

针对弱硬实时系统的 DRM 调度算法

赵 明^{1,2}, 赵 海¹, 张文波^{1,3}, 张浩华¹, 罗桂兰¹, 孙佩刚¹

(1. 东北大学信息科学与工程学院, 辽宁沈阳 110004; 2. 沈阳建筑大学信息与控制学院, 辽宁沈阳 110004; 3. 沈阳理工大学信息与工程学院, 辽宁沈阳 110004)

摘 要: 本文在定义支持多级 QoS 的弱硬实时系统周期任务模型的基础上, 提出基于 RM 调度策略的弱硬实时调度算法 DRM, 它具备可调度判定不等式, 不限定任务的 QoS 参数模式, 并通过在调度的过程中动态调整任务的优先级来反映其紧迫程度. 在系统过载时, DRM 调度算法可以采用 QoS 退化机制, 在保证紧要任务以及其所要求的最低 QoS 执行的同时, 适当降低某些任务的服务等级, 使得更多的任务可以有效运行, 以此来提高系统对负载的适应性. 最后, 本文通过仿真实验, 验证了 DRM 以及 QoS 退化机制的有效性.

关键词: 弱硬实时系统; 调度; QoS; 过载; 退化机制

中图分类号: TP316.2 **文献标识码:** A **文章编号:** 0372-2112(2008)01-0070-06

DRM Scheduling Algorithm for Weakly Hard Real Time System

ZHAO Ming^{1,2}, ZHAO Hai¹, ZHANG Wenbo^{1,3}, ZHANG Hao-hua¹, LUO Gui-lan¹, SUN Pei-gang¹

(1. School of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110004, China;

2. School of Information and Control, Shenyang Jianzhu University, Shenyang, Liaoning 110004, China;

3. School of Information and Engineering, Shenyang Ligong University, Shenyang, Liaoning 110004, China)

Abstract: After defining the model of periodic task with multiple QoS grades for weakly hard real time system, this paper presents DRM weakly hard real time scheduling algorithm based on RM scheduling policy. DRM, Which has inequation for schedulability test, not only supports task with any QoS parameter, but also changes the scheduling priority of task during scheduling to reflect its pressure. When system is overloaded, DRM can adopt QoS degradation mechanism, which enhances the flexibility of system to load by reducing service grade of tasks properly with the guarantee of important tasks scheduling with lowest QoS, so as to make more tasks running validly. At the end of this paper, simulated data is presented to show the efficiency of DRM and QoS degradation mechanism.

Key words: weakly hard real time system; schedule; quality of service; overload; degradation mechanism

1 引言

视频会议、无线通信、实时网络传输等弱硬实时系统^[1-3]的出现, 为实时调度算法提出了新的要求. 在这类实时应用中, 虽然允许任务的部分请求不满足截止期的要求, 但不满足的情况是受到限制的, 一般用弱硬实时规范 (m, k) -firm 约束描述这种实时 QoS (Quality of Service, 服务质量)^[4] 限制, 即该实时任务的每连续 k 个请求中至少有 m 个满足截止期的要求. 传统的适用于软实时系统、硬实时系统的调度理论和调度算法已无法满足这一类实时应用的要求, 基于硬实时和软实时的最优调度算法已证明在弱硬实时系统的应用中不再是最优^[5].

目前, 基于弱硬实时 QoS 的调度算法的研究已经有了一些成果, 主要分为动态调度算法和静态调度算法. 基于弱硬实时 QoS 的动态调度算法中比较有代表性的是基于 (m, k) -firm 失效距离的 DBP (Distances Based Priority) 算法^[4,6] 和采用动态窗口约束的 DWCS (Dynamic Window Constrained Scheduling)^[7] 算法, 这类调度算法均采用动态优先级策略, 不具备可调度判定, 无法对任务满足截止期的情况进行预测, 一旦系统瞬间过载, 所有任务的实时 QoS 都得不到保证. 基于弱硬实时 QoS 的静态调度算法可以对任务未来的执行情况进行预测, 其中最有代表性的是 RM-RTO (Rate Monotonic Red Task Only)^[8], 它采用硬实时方式对任务集的可调度性进行判断, 并基于 RM 调度策略对任务的调度优先级进行静

