

# 基于系统管理核的龙芯睿频方法<sup>①</sup>



高佳佳<sup>1,2</sup>, 张方舟<sup>3</sup>, 杨嘉勋<sup>2</sup>, 陈华才<sup>2</sup>, 龚声蓉<sup>1,3</sup>

<sup>1</sup>(东北石油大学 计算机与信息技术学院, 大庆 163318)

<sup>2</sup>(江苏航天龙梦信息技术有限公司, 苏州 215500)

<sup>3</sup>(常熟理工学院 计算机科学与工程学院, 苏州 215500)

通信作者: 张方舟, E-mail: [zhangfz@cslg.edu.cn](mailto:zhangfz@cslg.edu.cn)

**摘要:** 针对龙芯处理器调节自身电压会影响 CPU 负载的正常运行, 导致处理器无法运行在高压下进而影响高频的稳定运行问题, 提出一种基于系统管理核 (system management controller, SMC) 的睿频方法, 充分利用系统管理核实时监测并动态调节处理器核的电压和频率。同时, 为保证系统管理核程序运行的实时性以及其与处理器核之间的快速通信, 在其系统程序中搭载 RT-Thread 实时操作系统并设计“Service Request”核间通信协议; 结合动态调频调压模块和自动化温度控制模块, 实现系统管理核对处理器核频率、电压与温度的统一管理。龙芯 3A4000 处理器的实验数据表明, 该方法有效且可靠, 处理器在整体功耗增加了 25.5% 的情况下, 综合性能提升最高可达 34.2%。

**关键词:** 龙芯; 睿频; 系统管理核; 核间通信; 动态调频调压; 温度控制

引用格式: 高佳佳, 张方舟, 杨嘉勋, 陈华才, 龚声蓉. 基于系统管理核的龙芯睿频方法. 计算机系统应用, 2022, 31(7): 120-127. <http://www.c-s-a.org.cn/1003-3254/8597.html>

## Turbo Boost Method on Loongson Platform Based on System Management Controller

GAO Jia-Jia<sup>1,2</sup>, ZHANG Fang-Zhou<sup>3</sup>, YANG Jia-Xun<sup>2</sup>, CHEN Hua-Cai<sup>2</sup>, GONG Sheng-Rong<sup>1,3</sup>

<sup>1</sup>(School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China)

<sup>2</sup>(Jiangsu Lemote Technology Co. Ltd., Suzhou 215500, China)

<sup>3</sup>(School of Computer Science and Engineering, Changshu Institute of Technology, Suzhou 215500, China)

**Abstract:** The voltage adjustment of the Loongson processor will affect the normal operation of CPU load. It causes the processors to fail to run under high voltage and then affects the stable high-frequency operation. Accordingly, the turbo boost method based on the system management controller (SMC) is proposed, which makes full use of the SMC to monitor and adjust the voltage and frequency of processor cores in real time. At the same time, for the real-time operation of SMC as well as the fast communication between SMC and the processor cores, the real-time operating system RT-Thread is equipped and the inter-processor communication protocol “Service Request” is designed. Through dynamic frequency and voltage regulation modules as well as the automatic temperature control module, the unified management of the frequency, voltage, and temperature of processor cores is realized. The experimental data on the Loongson 3A4000 platform shows that this method is effective and reliable. Moreover, when the overall power consumption of processors is increased by 25.5%, the overall performance can be improved by 34.2%.

**Key words:** Loongson; turbo boost; system management controller (SMC); inter-processor communication; dynamic frequency and voltage regulation; temperature control

① 基金项目: 江苏省科技成果转化专项资金 (BA2018074)

收稿时间: 2021-10-20; 修改时间: 2021-11-18; 采用时间: 2021-12-08; csa 在线出版时间: 2022-05-31

处理器设计的中心目标一直是最大限度地利用可用资源以实现最大性能。在多核处理器中,操作情况是不可预测的:在有多个就绪的软件线程情况下,可以通过并行执行更多线程来提高系统性能;在其他情况下,提高单线程性能是改善整个系统性能的唯一途径<sup>[1]</sup>。为了更好地发挥单线程应用的性能,处理器制造商相继提出睿频技术,其中 Intel 的睿频技术为 turbo boost technology<sup>[2]</sup>、AMD 的睿频技术为 turbo core technology<sup>[3]</sup>。睿频技术在动态电压频率调节 (dynamic voltage and frequency scaling, DVFS) 基础上,增加了允许 CPU 频率超过默频的特性。与以前通过降低 CPU 时钟频率和电压来节省能源的 DVFS<sup>[4-6]</sup> 工作相反,当多核处理器具有空闲资源时,睿频技术降低空闲核心的功耗,提高活跃核心的 CPU 时钟频率,侧重于在热设计功耗 (thermal design power, TDP) 范围内最大化处理器性能。

