

动态消息队列负载均衡策略的研究与应用

嵇智辉^{1,2}, 倪宏², 刘磊^{1,2}

(1. 中国科学院研究生院, 北京 100039; 2. 中国科学院声学研究所国家网络新媒体工程技术研究中心, 北京 100080)

摘 要: 针对约束条件下多消息输入问题, 提出多消息队列并发处理模型, 权值负载预测模型和两阶段调度策略, 实现接入消息快速分发和消息队列间动态负载均衡。通过仿真证明其优越性, 并实现采用此策略的 IPTV 互动消息处理系统, 满足了互动消息业务对系统消息接收能力、并发处理能力和实时响应能力等方面的需求。

关键词: 约束条件; 消息队列; 权值负载预测模型; 负载均衡; 互动消息处理系统

Research and Application of Dynamic Message Queue Load Balance Strategy

Ji Zhi-hui^{1,2}, Ni Hong², Liu Lei^{1,2}

(1. Graduate University of Chinese Academy of Sciences, Beijing 100039;

2. National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100080)

【Abstract】 Aiming at the issue of multiple input message under restricted conditions, this paper provides multi-message-queue concurrent model and weight load forecast model, as well as a two-phase strategy, which makes the quick dispatching and dynamic load balance come true. It proves its advantages by simulation, and designs an interactive message management system, which meets requirements of interactive message service.

【Key words】 restricted condition; message queue; weight load forecast model; load balance; interactive message management system

IPTV 互动消息业务,是指用户在观看 IPTV 节目过程中,通过电视遥控器、手机或 PC 等终端,实时与 IPTV 业务系统交互,参与投票、竞猜或互动问答等娱乐活动的一类 IPTV 增值业务。针对该类业务交互时间集中,参与用户数量较大,对系统的消息接收能力、并发处理能力和实时响应能力需求较高的实际问题,本文基于业务实际需求和约束条件,改进传统消息队列负载模型,采用消息队列^[1]技术,将工程问题转化约束条件下的多输入多消息队列缓存并发处理模型,提出计算消息队列负载的权值预测模型和两阶段均衡调度策略,实现接入消息的快速分发和消息队列间动态负载均衡,从而提高消息接收能力和并发处理能力,降低任务处理时间,并将理论应用到工程中。

1 问题描述

用 $T=\{T_1, T_2, \dots, T_m\}$ 代表输入消息类别, $W=\{W_1, W_2, \dots, W_m\}$ 代表不同类别消息的处理时间权值, $I=\{I_1, I_2, \dots, I_n\}$ 代表多输入消息, $Q=\{Q_1, Q_2, \dots, Q_r\}$ 代表多消息队列, $P=\{P_1, P_2, \dots, P_r\}$ 代表各个消息队列对应的处理模块。其中, $T_i \leftrightarrow W_i, Type(I_i) \in T, Q_i \leftrightarrow P_i$ 。则问题可以表示为: 将从属于 T 类别的多个消息输入到系统, 系统将消息分发到多消息队列 Q , 各个消息队列对应的处理模块 P_i 实时从消息队列中提取消息进行处理; 采用何种消息分发和均衡调度策略, 能够实现消息的快速接入消息队列间的负载均衡, 并保障任务的总体完成时间最短。其数学模型如图 1 所示。

该模型不同于常规多任务多处理器模型, 依据实际情况存在约束条件, 具体如下:

(1) 消息输入具有高并发特征, 要求尽量减小消息分发时

间, 以免造成阻塞甚至丢包;

(2) 输入中存在大量同类消息, 消息具有处理时间权值 W_i , 不同类别消息处理时间差异较大。

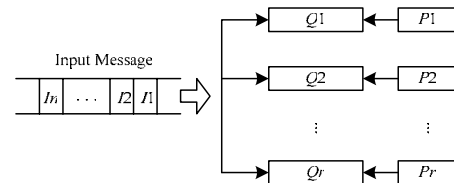


图 1 数学模型

由于存在上述约束, 传统的均衡调度策略如 Min-Min 算法^[2], 在消息分发前要计算任务完成时间, 增加了消息分发时间, 会导致部分输入消息阻塞在前端, 以至产生丢包问题; 轻载优先策略, 消息分发前需要计算消息队列负载, 同样增加了消息分发时间。常规消息队列负载模型, 如基于阈值的队列模型^[3], 根据队列消息长度和阈值判断队列负载, 不区分消息类别将消息等同对待, 违背了约束条件中不同类别消息处理时间存在差异较大的实际情况, 不适合使用。

2 动态负载均衡策略

本文采用权值负载预测模型, 依据输入消息处理时间权

基金项目: 国家科技支撑计划基金资助项目“数字媒体服务示范工程”(2006BAH02A22)

