

# 局间通信负荷监控问题的形式化方法<sup>\*</sup>

赖剑煌

(中山大学数学系, 广州 510275)

**摘要** 利用最新时段演算等理论, 形式化负荷监测功能的告警原理和组合输出问题, 并进行必要的形式化推导, 以确保规范说明具有可理解性和准确性. 还给出了实现监控的核心算法.

**关键词** 负荷监测, CPU 占用率, 时段演算, 形式化方法

**分类号** TP 316

局用交换机的负荷指标— CPU 占用率, 是评估系统负载能力的重要参数, 如果一个程控交换机的话务量忽然增加或者程控交换机出现话务故障, 都可能使交换机处于超负荷状态, 从而使得与该交换机联接的局间通信线路处于阻塞状态, 以至产生不必要的经济损失. 因此国标规定, 作为 C1, C2, C3, C4 的局的交换机必须具有负荷监测功能.

为了实现这一负荷监测功能, 软件开发的规范说明应具有可理解性和准确性. 将形式化的理论和方法用于规范说明, 有利于简明、准确地反映所要处理问题的本质, 并能进行形式化推理和证明<sup>[1]</sup>, 本文引入时段演算等数学方法, 对局用交换机的负荷监控的告警原理和 CPU 占用率组合输出等问题进行形式化定义.

## 1 时段演算

时段演算 (duration calculus) 是 interval temporal 逻辑的扩展<sup>[2]</sup>. 它利用对给定时间区间上的状态函数的积分时段, 来形式化实时、嵌入系统的软件问题. 其基本定义和定理可归纳如下.

**定义 1** 设  $[b, e]$  为任一观察区间,  $P$  为任意一个 (取值为 0 或 1 的) 状态函数,  $P$  的时段定义为

$$\int P = \int_b^e P(t) dt$$

显然,  $\int 1 = e - b$ .

**定义 2**  $[P] = (\int P = \int 1) \wedge (\int 1 > 0)$ .

$[P]$  是 interval temporal 逻辑的一个谓词.

**定义 3**  $[ ] = (\int 1 = 0)$

**定义 4** 设  $B$  和  $C$  是带有自由变量  $b$  和  $e$  的谓词, 定义 interval temporal 逻辑的模式

\* 收稿日期: 1995-10-31 赖剑煌, 男, 32 岁, 副教授

(modality)是“分号”,其语义为:  $A; B [b, e] = \exists m : b \leq m \leq e. A [b, m] \wedge B [m, e]$ .

定义 5 对于任意一个谓词  $B$ ,记:  $\diamond B = \text{true}; B; \text{true}$

定义 6 对于任意一个谓词  $B$ ,记:  $\square B = \neg \diamond \neg B$

时段演算有下列 6个公理.

公理 1  $\int 0 = 0$

公理 2  $\int P \geq 0$

公理 3  $\int P + \int Q = \int (P \vee Q) + \int (P \wedge Q)$

公理 4  $\int P = r + s \Leftrightarrow (\int P = r); (\int P = s)$

公理 5  $([\int P]; \text{true} \vee [\neg P]; \text{true})$

公理 6  $([\int \text{true}; [P] \vee \text{true}; [\neg P]])$

时段演算是一个相对完备的逻辑理论<sup>[3]</sup>,由上述 6个公理出发,可推出它的全部定理.

## 2 问题定义

程控交换机系统本质上可以看做多任务多进程的多 CPU(或单 CPU)系统.在多 CPU 情况下,往往由一个 CPU 负担呼叫处理,即进行呼叫产生检测,号码分析和处理,路由选择,通信信号的形成和发送等等.而其他 CPU 负责非呼叫处理,例如数据库的维护,系统各种性能的监控管理等等.在单 CPU 情形,由同一 CPU 负责两种业务,由于我们关心的是呼叫处理对交换机负荷的影响,故多 CPU 情形可转化为单 CPU 情形考虑.

在操作系统的控制下,各个任务可以周期性使用 CPU.在每一时刻,CPU 或者被某个任务占用,或者处于空闲状态.假设用 0 和 1 分别表示 CPU“闲”和“忙”状态,则 CPU 的状态函数  $f(t)$ 是从(代表时间的)实数域  $R$ 到集合  $\{0, 1\}$ 上的函数.记为

$$f(t) : R \rightarrow \{0, 1\}.$$

状态函数  $f(t)$ 表示每一时刻,程控交换机的所有任务占用 CPU 的情况.程控交换机的任务(TASK)可分为两大类:一类与呼叫处理有关;另一类与呼叫处理无关,而仅与计算机的基本操作控制,数据库的维护,系统各种性能的监控管理有关.