传统商用的睿频技术通过频率和电压调节,以不成比例的功率和能源消耗为代价,实现适度的性能增益<sup>[7]</sup>。睿频由嵌入式硬件控制器的固件控制,除了启用/禁用睿频功能之外,软件几乎无法控制睿频。然而仅依赖固件控制器打开睿频功能,无法最大化睿频的效果。针对上述问题,不少学者在传统睿频的基础上进行研究与改进。Richins 等人<sup>[8]</sup>根据应用程序特性来激活/停用睿频来加速串行和并行代码的执行。Kondguli 等人<sup>[1]</sup>提出自适应睿频技术,设计前瞻线程了解代码行为,将资源投入到具有最佳估计性能收益的核心中。Marques 等人<sup>[9]</sup>将睿频技术与动态并发节流技术 (dynamic concurrency throttling, DCT) 结合来加速顺序代码、平衡线程的并行执行。Zahedi 等人<sup>[10]</sup>提出了一种计算冲刺的架构,将频率和电压提升到超过 TDP 水平,并运行几秒,同时根据应用程序阶段和系统条件决定是否冲刺。Leva 等人<sup>[11]</sup>将冲刺应用作为基于时间/事件的混合系统进行分析,来获得稳定性和性能。

龙芯在处理器动态调频方面也有相应研究。顾丽红等人<sup>[12]</sup>根据龙芯 2F 单核处理器支持软件配置多种不同 CPU 主频的特性,编写 CPUFreq 的驱动代码,实现龙芯软件层的动态调频的方法;在龙芯 3A1000 多核处理器中,因其各个核心频率不能独立调节,陈华才等人<sup>[13]</sup>将动态调频和自动调核结合,实现单独为每个核设置不同的频率。龙芯现有的动态调频均是在默频范围内进行调节,当处理器需要运行在超过默频的高频时,处理器电压需调节到高压,否则处理器将无法稳定

运行在超过默频的高频,从而无法实现睿频功能。然而在龙芯动态调压方面的研究较少。现有的龙芯处理器多数运行在恒定电压,而龙芯 3A4000 处理器若利用处理器核来调节自身电压则会有以下问题:(1)影响 CPU 负载的正常运行:在调频调压过程中,处理器核需实时监测自身的频率和电压的状态,即处理器核直到一次调频调压结束后才会去运行 CPU 负载,而调频调压涉及到硬件操作,完成调频调压的时间无法确定,若系统运行进行多次的调频调压操作,那么处理器核将无法正常运行 CPU 负载;(2)缺少对温度的控制:由于系统将温度和频率割裂管理,当处理器运行在高压高频时,芯片的温度会上升,若此时对温度控制管理不及时,芯片将会面临被烧坏的风险。同时,现有的 Intel、AMD 睿频技术由于处理器指令集和微架构的差异、代码闭源等因素,无法将睿频技术直接从 Intel 或 AMD 平台移植到龙芯平台。针对上述问题,本文根据龙芯 3A4000 设计的特点和 Intel、AMD 平台的睿频思想<sup>[14,15]</sup>,提出龙芯平台基于系统管理核 (system management controller, SMC) 的睿频方法,利用 SMC 实时监测调节处理器核的频率和电压,同时,利用 SMC 监测并控制芯片的温度,最终实现综合考量频率、电压、温度的龙芯睿频技术。

## 1 基于 SMC 的龙芯睿频设计

### 1.1 总体框架设计

睿频技术需要处理器硬件和软件支持。在硬件支持方面,龙芯 3A4000 处理器在硬件上支持多种分级电源状态<sup>[16]</sup>,利用处理器核分频设置寄存器可设置各个核的主频,利用电源管理芯片 (power management IC, PMIC) 可管理处理器核的电压;在软件支持方面,睿频技术基于 Linux 内核的 CPUFreq 子系统<sup>[17]</sup>实现,提供龙芯 3A4000 的 CPUFreq 驱动就可以使用 CPUFreq 提供的调频策略,如 Ondemand、Conservative、Performance 等频率调节策略。

基于 SMC 的龙芯睿频的整体流程如下:首先,龙芯 3A4000 的每个处理器核根据用户接口传递的参数或 CPUFreq governor 中的调频策略来确定其要调节的目标频率并通过 CPUFreq core 模块传递给 CPUFreq driver 模块;接着,处理器核获取到 CPUFreq driver 模块中的目标频率后向 SMC 下达调频命令;最后,SMC 根据命令去调节频率、电压等相关寄存器的值,并将执行完成后的结果反馈给处理器核。同时在 SMC 中加

入温度控制功能,并实时监控处理器核的频率电压温度状况.龙芯 3A4000 基于 SMC 实现睿频的整体框架图如图 1 所示.

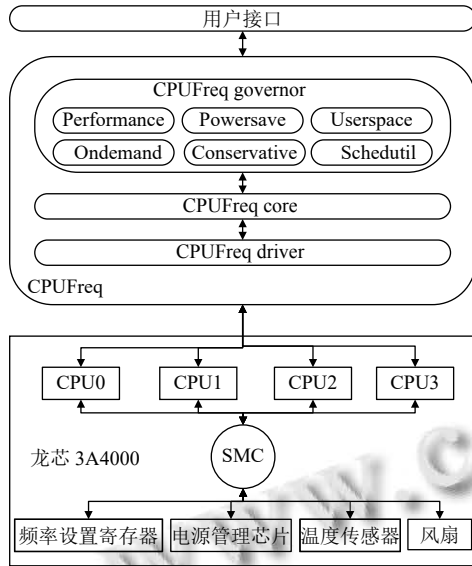


图 1 基于 SMC 的龙芯睿频总体框架图

### 1.2 SMC 的功能设计

由图 1 可知,龙芯 3A4000 基于 SMC 的睿频方法主要分为处理器核部分(CPUFreq 子系统)和 SMC 两部分.其中 SMC 部分需要包含动态调频调压功能、温度控制功能、与处理器核交互功能.