态分配,但要求任务的 QoS 参数必须为 Skipover 模式,即要求 $m = k - 1$ 。现有的这些调度算法存在的普遍问题是它们只支持固定的、静态的 (m, k) -firm 参数,而且这些参数一旦确定,不能根据系统负载的情况动态变化,这直接影响它们在实时系统中的应用。

针对以上分析,本文提出基于 RM^[9] 调度策略的 DRM(Dynamic Rate Monotonic) 调度算法,它具备可调度判定不等式,支持任意模式的弱硬实时 QoS 参数,在调度的过程中,通过将任务在抢占段和让步段之间切换来动态修改它的调度优先级,以反映其紧迫程度。而且,本文还针对于系统过载时紧要任务的 QoS 保证问题,基于 DRM 调度算法,提出系统过载时的 QoS 退化机制,其基本思想是一旦判断出无法为所有任务提供满足其正常 QoS 要求的服务时,按照任务的优先级,在保证为紧要任务提供最低 QoS 服务的情况下,适当降低部分任务或所有任务的 QoS 服务级别,直至可以服务最多的任务为止,以此来提高系统对负载变化的适应性。

2 弱硬实时系统模型与基本假设

为了方便描述弱硬实时系统中的任务调度问题,对弱硬实时系统中的周期任务模型进行定义。

定义 1 如果一个周期任务要求它的每连续 k 次请求中至少有 m 次在截止期之前完成,就称该任务具备 QoS 参数 $\lambda = (m, k)$ 的弱硬实时约束。

定义 2 一个弱硬实时系统中的周期性任务是一个由 n 个任务 $\tau_i (1 \leq i \leq n)$ 构成的集合 Π , τ_i 是一个由四元组 (T_i, C_i, QS_i, DP_i) 所确定的任务 $(C_i \leq T_i)$, 其中:

T_i : 任务 τ_i 的周期

C_i : 任务 τ_i 的执行时间

QS_i : 任务 τ_i 的 QoS 参数集合, 定义为 $\{\lambda_1, \lambda_2\}$, $\lambda_1 = (m_{i1}, k_{i1})$, $\lambda_2 = (m_{i2}, k_{i2})$, $m_{i1}/k_{i1} \leq m_{i2}/k_{i2}$, 其中, λ_2 为任务 τ_i 的正常服务级别的 QoS 参数, λ_1 为任务在系统过载时采用的退化服务级别的 QoS 参数, 一般将 λ_1 设置为任务可以接收的最低 QoS 级别。一个任务只有在满足最低 QoS 的情况下运行,才是有效的。

DP_i : 任务 τ_i 的退化优先级,当系统过载,无法为所有的任务提供满足其正常 QoS 级别的服务时,需要对系统中部分任务的 QoS 参数进行调整,优先调整退化优先级较低者。

定义 3 弱硬实时系统的任务集 Π 在时刻 t 被系统调度时,每个任务 τ_i 所使用的 QoS 参数为 $\lambda = (m_i, k_i) \in QS_i$, 定义 $\Pi_t = \{\tau_i(i) | \tau_i(i) = (T_i, C_i, \lambda_i)\}$ 为 Π 在时刻 t 的调度映射。

定义 4 弱硬实时系统中任务 τ_i 的有效利用率 $u_e(i)$ 定义为 τ_i 为了满足其 QoS 参数 (m_i, k_i) 的要求所必

须具备的最小利用率。显然, $u_e(i) = C_{m_i}/T_i k_i$, 将所有任务的有效利用率之和称为系统的有效利用率,记为 U_e 。

本文所提出的 DRM 算法及 QoS 退化机制均基于对弱硬实时系统的以下假设:

- (1) 系统中的所有任务均为周期任务;
- (2) 所有任务的相对时限等于各自的周期;
- (3) 任务间以及每个任务的请求之间是独立的;
- (4) 在任何时候高优先级的任务可以抢占低优先级任务的执行,而且不需任何代价。

3 DRM 调度算法

3.1 基本思想

对任务未来执行情况的预测以及对任意 QoS 参数的支持,是一个弱硬实时 QoS 调度算法所必须具备的能力。DRM 就是针对此目标进行的研究,它基于 RM 调度算法,但又不同于 RM 调度算法。