作者简介: 嵇智辉(1979 -), 男, 博士研究生, 主研方向: 宽带网络通信, 电信运营支撑系统; 倪宏, 研究员、博士生导师; 刘磊, 博士研究生

收稿日期: 2008-08-23 **E-mail:** jizh@dsp.ac.cn

值计算消息队列负载，结合预测算法判断队列状态，解决了传统模型中将不同类别输入消息同等对待并且忽略负载变化的问题，提高了负载评估的准确性和精确度。基于此模型，提出标记分发和均衡调度两阶段调度策略。第1阶段：采用 Round-Robin 算法，将接入消息在可用消息队列间顺序分发。第2阶段：动态采集成员队列的负载信息，采用消息队列权值负载预测模型和负载迁移均衡策略进行均衡调度，实现了消息队列间负载均衡，同时依据负载标记队列状态，指导消息分发。两阶段调度策略避免了动态计算负载，并依据负载进行消息分发所引起的处理时间消耗，进而造成消息接收阻塞和丢弃现象。

2.1 权值负载预测模型

在传统阈值队列模型的基础上，结合阈长阈值模型^[3]，提出消息队列权值负载预测模型。模型为每类消息基于处理时间分配消息权值，利用消息权值计算队列负载，同时结合预测算法判断队列状态。

模型以上一次调度结束时刻为零起点，统计在 $0 \sim \varepsilon$ 时间段内，出入队列消息状况，计算队列负载变化和系统消息处理能力，并依据此信息，预测队列负载变化曲线。

假定在 $0 \sim \varepsilon$ 时间段内，有 P 条消息输入队列； Q 条消息输出队列； ε 时刻队列内有 N 条消息； p_i 代表第 i 条输入消息权值； q_j 代表第 j 条输出消息权值； r_k 代表队列内第 k 条消息权值。

则在 $0 \sim \varepsilon$ 时间段内队列输入权值为 $R_{in} = \sum_{i=1}^P p_i$ ，输出权值为 $R_{out} = \sum_{j=1}^Q q_j$ ；单位时间系统处理能力为

$$D = (R_{out} - R_{in}) / \varepsilon = \left(\sum_{j=1}^Q q_j - \sum_{i=1}^P p_i \right) / \varepsilon ;$$

$$\text{队列在 } \varepsilon \text{ 时刻权值 } R_\varepsilon = \sum_{k=1}^N n_k, \text{ 消息平均权值为 } \bar{r} = \frac{R_\varepsilon}{N},$$

队列长度 $L_\varepsilon = N$ 。

依据上述信息，预测队列在 t 时刻权值负载为 $R_t = R_\varepsilon + D(t - \varepsilon)$ ，消息长度负载为

$$L_t = Rt / \bar{r} = \left[R_\varepsilon + D(t - \varepsilon) \right] / \bar{r} = N \left[1 + \left(\sum_{j=1}^Q q_j - \sum_{i=1}^P p_i \right) (t - \varepsilon) / \left(\varepsilon \sum_{k=1}^N n_k \right) \right]$$

在该模型中，队列有轻载、适中和超载 3 个状态，依据阈长阈值模型，定义队列长度 $L_t > \alpha M$ 为过载状态， $L_t < \beta M$ 为轻载状态， $\beta M < L_t < \alpha M$ 为负载适中状态(其中 α 为过载系数， β 为轻载系数， M 为队列容量)，并得到队列负载变化与状态曲线，如图 2 所示。

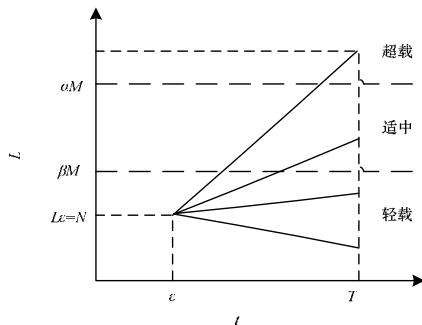


图 2 队列负载变化与状态曲线

队列负载计算公式为

$$F(t) = \begin{cases} 0 & L_t < \beta M & \text{队列轻载} \\ 1 & \beta M < L_t < \alpha M & \text{队列负载适中} \\ 2 & L_t > \alpha M & \text{队列超载} \end{cases}$$

如图 2 所示，当 T 时刻队列消息长度负载 $L_t < \beta M$ 时，说明队列负载较轻，可以成为负载迁移接受者；当 $\beta M < L_t < \alpha M$ 时，表示队列负载适中，当前所有任务可在本地节点消化，但不接收过载队列负载迁移；当 $L_t > \alpha M$ 时，表示队列负载较重，将停止参与任务分配，须进行负载迁移。