为了更好地在 SMC 上开发功能、管理 SMC 以及保证 SMC 的实时性,本文基于 RT-Thread 实时操作系统的线程调度功能模块化地编写 SMC 的各个功能模块,基于 RT-Thread 系统的 SMC 功能设计如图 2 所示.

图 2 中主要有 3 个线程:采样线程、决策线程和指令线程.采样线程主要运行温度模块来获取处理器核的温度;决策线程主要有 3 部分:根据温度调节风扇转速模块、频率调节模块和电压调节模块;指令线程主要运行指令模块,接收并处理来自处理器核的命令,实现处理器核和 SMC 的交互功能.SMC 的指令模块收到处理器核的命令后,决策线程根据命令对相应模块做出相对应的改变,实现频率和电压的调节功能.同时,采样线程会一直读取芯片的温度,温度过高时会提高风扇转速并降低频率和电压,实现 SMC 对处理器核温度控制的功能.SMC 基于 RT-Thread 实时操作系统循环调度这 3 个线程,实现对处理器核电压、频率、温度的实时监测与调节.

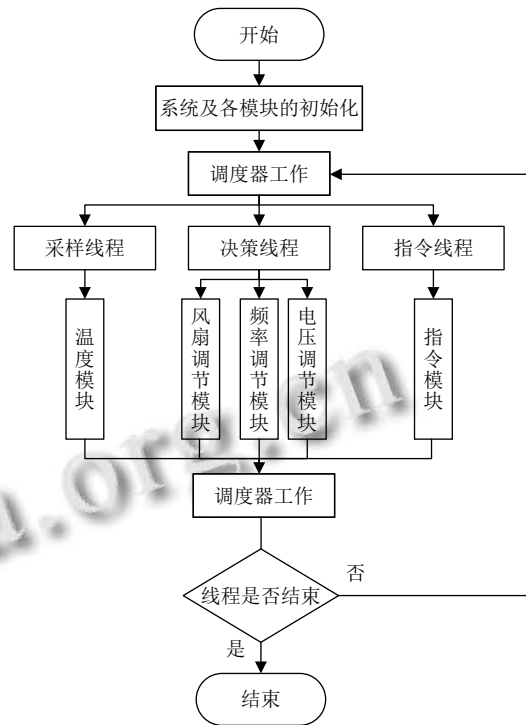


图 2 基于 RT-Thread 系统的 SMC 功能设计图

## 2 基于 SMC 的龙芯睿频实现

### 2.1 调频调压功能

龙芯 3A4000 利用处理器核软件分频设置寄存器对处理器核进行调频设置,该寄存器每 4 位为一个频控域,每个频控域控制一个处理器核,其中频控域的低 3 位为分频系数,提供多种分级频率<sup>[18]</sup>.当龙芯 3A4000 处理器通过睿频将最高频率从 1 800 MHz 变为 2 000 MHz 时,处理器频率被分为 10 个等级,频率表如表 1 所示.根据电压芯片能承受的最大电压来确定处理器能稳定运行的最高频率.

表 1 龙芯 3A4000 频率表

等级	频率 (MHz)	处理器/PLL状态
1	187	Idle
2	375	Idle
3	562	Idle
4	750	Idle
5	937	Normal
6	1 125	Normal
7	1 312	Normal
8	1 500	Normal
9	1 750	Boost
10	2 000	Boost

龙芯 3A4000 通过锁相环 (phase locked loop, PLL) 来调节处理器的频率,每个时钟配置有 3 个参数:



$DIV\_REFC$ 、 $DIV\_LOOPC$  和  $DIV\_OUT$ 。根据 3 个参数经式 (1) 可得出最终的时钟频率:

$$freq = (OSC\_FREQ / DIV\_REFC \times DIV\_LOOPC) / DIV\_OUT \quad (1)$$

其中,  $OSC\_FREQ$  为外部参考时钟频率, 设为 100 MHz。

为了减少对 PLL 和电压的修改, 根据处理器运行的频率将处理器分为 3 个状态: 空闲 (idle) 状态、普通 (normal) 状态和睿频 (boost) 状态, 同时依次从低到高对应着 PLL 的 3 个等级, 如表 1 的第 3 列所示。通过设定最高分频系数和最低分频系数对调频调压的范围进行限制, 其中空闲状态下调节频率的最高等级为 4, 即 750 MHz; 普通状态下调节频率的最高等级为 8, 即 1 500 MHz; 睿频状态下调节频率的最高等级为 10, 即 2 000 MHz。当 PLL 状态改变时, 相对应的时钟参数  $DIV\_REFC$ 、 $DIV\_LOOPC$ 、 $DIV\_OUT$  都需要重新设置。同时, 为了保持请求一致性, HyperTransport (HT)、片上