DRM 算法同 RM 算法一样,根据任务的周期确定其调度优先级,并且同样采取周期小的任务优先运行的调度策略。但 DRM 调度算法在分配调度优先级时将一个周期为 T 、弱硬实时约束为 (m, k) 的任务 τ , 看作是 k 个周期为 $T \times k$ 的任务,并以 $T \times k$ 为周期分配 τ 的调度优先级。

RM 算法一旦为某个任务分配了调度优先级,那么该任务的优先级在调度的过程中基本是固定不变的。而 DRM 调度算法将任务的执行分为两段, P (Preempt, 抢占) 段和 Y (Yield, 让步) 段。P 段为任务的抢占段,若任务处于此执行段,表明在某段时间内该任务满足截止期时限的请求次数还没有达到其 QoS 参数的要求,此时,它具备由周期和 QoS 参数决定的较高调度优先级,可以抢占其他处于 P 段但调度优先级较低的任务。Y 段为任务的让步段,任务一旦在某段时间内满足截止期约束的请求数目达到了其 QoS 参数的要求,就进入 Y 执行段,此时,它的调度优先级下降至 P_y , P_y 为所有处于 Y 执行段的任务的优先级,且 P_y 小于所有处于 P 执行段的任务的优先级,因此,该任务无法抢占任何处于 P 段任务的执行。可见,任务在使用 DRM 调度算法进行调度的过程中,其调度优先级是不断变化的。以下给出 DRM 的可调度性分析及可调度判定不等式。

定理 1 任务集 Π 在 t 时刻被 DRM 调度时的映射 $\Pi_t = \{\tau_i(1), \tau_i(2), \dots, \tau_i(n)\}$, $\tau_i(i) = (T_i, C_i, \lambda_i)$ 可被 DRM 调度的充分条件是 $\Pi_t = \{\tau_{11}, \tau_{12}, \dots, \tau_{1m_1}, \dots, \tau_{i1}, \tau_{i2}, \dots, \tau_{im_i}, \dots, \tau_{n1}, \tau_{n2}, \dots, \tau_{nm_n}\}$ 按照 RM 是可调度的,其中, $\tau_j = (T_{ij}, C_{ij})$, $1 \leq i \leq n$, $1 \leq j \leq m_i$, $T_j = T_i \times k_i$ 为任务 τ_j 的周期, $C_j = C_i$ 为任务 τ_j 的执行时间。

证明 DRM 调度算法在设置任务的调度优先级

时,将每个弱硬实时约束为 (m_i, k_i) 的任务 τ_i 当作 k_i 个周期为 $T_i \times k_i$ 的任务 $\tau_{i1}, \tau_{i2}, \dots, \tau_{ik_i}$,其中, τ_{ij} 的周期 $T_{ij} = T_i \times k_i$,执行时间 $C_{ij} = C_i$,即 $\tau_{ij} = (T_i \times k_i, C_i)$, $0 \leq j \leq k_i$,然后,按照周期小者优先执行的RM调度策略进行调度,并且只要满足其中的 m_i 个任务的截止期时限,就满足了任务 τ_i 的QoS参数要求,因此,只要任务集 $\Pi'_i = \{ \tau_{ij} | \tau_{ij} = (T_i \times k_i, C_i), 1 \leq i \leq n, 1 \leq k \leq m_i \}$ 按照RM可调度的,那么任务集 Π_i 按照DRM算法就是可调度的。

证毕。

为了给出DRM算法的可调度判定不等式,引入如下定理。

引理 1 如果系统中具有 n 个独立的、可抢占的周期任务,并且任务的相对时限等于各自的周期,当所有任务的利用率之和 $U = \sum_{i=1}^n u_i$ 小于或等于

$$U_{RM}(n) = n(2^{1/n} - 1)$$

时,该系统的任务集按照RM算法是可调度的,其中 u_i 为任务 τ_i 的利用率^[10]。

定理 2 如果 Π_i 中的 n 个任务有效利用率之和

$$U_e = \sum_{i=1}^n u_e(i) \leq n(2^{1/n} - 1)$$

则 Π_i 按照DRM算法是可调度的。

证明 根据定理1及本算法对弱硬实时系统的基本假设,只要保证 $\Pi'_i = \{ \tau_{i1}, \tau_{i2}, \dots, \tau_{im_i}, \dots, \tau_{i1}, \tau_{i2}, \dots, \tau_{im_i}, \dots, \tau_{n1}, \tau_{n2}, \dots, \tau_{nm_n} \}$ 按照RM可调度即可,计算任务的利用率之和

$$\begin{aligned} U' &= \sum_{i=1}^n \sum_{j=1}^{m_i} u_{ij} = \sum_{i=1}^n \sum_{j=1}^{m_i} C_{ij}/T_{ij} = \sum_{i=1}^n \sum_{j=1}^{m_i} C_i/k_i T_i \\ &= \sum_{i=1}^n m_i C_i/k_i T_i = \sum_{i=1}^n u_e(i) = U_e \end{aligned}$$