定义 7 与呼叫处理相关的任务的全体,称为 I 类任务,其他的任务称为 II 类任务.

相应地,我们可以分别定义 I 类任务和 II 类任务占用 CPU 的状态函数  $f_1(t)$ 和  $f_2(t)$ ,状态函数  $f(t)$ ,  $f_1(t)$ 和  $f_2(t)$ 显然有下列的性质.

性质 1  $f(t) = f_1(t) \vee f_2(t), \forall t \in R.$

性质 2  $f_1(t) \wedge f_2(t) = 0, \forall t \in R.$

为了评估系统的负荷情况,在实际应用中,将以固定的时间周期  $T$ ,反复测量 CPU 占用情况.即我们将考查(代表时间的)实数域区间  $(t_0, \infty)$ ,该区间被等分成若干个长度为  $T$  的观察区间.假设  $(t_{i-1}, t_i]$  为其中一个观察区间,我们关心的是其上的 CPU 被占用总体情况.

定义 8 在观察区间  $(t_{i-1}, t_i]$  上,分别定义所有任务 CPU 的总忙时、I 类任务 CPU 的总忙时、II 类任务 CPU 的总忙时为  $\int f, \int f_1, \int f_2.$

由上述公理 3 和性质 1, 2, 可以推出性质 3.

$$\text{性质 3 } \int f = \int f_1 + \int f_2$$

定义 9 在一个长度为  $T(T > 0)$  的观察区间  $(t_{i-1}, t_i]$  上, 定义 CPU 占用率 (RATE) 如下:

$$\text{RATE} = \left( \int f \right) / T \times 100\%$$

其中,  $\int 1 = t_i - t_{i-1} = T$ . CPU 占用率实质上是在观察区间上, 所有任务平均占用 CPU 的时间比率, 相应地, 可分别定义 I 类任务的 CPU 占用率 (RATE<sub>1</sub>), II 类任务的 CPU 占用率 (RATE<sub>2</sub>) 为:

$$\text{RATE}_1 = \left( \int f_1 \right) / T \times 100\%, \text{RATE}_2 = \left( \int f_2 \right) / T \times 100\%$$

根据定义 9, 由性质 3 可以推出性质 4.

性质 4  $\text{RATE} = \text{RATE}_1 + \text{RATE}_2$

RATE<sub>2</sub> 是可控制的, 在不做任何数据库的维护和系统各种性能的监控管理的情况下, 程控交换机维持基本的操作控制, 一般需花费大约 5% 的 CPU 占用率, 即可假设 RATE<sub>2</sub> = 5%. 而 RATE<sub>1</sub> 的是随机的, 取决于各个时期话务量的多少. 因此, 局用交换机负荷问题实质上是对 I 类任务的 CPU 占用率的测量与控制问题. 由性质 4 和 RATE<sub>2</sub> ≤ 100%, 可知.

性质 5 在不做任何数据库的维护和系统的监控管理的情况下, RATE<sub>1</sub> ≤ 95%.

### 3 观察区间的选择和组合输出

在应用中, 将以时间周期  $T$  (即固定的观察区间长度  $T$ ), 反复测量 I 类任务的 CPU 占用率, 每获得一个 CPU 占用率, 就判断一次告警条件, 并将其存于数据库中, 以供用户查询.

观察区间大小的选择对局用交换机负荷的监控至关重要. 因为如果观察区间太小, 则取样和计算 CPU 占用率过频, 势必增加系统负荷; 如果观察区间太大, 则不能及时发现系统的超负荷情况, 从而及时地进行控制. 一般情况下, 观察区间长度应为程控交换机任务切换周期的倍数, 取值范围以 1~60 s 为宜.

另一方面, 观察区间太大也会影响用户查询结果的精确度. 而利用小观察区间上的 CPU 占用率, 通过连续  $n$  个区间迭加的方法, 可以重新计算出连续  $n$  个区间的 CPU 占用率.