网络 (network-on-chip, NoC)、共享缓存 (share cache)、静态随机存储器 (static random-access memory, SRAM) 也会进行相应的分频操作。

由于现有一个 PMIC 控制 4 个处理器核的电压, 因此要根据 4 个处理器核中的最高频率来判断 PLL 的最终状态。电压会随着 PLL 状态的改变而改变, 空闲状态时电压为 1 V, 普通状态时电压为 1.15 V, 睿频状态时电压为 1.4 V。当 PLL 状态升级时, 先升压再升频; 当 PLL 状态降级时, 先降频再降压。SMC 通过 I2C 控制 PMIC 实现对处理器核电压的控制。电压值以毫伏 (mV) 为单位, 并经过式 (2) 转换得到最后的数据, 将最后得到的数据写进 uP9512s 电压芯片, 最终实现调节处理器核电压的功能。

$$V = (volt \times 1000 - 210000) / 6587 \quad (2)$$

SMC 调节处理器核的频率和电压的整体流程如图 3 所示。

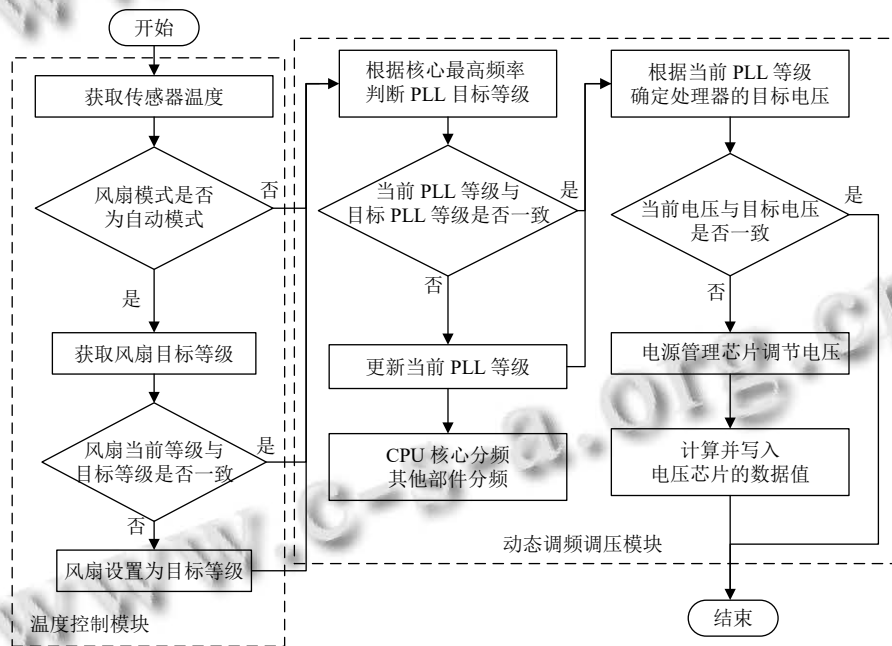


图 3 SMC 的循环调频调压流程图

由图 3 可知, SMC 先获取处理器核温度, 然后根据温度调节风扇的转速, 接着根据 4 个核中的最高频率判断 PLL 等级等信息来得出最终的 PLL 等级, 最后根据 PLL 等级调整处理器核频率电压以及系统其它组件的频率。SMC 的主线程会一直循环运行这个流程, 实现对处理器核频率、电压、温度的实时监测和统一管理。

## 2.2 温度控制功能

温度控制模块是根据传感器获取的温度改变脉冲宽度调制 (pulse width modulation, PWM) 的占空比来调节相对应的风扇转速, 以此实现芯片的降温, 达到保护芯片的目的。

龙芯 3A4000 内部集成两个温度传感器, 可以通过采样寄存器进行观测, 在调频时对处理器核的温度进

行判断,判断是否超过设定的安全温度,若超过则降频.读取的寄存器的值经过式(3)的计算得出最后处理器核的温度.

$$Temp = [(data \& 0xFFFF) \times 731] / 0x4000 - 273 \quad (3)$$

其中,  $data$  为从温度采样寄存器读取的值.处理器工作时的温度范围应为 $-40\text{ }^{\circ}\text{C}$ 至 $125\text{ }^{\circ}\text{C}$ ,即  $Temp$  的取值范围在 $-40\text{ }^{\circ}\text{C}$ 至 $125\text{ }^{\circ}\text{C}$ 之间.

SMC 根据获取到的温度进行风扇转速的调节.龙芯 3A4000 处理器采用自研的 7A1000 桥片,将桥片集成的 PWM 接口和对处理器核的温度监测相结合,并应用于散热器风扇的转速控制上. PWM 调节风扇转速根据不同温度会有不同的占空比从而风扇会有不同的转速,温度越高,占空比越高,风扇转速就越快.利用 PWM 的低脉冲宽度寄存器用来配置 PWM 输出的周期信号的低电平,脉冲周期宽度寄存器用来配置总周期的长度,单位为 PWM 模块的时钟周期.设置总周期长度为 255,通过改变低电平的长度来改变占空比,实现对风扇转速的控制.本文将风扇转速划分为 5 个等级,如表 2 所示.