根据引理1只要满足

$$U_e \leq n(2^{1/n} - 1)$$

则任务集按照DRM就是可调度的。

证毕。

3.2 算法设计

DRM算法将每个周期为 T 、弱硬实时约束为 (m, k) 的任务当作周期为 $T \times k$ 的任务按照RM策略分配调度优先级 P ,为 $T \times k$ 较小者分配较高调度优先级,然后将等待执行的任务按照设置的调度优先级的升序进行排队,并按照以下规则进行调度:

- (1) 调度优先级最小者优先(RM);
- (2) 若优先级相同,则当前QoS = m'/k' 小者优先;
- (3) 若优先级相同,且当前QoS相同,则 $(k - k')$ 小者优先;
- (4) 否则,先进先出。

DRM算法在调度的过程中,采用 m' 、 k' 跟踪记录任务每连续 k 次请求的完成情况, m' 、 k' 的含义为任务第 k' 次请求之前已有 m' 次在截止期之前完成(初始值分别是0和1),并根据此情况确定是否修改任务的调度优先级 p' ,从而使它进入不同的执行段,具体规则如下:

(1) 一旦任务的某次请求被成功调度运行,按照以下规则更新任务状态。

$$m' = m' + 1; k' = k' + 1;$$

if ($m' = m$) and ($k' \leq k$) then

$$p' = P_y; // \text{进入Y执行段}$$

else if ($k' = k + 1$) then

$$p' = P; m' = 0; k' = 1; // \text{继续P执行段} = \text{其中, } m, k \text{ 为任务本次调度中使用的QoS参数, } P \text{ 为调度算法为任务分配的初始调度优先级。}$$

(2) 一旦任务的某次请求错过截至期,按照以下规则更新任务状态。

$$k' = k' + 1;$$

if ($k' = k + 1$) then

$$p' = P; m' = 0; k' = 1; // \text{进入P执行段}$$

3.3 QoS退化机制

系统瞬间过载时紧要任务的实时QoS保证和较高的系统利用率,一直是弱硬实时QoS调度算法急需解决的问题。现有的算法要么不具备可调度判定,无法对系统过载时任务的执行状况进行预测,要么只支持唯一的QoS参数,即使判断出系统过载,也无法根据负载情况对任务的QoS服务等级进行调整。显然,一个适应的调度算法应该在对任务的未来执行情况进行预测的基础上,判断出系统过载后,在保证紧要任务以其要求的最低QoS运行的同时,适当降低某些优先级较低的任务的QoS服务级别,使更多的任务可以有效运行。因此,本文基于DRM调度算法提出QoS退化机制(QDM, QoS Degradation Mechanism),DRM在开始调度前,首先按照以下步骤使用定理2对任务集的可调度性进行判断,并分配任务的调度优先级:

(1) 按照任务的正常服务级别的QoS参数计算它们的有效利用率之和 U_e ,若 $U_e \leq n(2^{1/n} - 1)$,则根据周期和正常服务级别的QoS参数分配任务的调度优先级,否则转2;

(2) 将任务集 Π 分为两组, Π_1 和 Π_2 , $\Pi = \Pi_1 \cup \Pi_2$ 为空,依次从 Π_1 中取出退化优先级最低的任务加入到 Π_2 中,并以正常级别的QoS参数计算 Π_1 中任务的有效利用率之和 U_{e1} ,以退化服务级别的QoS参数计算 Π_2 中任务的有效利用率之和 U_{e2} ,若 $U_{e1} + U_{e2} \leq n(2^{1/n} - 1)$ 则转3,否则继续2,直至 Π_1 为空,转4;

从表 6 可以看出, 虽然系统过载, 但是采用了 QoS 退化机制的 DRM 调度算法为每个任务提供了满足其最低 QoS 的服务. 尤其是, 由于定理 2 是可调度的充分条件, 因此, 虽然为任务 τ_7 、 τ_8 和 τ_9 配置了尽力而为服务的优先级 P_b , 但是仍为它们提供了最低 QoS 级别的服务.