定理 1 假设区间  $(t_0, t_n]$  可等分割为  $n$  个区间  $(t_{i-1}, t_i]$ , 即:  $t_0 < t_1 < \dots < t_{i-1} < t_i < \dots < t_{n-1} < t_n$ ,  $t_i - t_{i-1} = T (i = 1, 2, \dots, n)$ ,  $R$  和  $R_i$  分别为区间  $(t_0, t_n]$  和  $(t_{i-1}, t_i]$  上的 CPU 占用率, 则有:

$$R = \sum_{i=1}^n R_i / n$$

$$\text{证明 } \because R = \left( \int f \right) / T \times 100\% = \sum_{i=1}^n \left( \int_{t_{i-1}}^{t_i} f_i \right) / T \times 100\% = \sum_{i=1}^n \left( \int_{t_{i-1}}^{t_i} f_i / (n \cdot T) \right) \times 100\%$$

$$\therefore R = \sum_{i=1}^n \left( \int_{t_{i-1}}^{t_i} f_i / T \right) \times 100\% / n = \sum_{i=1}^n R_i / n$$

### 4 数据来源和处理

负荷监控的数据主要来源于操作系统 (OS). 程控交换机一般都有一个任务运行管理子系统, 负责监控系统的各任务的运行, 并分别统计各个任务占用 CPU 的时间. 为了对局用交换机进行负荷统计, 只需周期性地从 OS 获取各个任务占用 CPU 时间, 从中统计出 I 类

任务总忙时,再根据定义 9 计算出 I 类 CPU 占用率.每获得一个新的 I 类 CPU 占用率,都将进行负荷统计,判断是否达到告警条件,或者是否达到解除告警条件,并将 I 类 CPU 占用率数据存于数据库,便于定时跟踪或查询处理.其算法如下.

- (1) 定时从 OS 获取各任务占用的时间;
- (2) 计算 I 任务总忙时和计算 I 类任务 CPU 占用率;
- (3) 进行负荷统计,判断是否达到告警,或者判断是否解除告警;
- (4) 若达到告警条件,则做相应告警控制;若达到解除告警条件,则解除其告警控制;
- (5) 将 I 的类任务 CPU 点用率存于指定的数据库中;
- (6) 重复步骤 1.

## 5 负荷统计的参数

程控交换机超负荷告警可分为两级告警,其中第一级为小告警,第二级为大告警.若  $i = 1, 2$ , 分别代表小告警和大告警,则告警的判断与下列参数有关:

- (1)  $i$  级告警临界值 ( $ALARM(i)$ ): 当 CPU 占用率大于或等于这个值时,就开始进行第  $i$  级告警统计,以判断其条件是否满足.
- (2)  $i$  级告警解除临界值 ( $ALM\_RELIF(i)$ ): 当 CPU 占用率低于这个值时,就开始进行第  $i$  级解除告警统计,以判断其条件是否满足.
- (3)  $i$  级告警最小时间段 ( $ALM\_TIME(i)$ ): 当不低于  $i$  级告警临界值的 CPU 占用率的连续长度不低于这个时间段,则说明第  $i$  级告警条件已达到,可发出第  $i$  级告警.
- (4)  $i$  级告警解除最小时间段 ( $AR\_TIME(i)$ ): 当低于  $i$  级告警解除临界值的 CPU 占用率的连续长度大于或等于这个时间段,则说明第  $i$  级解除告警条件已达到,如果此时正处于第  $i$  级告警状态,便发出第  $i$  级告警解除信息.

其中,可由用户通过数据库的设置命令灵活设定这些参数.图 1 展示了这些参数的意义.设置告警临界值  $ALARM(i)$ ,不能太接近其 CPU 占用率允许的上限,应该为告警出现后的控制程序的运行留些余地.由性质 5 可知:  $ALARM(2) \leq 95\%$ .假设预留 5% 的 CPU 占用率用于负荷控制程序的运行,则  $ALARM(2) \leq 90\%$ ,并且告警参数必须满足下列性质:

性质 6  $90\% \geq ALARM(2) > ALM\_RELIF(2) > ALM\_RELIF(1)$

性质 7  $ALARM(2) > ALARM(1) > ALM\_RELIF(1) > 0$

## 6 负荷监控的告警原理和算法

对于负荷监控而言,最关键的是设计一个恰如其分的告警原理.由于每收到一个 CPU 占用率,都要判断是否达到告警条件,因此这一判断的算法复杂度不能太大,否则将浪费太多的 CPU 时间.