表 2 风扇转速等级表

等级	温度下限 ( $^{\circ}\text{C}$ )	温度上限 ( $^{\circ}\text{C}$ )	高电平长度
1	-40	15	30
2	15	40	100
3	40	50	150
4	50	70	175
5	70	125	255

风扇有两种模式,一种为人工 (manual) 模式,一种为自动 (auto) 模式,将风扇设置为自动模式,在调频调压的过程中,SMC 会一直根据从传感器获取的温度按照表 2 来调节风扇的转速.

## 2.3 SMC 与处理器核的交互功能

### 2.3.1 核间通信

由于处理器核和 SMC 是相互独立的核,需建立核间通信才能完成命令的传递和结果的反馈.选取处理器核和 SMC 共享的、均能访问到的 32 位邮箱寄存器作为中间传递寄存器并进行核间通信.处理器核将命令内容放在邮箱寄存器中,SMC 的指令模块轮询读取命令并根据命令进行相对应的操作.SMC 处理完成后,将反馈给处理器核的结果放在邮箱寄存器,处理器核轮询读取结果.图 4 为处理器核和 SMC 利用邮箱寄存器进行核间通信的大致过程.

为保证处理器核和 SMC 之间的快速通信,在核间

通信的基础上,设计了一套“Service Request”的通讯协议.在寄存器的内部定义了一个 `smc_message` 的结构体, `smc_message` 由 1 位命令是否完成的标志位 `complete`、7 位命令类型 `cmd` 和 24 位参数 `arg` 组成,以此来建立各种命令.处理器核按照“Service Request”协议向 SMC 下达的命令可分为特征使能、获取频率等级、获取频率信息、获取特征和设置频率等级.其中,当命令为调频时,参数为频率索引.

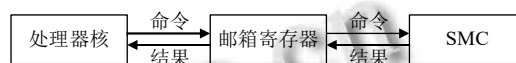


图 4 SMC 和处理器核的核间通信过程

### 2.3.2 处理器核的驱动支持

正如第 1.1 节所述,龙芯 3A4000 处理器实现睿频技术需添加相应的 CPUFreq 驱动支持.在 SMC 功能大致实现后,将龙芯 3A4000 处理器的相关参数提交给 CPUFreq 子系统,往 CPUFreq core 模块注册相应的驱动.在龙芯 3A4000 的 CPUFreq driver 模块中定义 `cpufreq_driver` 类型的 `loongson3_cpufreq_driver` 变量,该变量中需实现的几个具体的功能函数如下:

(1) `init()`: 在注册龙芯驱动时调用,具体的工作是初始化 `cpufreq_policy` 等相关的数据结构;提供 10 级频率表等.

(2) `verify()`: 校验 CPUFreq governor 调频策略提供的目标频率的有效性.

(3) `target_index()`: 在切换频率时调用.根据要调节的目标频率在频率表中找到对应的频率索引.

(4) `get()`: 获取当前 CPU 的频率.

(5) `exit()`: 在驱动退出时调用.

当处理器核根据调频策略确定要调节的目标频率后,经过 CPUFreq driver 中的 `target_index()` 函数指针从表 1 频率表中得出相应的目标频率索引.处理器核将频率索引作为调频命令的参数通过核间通信发送给 SMC.龙芯基于 SMC 的睿频流程图如图 5 所示.

图 5 分成左右两部分,分别对应着处理器核和 SMC.处理器核在向 SMC 下达调频命令后,剩下的调频步骤全都在 SMC 内完成,处理器核只需要轮询读取邮箱寄存器的内容,根据内容判断是否调频成功.此种睿频方法可以使处理器核在不影响正常 CPU 负载运行的情况下进行快速的调频调压.同时,SMC 会一直循环图 3 的调频调压的整个逻辑流程,实时监测并调节处理器核的频率、电压和温度状态.最终在龙芯平台上既实