2.2 两阶段调度策略

本文采用标记分发和均衡调度两阶段调度策略，实现了接入消息的快速分发和消息队列间动态负载均衡。消息队列分为轻载、适中和超载 3 种状态，轻载和适中表示任务可在本地消化，队列允许接入消息，超载表示队列压力较大，需要暂停一段时间再接收消息。队列初始状态均为轻载，均衡调度阶段依据队列负载变化情况，结合权值负载预测模型标记队列状态，指导标记分发执行。

2.2.1 标记分发

系统将消息队列按序号排列，保存消息最后分发的队列序号 Sequence，并为每个队列保存状态标识，轻载和适中队列标识为可用，超载队列标识为暂停。当接收到接入请求时，采用 Round-Robin 算法，选择 Sequence 下一个消息队列，如果该队列为可用状态，将接入消息发送到该队列处理；否则继续查找下一个队列。采用此轮询算法，不需要实时掌握队列负载，只需了解队列当前状态即可，提高了系统消息接收能力。算法如下：

```
P_DISPATCH(Message msg)
index = Sequence
while(true)
index = (index++)%QUEUE_SIZE
if(Queue[index].state 为可用状态)
transfer(msg, Queue[index])
```

2.2.2 均衡调度

动态负载均衡阶段，采用队列主动申请调度结合中心定时调度，实现消息队列间负载均衡。队列主动申请调度，是指各消息队列在 ε 时刻依据负载计算模型，统计并预测出消息队列在 T 时刻负载后，如果判断队列超载则上报中心，中心采集各消息队列负载，控制消息队列进行负载均衡调度；中心定时调度，是指消息队列统计负载后，没有发现队列超载，则中心在 T 时刻主动发起调度，采集消息队列负载，实现队列间负载均衡。系统采用超载队列标记与负载迁移结合的均衡策略，实现了负载调度和负载均衡^[4]。该方案支持系统动态负载平衡，能有效提高系统整体性能。

(1) 队列主动申请调度

当任一消息队列预测到超载，则向中心发起调度申请。中心实时采集所有队列负载 L_ε 和 L_t ，依据 L_t 判断队列负载状态，将队列依据状态标记为超载、适中和轻载，超载队列不接受消息输入，直到队列状态被标记为适中或轻载为止。通过超载标记，避免了超载队列中消息继续增加，导致输入消息丢弃等现象发生。

同时，中心根据队列负载状态，将超载队列与轻载队列依据当前负载 L_ε 进行排序，并在负载最重与负载最轻队列之间进行负载迁移，将超载队列中超过二者负载均值的消息迁移到轻载队列，迁移成功后，2 个队列都基本处于负载适中状态。系统持续执行该步骤，直到将排列中所有超载队列处

理完成为止。

负载均衡调度完成后，中心与各队列将系统时间标记为零，重新开始下一调度周期。算法如下：

```
ACTIVE_SCHEDULE(Queue[] Q)
Collect  $L_e$  and  $L_T$  of all Q
for each q in Q
    if( $L_T$  为超载) mark state 暂停
    else mark state 可用
Sort Q by  $L_e$  asc
i = 0, j = QUEUE_SIZE-1
while(Q[j] 超载)
    Q[j]到 Q[i]过载迁移
    i++, j--
(2)中心定时调度
```

在上一次调度发生后的 T 时间内，系统经过多个 ε 时间点计算并预测负载后，没有发现队列超载情况，进入中心定时调度。中心定时调度目的是在轻载队列和适中队列间进行适当的平衡，增加部分轻载甚至没有接入消息的队列负载，提高系统的处理效率和利用率。调度发生时，中心采集队列 T 时刻负载，将消息队列按照负载排序，在负载最重与负载最轻队列之间进行负载迁移，迁移之后队列负载变为平衡。中心与各消息队列将系统时间标记为零，重新开始下一调度周期。

2.3 仿真分析

本文对不采用均衡调度、采用阈值模型和权值预测模型 3 种方案进行仿真。实验中采用 5 个消息队列，队列长度为 10 000；5 类输入消息，权值分别为 100 ms, 200 ms, 400 ms, 700 ms 和 1 000 ms。阈值模型调度阈值为 3 000，调度时间间隔 $T=10\ 000$ ms；权值模型队列轻载长度为 1 000，超载长度为 8 000，主动调度时间间隔 $\varepsilon=1\ 000$ ms，中心定时调度时间间隔 $T=10\ 000$ ms。选取系统某一时刻状态为零起点，10 ms 为消息输入间隔，输入消息分别为 100, 500, 1 000, 2 500, 5 000, 7 500, 10 000, 25 000, 50 000, 100 000，任务完成时间如图 3 所示。