CPU 占用率实质上是实数区间  $(t_0, \infty)$  上的阶梯函数,设为  $h(t)$ .  $(t)$  具有下列性质:

实数区间  $(t_0, \infty)$  可分割成若干长度为  $T$  的小区间  $(t_{j-1}, t_j]$ , 即  $t_0 < t_1 < t_{j-1} < t_j < \dots < \infty$ ,  $t_j - t_{j-1} = T (j \in N)$ , 对  $\forall t \in (t_{j-1}, t_j] (j \in N)$ , 有  $h(t) = h(t_j)$ .

定义 10 定义  $i (i=1, 2)$  级告警点为, 找一个区间  $(t_k', t_k]$  的右端点  $t_k$ , 满足下列条件:

- ①  $t_k', t_k \in \{t_j (j \in N) | t_j \text{ 为阶梯函数 } h(t) \text{ 的分割点}\}$ , 且  $k' \neq k$ .
- ②  $\forall t \in (t_k', t_k], h(t) \geq ALARM(i)$ , 并且  $t_k' = t_0$  或  $h(t_k') < ALARM(i)$

图 1 一级告警及其解除示意图

Fig. 1 An Illustration of The Overload control principle

③  $t_k - t_k' \geq \text{ALM\_TIME}(i)$  且  $t_k - t_k' < \text{ALM\_TIME}(i) + T$

定义 11 定义  $i(i=1, 2)$  级告警解除点为, 找一个区间  $(t_k', t_k]$  的右端点  $t_k$ , 满足下列条件:

①  $t_k', t_k \in \{t_i \mid t_i \in N\}$   $t_i$  为阶梯函数  $h(t)$  的分割点, 且  $k' \neq k$ .

②  $\forall t \in (t_k', t_k], h(t) < \text{ALM\_RELIF}(i)$  且  $h(t_k') \geq \text{ALM\_RELIF}(i)$ .

③  $t_k - t_k' \geq \text{AR\_TIME}(i)$  且  $t_k - t_k' < \text{AR\_TIME}(i) + T$

图 2 提供了负荷监控的告警处理算法. 其思想是: 当系统处于  $i$  级非告警状态, 该算法

图 2 负荷监控的核心算法

Fig. 2 The heart of the overload control algorithm

监测  $i$  级告警条件是否成熟,即寻找一个  $i$  级告警点;当系统处于  $i$  级告警状态,该算法监测  $i$  级告警解除条件是否成熟,即寻找一个  $i$  级告警解除点,其中  $i=1,2$ .

其中  $DURATION(i)$  ( $i=1,2$ ) 为一时间段参数,当系统还未达到第  $i$  级告警时,  $DURATION(i)$  中存放 CPU 占用率实时连续大于第  $i$  级告警临界值 ( $ALARM(i)$ ) 的时间长度;当系统处于第  $i$  级告警阶段时,  $DURATION(i)$  中存放 CPU 占用率实时连续小于第  $i$  级告警解除临界值 ( $ALM\_RELIF(i)$ ) 的时间长度.

致谢:感谢联合国大学国际软件研究所(澳门) Dines Bjørner 所长的邀请和周巢尘教授的悉心指导.

## 参 考 文 献

- 1 朱冰,梅宏,杨芙清.软件开发过程中的形式化方法.计算机科学,1995,22(1): 31~ 37
- 2 Zhou Chaochen, Hoare C A R, Ravn A P. A Calculus of durations. Information Processing Letter, 1991, 40(5): 40, 269~ 276
- 3 Hansen M R, Zhou Chaochen. Semantics and completeness of duration calculus. Bakker, JW de Huizing, C Roever, W - P de (Eds). Real- Time LNCS 600, Theory in Practice, REX Workshop. 1992, 209
- 4 Hehner E C. Whats Wrong With Formal Programming Method? LNCS 497 Advance in Computing and Information- ICC'91, 1991. 2~ 23
- 5 Narayana K T. The Formal Specification of a small bookshop information system. IEEE T- SE, 1988, 14(2): 1089~ 1103

## A Formal Specification of C. O. Switch Platform Overload Control

Lai Jianhuang\*

**Abstract** The CPU occupancy rate is a critical parameter to evaluate the system's loading ability. The overload control of C. O. switch platform is an important function that is required by China MPT. A new approach is presented for formalizing the function specification of the overload control of C. O. switch platform with the aid of the duration calculus theory. The heart of the overload control algorithm is given also.

**Keywords** overload control, CPU occupancy rate, duration calculus, formal specification.

\* Department of Mathematics, Zhongshan University, Guangzhou 510275