现了根据 CPU 负载进行动态调频调压又实现了对频率、电压、温度综合考量的睿频方法。

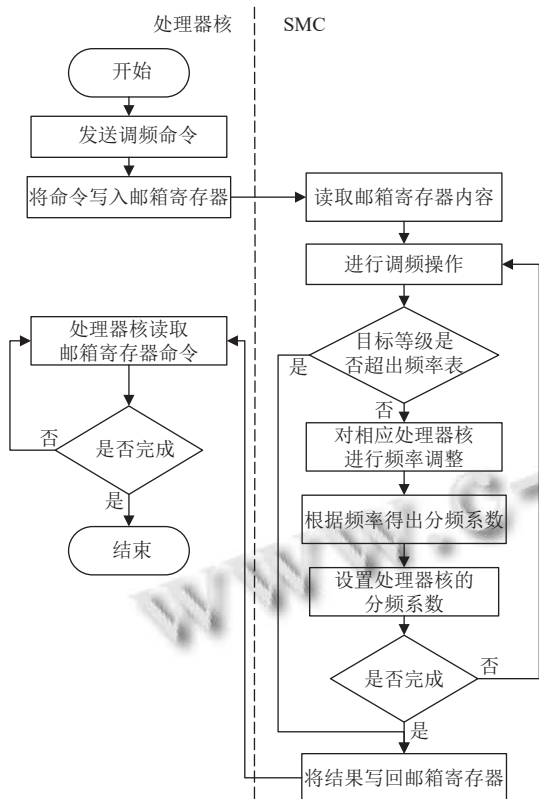


图5 龙芯 3A4000 基于 SMC 实现处理器核睿频流程图

### 3 实验测试与分析

本节通过实验从性能和功耗两方面对本文提出的基于 SMC 的睿频方法进行综合评估。实验平台是龙芯 3A4000 处理器, 操作系统为 Fedora 28, 使用 Linux 内核 5.4.66, 内存为 DDR4 2400 8 GB 内存。龙芯 3A4000 默认的标准频率是 1.8 GHz, 此时处理器核的电压为 1.25 V, 系统运行期间频率和电压都不会改变。当处理器核检测到 SMC 有调频功能时, 在 /sys/devices/system/cpu/cpufreq 文件夹下会生成 boost 文件, 手动将 boost 置 1 后, 处理器的最高频率变为 2 000 MHz, 此时处理器核开启睿频功能, 即基于 SMC 的龙芯睿频方法生效。用“lscpu”命令可查看当前处理器的相关信息, 图 6 为龙芯 3A4000 处理器开启睿频后的处理器信息图。

由图 6 可知, 此时, 处理器支持的最高频率为 2 000 MHz, 最低频率为 187 MHz, 分别与表 1 的第 10 级和第 1 级频率相对应。

实验将现有的没有睿频功能的情况 (频率为默认 1.8 GHz、电压恒为 1.25 V) 和开启睿频功能的情况

(即本文提出的基于 SMC 的睿频方法生效后的情况, 最高频率为 2 GHz、电压最高为 1.4 V) 进行对比。为了更好地分析开启睿频功能后的性能和功耗情况, 又将开启睿频功能的情况细分为采取 Ondemand (按需) 策略的睿频情况和 Performance (性能) 策略的睿频情况。其中 Ondemand (按需) 策略会根据运行时的 CPU 负载动态调节频率, Performance (性能) 策略会让处理器一直运行在最高频。

```
[leote@localhost ~]$ lscpu
Architecture: mips64
Byte Order: Little Endian
CPU(s): 4
On-Line CPU(s) List: 0-3
Thread(s) per core: 1
Core(s) per socket: 4
Socket(s): 1
NUMA node(s): 1
Model name: Loongson-3A R4 (Loongson-3A4000) @ 1800MHz
CPU max MHz: 2000.0000
CPU min MHz: 187.0000
BogoMIPS: 3594.02
L1d cache: 64K
L1i cache: 64K
L2 cache: 2048K
NUMA node0 CPU(s): 0-3
```

图 6 龙芯 3A4000 处理器开启睿频的处理器信息图

实验分为两个部分: 第 1 部分使用 Unixbench 基准程序来获得 3 种情况的综合性能数据; 第 2 部分获得 3 种情况在运行 Unixbench 基准程序时的功耗情况。

#### 3.1 性能测试

本节采用性能测试工具 Unixbench 测试 3 种情况下的龙芯 3A4000 处理器的综合性能。Unixbench 从系统调用、IO 读写、进程、管道、运算等多方面进行性能测试, 能较为全面地综合评价系统各方面性能, 其默认测试集由 dhry、whets、execl、fstime、fsbuffer、fsdisk、pipe、contextl、spawn、syscall、shell1 和 shell8 这 13 个基准测试程序组成, 具体描述如表 3 最后一列所示。

Unixbench 测试默认会运行单路和 4 路并行测试。由于本实验环境中, 一个电压芯片控制 4 个处理器核的电压, 当处理器开启睿频功能并处于满负载时, 处理器频率会调节到最高频, 电压会被调节到最高压, 此时 4 个处理器核的频率均可以调节到最高频率。为了更好地体现睿频后性能提升的效果, 取 4 路并行测试结果进行分析, 使用“./Run -c 4 index”命令对所包含的测试程序进行 4 路测试。龙芯 3A4000 在 3 种情况下运行 Unixbench 性能测试的结果如表 3 所示, 第 2-13 行分别对应着 Unixbench 中的各项性能测试的名称、3 种情况各项性能测试分数以及对性能测试项的描述, 最后一行是 Unixbench 测试得出的 3 种情况的综合性能分数。

从表 3 的最后一组 (system benchmarks index score) 综合性能分数来看, 睿频后处理器无论是运行



CPU 密集型任务 (dhrystone、whetstone、pipe throughput、system call overhead) 还是 IO 密集型任务 (文件系统相关任务 file copy) 还是进程创建、切换相关任务 (excel throughput、pipe-based context switching、process creation、shell scripts) 都可以看到睿频带来的性能提升, 并且处理器无论是采用按需策略还是性能策略, 综

