \text{otherwise}\end{cases} \tag{1}$$

区间推导详见附录.

全过程成本函数是:

$$J(x_0;u_0,u_1,...)=\sum_{k=0}^{N-1}g(u_k) \tag{2}$$

我们可以将 SSD 调控问题描述为，找到控制序列 $\pi=\{u_0,u_1,...\}$，使得全过程成本函数最小，即尽量使得 IOPS 波动平稳且高. 并且保证总时间段内所有 IO 均被执行 (约束8 & 9).

$$\min_{\pi} J_\pi(x_0)=\min_{\{u_0,u_1,...\}}\sum_{k=0}^{N-1}g(u_k) \tag{3}$$

$$s.t.$$

$$y_j=\frac{\sum_{k=(j-1)\cdot M}^{j\cdot M-1}x_k}{M}.\forall j=1,2,...,\lfloor\frac{N}{M}.\rfloor \tag{4}$$

$$GC\_N_k=f_{SSD}(ur_k,uw_k).\forall k=0,1,2,...,N-1. \tag{5}$$

$$\sum_{t=0}^{t=k}ur_t\le\sum_{t=1}^{t=k+1}IO\_R\_N_t.\forall k=0,1,2,...,N-1. \tag{6}$$

$$\sum_{t=0}^{t=k}uw_t\le\sum_{t=1}^{t=k+1}IO\_W\_N_t.\forall k=0,1,2,...,N-1. \tag{7}$$

$$\sum_{t=0}^{t=N-1}ur_t=\sum_{t=1}^{t=N}IO\_R\_N_t. \tag{8}$$

$$\sum_{t=0}^{t=N-1}uw_t=\sum_{t=1}^{t=N}IO\_W\_N_t. \tag{9}$$

$$VPC_l=GC\_blk\_VPC_l-GC\_blk\_VPC_{l-1}. \tag{10}$$

$$VPC_0=GC\_blk\_VPC_0. \tag{11}$$

$$VPC_1,...,VPC_N\text{(按顺序挑选的非零 VPC 序列)}\{VPC_m\} \tag{12}$$

$$BLK\_N_{l+1}=BLK\_N_l+\frac{GC\_N_k}{\Delta t/10ms}\cdot\frac{FKB\_N-VPC_m}{VPC_m}-\frac{uw_k}{\Delta t/10ms},\forall k=0,1,2,...,N-1,$$

从初始时刻开始，当执行的 GC 总数超过 VPC_1，则调用 VPC_2 代入计算，以此类推 (13)

$$(k)\cdot(\Delta t/10ms)\le l<(k+1)\cdot(\Delta t/10ms). \tag{14}$$

$$LBB\le BLK\_N_l\le UBB,\quad LBB=0,\quad UBB=576000,\ \forall l=1,2,...,L. \tag{15}$$

函数 $GC\_N_k = f_{SSD}(ur_k, uw_k)$ 输出为以下方程变量 GCW 的解:

$$c_1 \cdot GCR1 + c_2 \cdot GCR2 + d \cdot GCW = P - a \cdot ur_k - b \cdot uw_k \tag{16}$$

$$GCW - GCR1 - GCR2 = 0 \tag{17}$$

$$GCR1 : GCR2 = constant \tag{18}$$

$$GCR1, GCR2 \geq 0 \tag{19}$$

$$(a, b, c_1, c_2, d) = (0.1343, 0.1851, 0.2297, 0.4508, 1.0852 \times 10^{-13}) \tag{20}$$

$$P = 467798.1 \tag{21}$$

2 附录

2.1 反馈函数推导

公式 (1) 的推导过程如下:

带宽波动标准为: (平均值-最小值)/平均值 ≤ 10%. 即 $\frac{AVG_j - MIN_j}{AVG_j} \leq 10\%$. 并且可以得到

$$MIN_j \geq 0.9 \cdot AVG_j \tag{22}$$

在加入 y_{j+1} 后的平均值 AVG_{j+1} 是

$$AVG_{j+1} = \frac{j \cdot AVG_j + y_{j+1}}{j+1} \tag{23}$$

所以, y_{j+1} 应该满足

1. 如果 y_{j+1} 是最小值, 此时有 $y_{j+1} \leq MIN_j$.

$$\frac{AVG_{j+1} - y_{j+1}}{AVG_{j+1}} \leq 10\% \tag{24}$$

即

$$\frac{j \cdot AVG_j + y_{j+1} - (j+1) \cdot y_{j+1}}{j \cdot AVG_j + y_{j+1}} \leq 10\% \tag{25}$$

进一步

$$y_{j+1} \geq \frac{0.9j \cdot AVG_j}{j+0.1} \tag{26}$$

并且有

$$\frac{0.9j \cdot AVG_j}{j+0.1} \leq \frac{j \cdot MIN_j}{j+0.1} < MIN_j \tag{27}$$

2. y_{j+1} 非最小值, 此时有 $y_{j+1} \geq MIN_j$.

$$\frac{AVG_{j+1} - MIN_j}{AVG_{j+1}} \leq 10\% \tag{28}$$

即

$$\frac{j \cdot AVG_j + y_{j+1} - (j+1) \cdot MIN_j}{j \cdot AVG_j + y_{j+1}} \leq 10\% \tag{29}$$

进一步

$$y_{j+1} \leq \frac{-0.9j \cdot AVG_j + (j+1)MIN_j}{0.9} \tag{30}$$

并且有

$$\frac{-0.9j \cdot AVG_j + (j+1)MIN_j}{0.9} \geq \frac{-j \cdot MIN_j + (j+1)MIN_j}{0.9} = \frac{MIN_j}{0.9} \geq MIN_j \tag{31}$$

即 $MIN_j \leq \frac{-0.9j \cdot AVG_j + (j+1)MIN_j}{0.9}$.