表 6 时刻 16~ 32 任务最低 QoS 满足情况统计

任务	满足截止期限的请求数	按 QoS 参数应满足截止期约束的请求数	是否满足
1	3	2	满足
2	1	1	满足
3	2	2	满足
4	2	2	满足
5	2	2	满足
6	2	2	满足
7	2	2	满足
8	1	1	满足
9	1	1	满足

4 仿真实验

为了说明 DRM 算法及 QoS 退化机制的有效性, 本文采用 Matlab 对 RM-RTO、DRM、采用 QoS 退化机制的 DRM+ QDM 三种调度算法进行了不同任务负载的仿真实验.

RM-RTO^[8] 同 DRM 一样, 是一个基于 RM 调度策略的弱硬实时调度算法, 它与 DRM 的不同之处在于, DRM-RTO 仅基于任务的周期分配调度优先级, 并采用红、蓝两种颜色对任务的每次请求进行标记, 若任务的某次请求可以错过截止期, 就将其标识为蓝色, 若不可以错过截止期, 就标记为红色. RM-RTO 在调度时优先调度标记为红色的任务. RM-RTO 只支持任务的 QoS 参数的模式为 SkipOver 模式, 即 $(k-1, k)$ 形式, 并且没有提供任何系统过载时的处理措施.

在仿真实验中, 假定系统的负载由数目相等的两类任务组成, 第一类任务的周期为 120ms, 执行时间为 1ms, 正常服务级别以及退化服务级别的 QoS 参数分别为 (7, 8) 和 (3, 4), 第二类任务的周期为 240ms, 执行时间

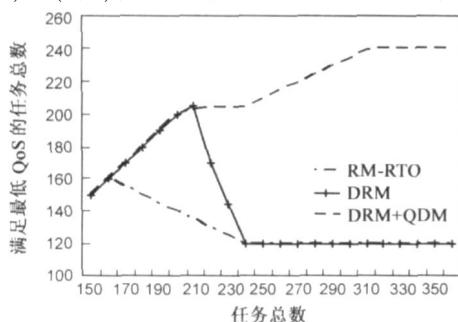


图 3 满足最低 QoS 的任务数与任务总数统计图

为 1ms, 正常服务级别以及退化服务级别的 QoS 参数分别为 (3, 4) 和 (1, 2). 本文分别使用 RM-RTO、DRM 和 DRM+ QDM 在多种系统负载下进行调度, 并对每次调度 0ms~ 960ms 内任务满足最低 QoS 参数的情况进行统计, 实验结果见表 7, 相应统计对比图见图 3 和图 4.

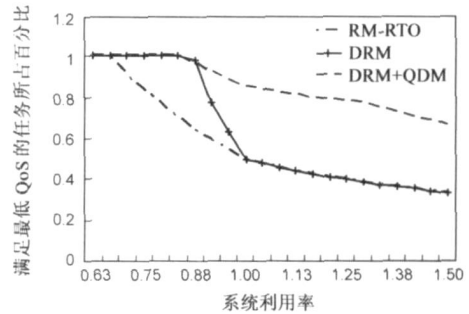


图 4 满足最低 QoS 任务所占百分比与系统利用率的统计图

表 7 仿真实验结果

系统负载		满足最低 QoS 的任务数		
任务数	有效利用率 U_e	RM-RTO	DRM	DRM+ QDM
160	0.6250	150	150	150
170	0.6667	160	160	160
180	0.7083	155	170	170
190	0.7500	150	180	180
200	0.7917	145	190	190
210	0.8333	140	200	200
220	0.8750	135	205	203
230	0.9167	130	170	204
240	0.9583	125	145	204
250	1.0000	120	120	204
260	1.0417	120	120	209
270	1.0833	120	120	214
280	1.1250	120	120	219
290	1.1667	120	120	224
300	1.2083	120	120	229
310	1.2500	120	120	234
320	1.2917	120	120	239
330	1.3333	120	120	240
340	1.3750	120	120	240
350	1.4167	120	120	240
360	1.4583	120	120	240