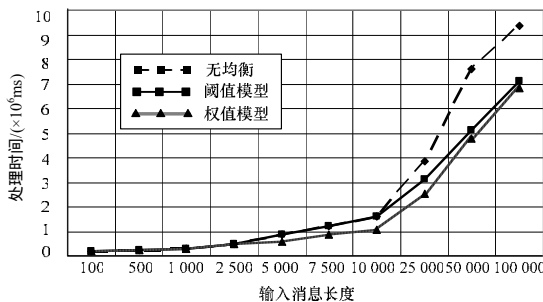


图 3 任务完成时间

可见，当输入消息数量较少($MSG_SIZE \leq 2500$)时，队列处于轻载状态，阈值模型方案和权值模型方案均不对队列进行负载均衡调度，3 种策略处理结果相近；当输入消息增加，队列进入负载适中状态($5\ 000 < MSG_SIZE < 10\ 000$)时，权值模型方案对队列进行负载均衡调度，而由于此时队列负载适中，且未达到阈值模型的调度阈值，阈值方案未进行负载均衡调度，因此权值模型处理效果最佳；当输入消息进一步增加，队列逐渐进入超载状态($MSG_SIZE > 25\ 000$)时，阈值模型方案和权值模型方案均开始对队列进行负载均衡调度，从数据中可以看出 2 种方案均优于不采用均衡调度方案，其中

权值模型方案相对略优。同时由于该方案采用更细粒度基于处理时间的消息权值进行队列负载预测，队列负载状态精确度更高。

图 4 和图 5 分别为输入消息长度为 10 000 和 100 000 时各消息队列对应处理器任务执行时间。

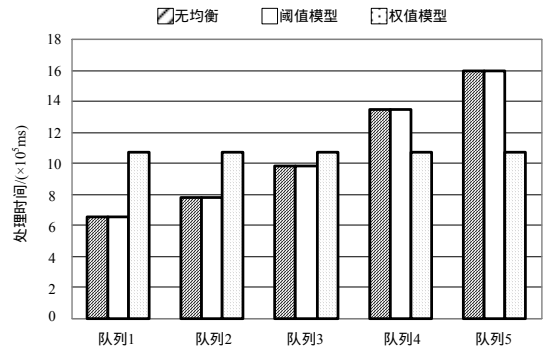


图 4 输入消息长度为 10 000 时任务执行时间

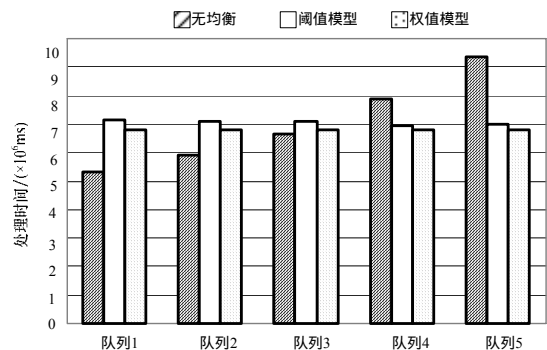


图 5 输入消息长度为 100 000 时任务执行时间

从图 4 可以看出，在队列负载适中情况下，基于权值预测模型均衡调度策略，能保证各个消息队列负载均衡，使各个消息队列对应的处理模块基本完成相近的处理任务，远优于其他 2 种方案；从图 5 可以看出，在队列超载情况下，基于权值预测模型和基于阈值模型的均衡调度策略都能够保证各个消息队列负载相近，其中基于权值模型的调度策略结果相对较优。

3 工程应用

基于上述负载模型和均衡调度策略，本文实现一种基于中心管控机制的互动消息处理系统。系统架构如图 6 所示。

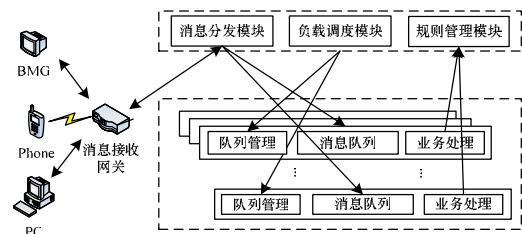


图 6 系统架构

系统采用分层、开放的体系架构，以消息适配、消息分发、消息队列负载均衡调度和消息处理流程控制作为支撑，对接入消息、消息队列和消息处理流程进行统一管理。依据功能在系统中抽象出消息接收网关、消息分发模块、负载调度模块、业务规则管理模块、消息队列、队列管理模块和业务处理模块，分为消息接入层、控制层和消息处理层。

(下转第 37 页)