合性能都有一定的提升, 其中按需策略性能提升了约 25.6%, 性能策略性能提升了约 34.2%。因为采用性能策略时处理器一直运行在最高频, 按需策略在任务切换空隙时处理器负载降低, 处理器核运行的频率随之降低, 运行测试程序时为满负载状态, 处理器核的频率又会回到最高频, 所以按需策略相较于性能策略分数略低。

表 3 龙芯 3A4000 处理器 3 种情况下运行 Unixbench 性能测试结果表

测试程序	没有睿频功能 (1 800 MHz)	开启睿频功能 (按需策略)	开启睿频功能 (性能策略)	测试程序描述
dhry	4073.6	5329.4	5398.1	测试整数和逻辑运算性能
whets	1427.8	1903.0	1907.5	测试浮点运算性能
execl	2070.6	2541.5	2696.2	测试每秒钟可执行的execl次数
fstime	585.6	772.8	809.4	测试250 MB文件传输数据的速率
fsbuffer	365.4	488.0	499.5	测试16 MB文件传输数据的速率
fsdisk	1289.8	1607.2	1720.4	测试3.9 GB文件传输数据的速率
pipe	2167.1	2901.4	2975.2	测试每秒钟一个进程向管道写512字节数据并读回的次数
context1	1225.1	1646.3	1693.9	测试每秒钟两个进程通过管道交换数据的次数
spawn	985.2	1088.7	1293.3	测试每秒钟一个进程创建子进程并收回的次数
shell1	2248.7	2256.2	2874.3	测试每秒钟一个进程并发执行一个脚本的一个拷贝的次数
shell8	2173.2	2657.3	2791.5	测试每秒钟一个进程并发执行一个脚本8个拷贝的次数
syscall	1562.3	2093.5	2127.8	测试一次系统调用的代价
system benchmarks index score	1418.5	1781.2	1903.6	综合性能分数

### 3.2 功耗测试

在功耗测试中, 在 3 种情况运行 Unixbench 测试的同时, 利用功率计 66205 测量整机功耗, 并计算得出平均功耗进行对比, 功耗测量结果如表 4 所示。

由表 4 可知, 在空闲状态下, 处理器睿频后采用按需策略时功耗最低, 原因是空闲状态下处理器一直处于最低频率 187 MHz, 与处理器没有睿频功能时一直运行在 1 800 MHz 时的功耗相比, 功耗降低了 33.4%。处理器采用性能策略时运行频率一直在 2 000 MHz, 与其恒运行在 1 800 MHz 频率功耗相比, 略有提升。

在运行 Unixbench 测试程序期间, 处理器开启睿频功能采用按需策略与其采用性能策略相比, 因按需策略不会一直运行在最高频, 所以功耗较低; 但由于 Unixbench 测试大多数情况都是处理器满负载的情况, 所以处理器运行这两个策略之间的功耗相差不大。处理器采用按需和性能策略比其恒运行在 1 800 MHz 时平均功耗分别提高了 23.3%、25.5%。除了频率和电压的升高, 还有散热的问题也会导致整体功耗的提高。

综合第 3.1 节性能测试和第 3.2 节功耗测试结果, 龙芯 3A4000 基于 SMC 的睿频方法与未进行睿频的情况相比, 按需策略满负载时性能提升了 25.6%、功耗提高了 23.3%, 空闲时功耗降低了 33.4%; 性能策

略满负载时性能提升了 34.2%、功耗提高了 25.5%, 空闲时功耗相差不大。由此得出, 龙芯 3A4000 基于 SMC 的睿频方法虽增加了功耗, 但性能提升的效果更明显。在实际情况中, 处理器不会一直运行在满负载的情况下, 使用按需策略动态调节电压和频率, 既可以在高负载的时候提高处理器的频率和电压来提高运算能力, 也可以在低负载的时候降低处理器的频率和电压来降低能耗。用户可以根据自己的需求选择是否开启睿频、选择合适的调频策略, 以此来满足更好的用户体验。

### 4 结论与展望

龙芯 3A4000 之前的处理器无法动态调节自身电压, 这不仅阻碍了频率提升到高频, 也阻碍了能耗的节约。本文提出了基于 SMC 的睿频方法, 让龙芯 3A4000 处理器可以运行在超过默认频率的高频, 在提高电压来维持高频稳定运行的同时, 对处理器芯片温度进行监测保护, 提高了系统可靠性。通过性能测试和功耗测试来验证该方法的可行性, 实验结果表明, 该方法有效且可靠。

未来的研究工作在完善 SMC 温度散热控制、电压细粒度快速调节的同时, 还要完善龙芯平台下的调频策略等, 使龙芯处理器能提升到更高的主频, 达到能耗和性能之间更好的平衡点。