从图 3 可以看出, 在任务数小于 160, 即系统的有效利用率小于 0.625 时, 在使用三种调度算法的系统中, 满足最低 QoS 的任务数随着任务总数即系统负载的增加而增加; 当任务数达到 160, 使用 RM-RTO 的系统中, 满足最低 QoS 的任务数开始随着任务总数的增长而下降, 直至任务数达到 240, 即系统有效利用率达到 0.9583 后趋于一个稳定的值; 而在使用 DRM 的系统中, 当任务数达到 220, 即系统有效利用率达到 0.875 以后, 满足最低 QoS 的任务数才开始随着任务总数的增

长而快速下降, 直至在任务数达到 240 后趋于一个稳定值; 性能最好的是采用 DRM-QDM 的系统, 由于采用 QoS 退化机制和紧要任务的最低 QoS 保证, 满足最低 QoS 的任务数一直随着任务总数的增加而增加, 直至任务总数增加到 330 以后, 趋于一个系统可以支持的最小值。图 4 给出满足最低 QoS 的任务所占任务总数的百分比随着系统有效利用率增加而变化的情况, 从图 3 和图 4 不难看出 DRM 及 QoS 退化机制的优越性。

5 结论

本文在定义弱硬实时系统的周期任务模型的基础上, 提出支持任意 QoS 参数、优先级动态调整的 DRM 调度算法, 并给出 DRM 的可调度利用率。在系统过载时, DRM 可以采用 QoS 退化机制, 在保证为紧要任务提供满足最低 QoS 的服务的前提下, 最大限度的为所有任务提供有效的服务。仿真实验表明, DRM 调度算法在系统不同负载情况下均优于 RM-RTO 算法, 尤其是采用了 QoS 退化机制的 DRM, 无论系统负载如何, 都能够保证紧要任务以最低的 QoS 进行, 而且使用它的系统中有效任务的吞吐率在系统过载的情况下远大于 DRM 和 RM-RTO。

参考文献:

- [1] Bernat G, Bums A, Liamosi A. Weakly hard real time systems [J]. IEEE Transactions on Computers, 2001, 50(4): 308–321.
- [2] Bernat G. Specification and analysis of weakly hard real time systems[D]. Spain: Universitat de les Illes Balears, 1998.
- [3] Gendy M A E, Bose A, Shin K G. Evolution of the Internet QoS and support for soft real time applications[J]. Proceeding

of IEEE, 2003, 91(7): 1086–1104.

- [4] Hamdaoui M, Ramanathan P. A dynamic priority assignment technique for streams with (m, k) -firm deadlines [J]. IEEE Transactions on Computers, 1995, 44(4): 1443–1451.
- [5] Buttazzo G, Spuri M, Sensini F. Value vs. Deadline scheduling in overload conditions [A]. Proceedings of 16th IEEE Real Time Systems Symposium [C]. Pisa: IEEE Computer Society Press, 1995. 90–99.
- [6] Ramanathan P. Overload management in real time control applications using (m, k) -firm guarantee [J]. IEEE Transaction on Parallel and Distributed Systems (Special Issue on Dependable Real-time Systems), 1999, 10(6): 549–559.
- [7] West R, Zhang Y T, Schwan K, et al. Dynamic window constrained scheduling of real time streams in media servers [J]. IEEE Transactions on Computers, 2004, 53(6): 774–759.
- [8] Koren G, Shasha D. Skip over: Algorithms and complexity for overloaded system that allows skips [A]. Proceedings of Real Time Systems Symposium [C]. Pisa: IEEE Computer Society Press, 1995. 110–117.
- [9] Liu C L, Layland J W. Scheduling algorithm for multiprogramming in a hard real time environment [J]. Journal of ACM, 1973, 20(1): 46–61.
- [10] Liu W S. Real Time Systems [M]. Beijing: Higher Education Press, 2002. 152–156.

作者简介:

赵 明 女, 1977 年 9 月出生于辽宁辽中, 沈阳建筑大学信息与控制学院教师, 目前于东北大学信息与工程学院从师赵海教授攻读计算机应用专业博士研究生。研究方向为嵌入式系统, 当前主要对弱硬实时系统中的调度算法进行研究。E-mail: zhaom@neucera.com