表4 龙芯 3A4000 处理器 3 种情况下运行 Unixbench 的功耗测量结果表

测试程序	没有睿频功能 (1 800 MHz)(W)	开启睿频功能 (按需策略)(W)	开启睿频功能 (性能策略)(W)	测试程序描述
dhry	53.21	73.68	74.83	测试整数和逻辑运算性能
whets	48.66	61.78	62.87	测试浮点运算性能
execl	48.73	65.94	66.22	测试每秒钟可执行的execl次数
fstime	48.97	52.59	52.88	测试250 MB文件传输数据的速率
fsbuffer	48.84	50.67	52.35	测试16 MB文件传输数据的速率
fsdisk	49.36	50.24	53.59	测试3.9 GB文件传输数据的速率
pipe	49.93	64.53	64.68	测试每秒钟一个进程向管道写512字节数据并读回的次数
context1	48.95	65.40	65.92	测试每秒钟两个进程通过管道交换数据的次数
spawn	48.86	58.67	59.71	测试每秒钟一个进程创建子进程并收回的次数
shell1	48.98	59.23	60.62	测试每秒钟一个进程并发执行一个脚本的一个拷贝的次数
shell8	48.76	63.59	65.34	测试每秒钟一个进程并发执行一个脚本8个拷贝的次数
syscall	49.62	66.63	66.91	测试一次系统调用的代价
空闲状态	48.21	32.13	49.85	不运行任何程序时

## 参考文献

- Kondguli S, Huang M. A case for a more effective, power-efficient turbo boosting. *ACM Transactions on Architecture and Code Optimization*, 2018, 15(1): 5. [doi: 10.1145/3170433]
- Kalmbach M, Gottschlag M, Schmidt T, *et al.* TurboCC: A practical frequency-based covert channel with Intel turbo boost. arXiv: 2007.07046, 2020.
- Brookwood N. EPYC(霄龙) 处理器高能效 CPU 设计研究. *中国集成电路*, 2020, 29(11): 31–36. [doi: 10.3969/j.issn.1681-5289.2020.11.007]
- 郭兵, 张鹏博, 沈艳, 等. 基于 BP 神经网络的处理器节能技术研究. *工程科学与技术*, 2018, 50(1): 107–112. [doi: 10.15961/j.jsuese.201700175]
- Zhang W, Zhang H, Lash J. Extending performance-energy trade-offs via dynamic core scaling. *IEEE Transactions on Computers*, 2021, 70(11): 1875–1886. [doi: 10.1109/TC.2020.3029306]
- Scordino C, Abeni L, Lelli J. Real-time and energy efficiency in Linux: Theory and practice. *ACM SIGAPP Applied Computing Review*, 2019, 18(4): 18–30. [doi: 10.1145/3307624.3307627]
- Marques SMVN, Medeiros TS, Rossi FD, *et al.* The impact of turbo frequency on the energy, performance, and aging of parallel applications. 2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration. Cuzco: IEEE, 2019.
- Richins D, Ahmed T, Clapp R, *et al.* Amdahl's law in big data analytics: Alive and kicking in TPCx-BB (BigBench). *IEEE International Symposium on High Performance Computer Architecture*. Vienna: IEEE, 2018. [doi: 10.1109/HPCA.2018.00060]
- Marques SM, Medeiros TS, Serpa M, *et al.* Optimizing parallel applications via dynamic concurrency throttling and turbo boosting. 2021 29th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP). Valladolid: IEEE, 2021: 153–160. [doi: 10.1109/PDP52278.2021.00032]
- Zahedi SM, Fan SC, Faw M, *et al.* Computational sprinting: Architecture, dynamics, and strategies. *ACM Transactions on Computer Systems*, 2017, 34(4): 12. [doi: 10.1145/3014428]
- Leva A, Terraneo F, Cimino C, *et al.* An event-based multi-purpose approach to computational sprinting. *IFAC-PapersOnLine*, 2019, 52: 38–43. [doi: 10.1016/j.ifacol.2019.12.729]
- 顾丽红, 林志强, 吴少刚. 基于龙芯的软件层动态变频研究与实现. *计算机工程*, 2011, 37(9): 41–43. [doi: 10.3969/j.issn.1000-3428.2011.09.014]
- 陈华才, 张福新, 王剑. CPUAutoplug: 动态变频与自动调核相结合的电源管理方法. *小型微型计算机系统*, 2015, 35(11): 2586–2592.
- Zhou L, Zhang FW, Xiao JD, *et al.* A coprocessor-based introspection framework via Intel management engine. *IEEE Transactions on Dependable and Secure Computing*, 2021, 18(4): 1920–1932. [doi: 10.1109/TDSC.2021.3071092]
- Zhang FW, Zhang HW. SoK: A study of using hardware-assisted isolated execution environments for security. *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016*. New York: Association for Computing Machinery, 2016. [doi: 10.1145/2948618.2948621]
- 黄伟. 基于龙芯平台的桌面操作系统电源管理技术的研究与实现 [硕士学位论文]. 南京: 南京邮电大学, 2019.
- 赵婉芳. Linux 系统中 CPUFreq 电源管理模块的探究与应用. *数字技术与应用*, 2018, 36(10): 67–68. [doi: 10.19695/j.cnki.cn12-1369.2018.10.33]
- 陈华才. 用“芯”探核: 基于龙芯的 Linux 内核探索解析. 北京: 人民邮电出版社, 2020.

(校对责编: 孙君艳)