唐金阳, 何升, 杭骁骞, 等. 基于申威平台运行时电源管理研究[J]. 智能计算机与应用,2025,15(1):52-58. DOI:10.20169/j.issn.2095-2163.250108

基于申威平台运行时电源管理研究

唐金阳[1,2], 何 升[2], 杭骁骞[2], 付 雄[1]
(1 南京邮电大学 计算机学院, 南京 210023; 2 无锡先进技术研究院, 江苏 无锡 214062)

摘 要: 运行时的电源管理,是一种在系统运行过程中根据系统负载情况和相应策略,动态控制系统功耗的技术。当前,申威国产处理器平台未能实现该特性,仅支持手动调节处理器频率。为解决该问题,在申威平台上实现了运行时电源管理模型(swDFCS),该模型可以根据负载变化,实时调整处理器的频率和核数,在降低能耗的同时保证了性能需求。实验数据表明,该模型能够以极小的性能损失换取 15%的能耗降低,为国产申威平台下软硬件协同的电源管理研究提供参考,提升申威平台在低功耗场景下的适应性。

关键词: 系统功耗; 处理器频率; 运行时电源管理; swDFCS; 申威平台
中图分类号: TP311 **文献标志码**: A **文章编号**: 2095-2163(2025)01-0052-07

Research on runtime power management based on Sunway platform

TANG Jinyang[1,2], HE Sheng[2], HANG Xiaoqian[2], FU Xiong[1]
(1 School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China;
2 Wuxi Institute of Advanced Technology, Wuxi 214062, Jiangsu, China)

Abstract: Runtime power management is a technology that dynamically controls the system power consumption during system operation according to the system load and the corresponding strategy. The current Sunway domestic processor platform fails to implement this feature and only supports manual adjustment of processor frequency. In order to solve this problem, this paper implements a runtime power management model (swDFCS) on the Sunway platform, which can adjust the processor frequency and number of cores in real time according to load changes, reducing energy consumption while ensuring performance requirements. The experimental data show that the model is able to reduce energy consumption by 15% with a very small performance loss, which provides a reference for the research on the collaborative power management of software and hardware under the domestic Sunway platform and improves the adaptability of the Sunway platformin low-power scenarios.

Key words: system power consumption; processor frequency; runtime power management; swDFCS; Sunway platform

0 引 言

申威处理器是中国自主研发的一款高性能多核处理器,具有完全知识产权,是中国少有采用自主指令集的处理器[1]。申威架构发端于 Alpha,经过不断发展,现已实现指令集完全自主可控[2]。最为经典的应用是神威·太湖之光,曾经在全球超级计算机中排名第一[3]。除了超级计算机,申威处理器还可应用于多核服务器,如申威 3231 服务器[4]。随着处理器核心数量的增加和计算能力的提升,高性能所带来的高功耗问题将对芯片的散热、使用寿命产生重大影响[5]。因此,对申威处理器进行有效的电源管理,尤其是在能耗方面的优化,成为当前研究的重点[6]。本文的研究主要基于申威 3231 型服务器展开,研究申威处理器平台运行时电源管理模型。

计算机常见的电源管理标准是高级电源配置接口(Advanced Configuration Power Interface, ACPI)[7],ACPI 是英特尔公司提出的电源管理标准,广泛应用于 X86 架构。2008 年,英特尔提出睿频技术[8],其原理是允许 CPU 在处理轻负载时以基本时钟频率运行,而在处理高负载时提升至更高的时钟频率[9]。ARM 采用 big. LITTLE[10] 大小架构

作者简介: 唐金阳(1997—),男,硕士研究生,主要研究方向:操作系统; 何 升(1987—),男,硕士,工程师,主要研究方向:计算机系统结构,操作系统; 杭骁骞(1990—),男,学士,工程师,主要研究方向:计算机系统结构,操作系统。
通信作者: 付 雄(1979—),男,博士,教授,博士生导师,主要研究方向:云计算,物联网。Email:fux@njupt.edu.cn。

收稿日期: 2023-07-12

核，其原理是通过将高性能的大核（big core）与能效更高的小核（LITTLE core）相结合，实现在不同计算任务下的性能与能效的平衡。

CPU 运行时电源管理在 Linux 内核中被称为 CPUFreq 子系统，在一些文献中也被称为动态电压频率调节技术（Dynamic Voltage and Frequency Scaling，DVFS）[11]，主要适用于单个 CPU 利用率在 5%～100%动态变化的场景，基本的方法是动态变频和动态变压[12]，CPUFreq 包含框架和驱动两部分。框架与体系结构无关，包含动态调频的核心策略以及何时进行调频；驱动与体系结构相关，执行具体调频操作，并负责调整调频所带来的附加效应。目前，申威平台下的电源管理方案还不完善，处于发展阶段。近几年，申威平台已经构建了基本的 CPUFreq 框架，称为 CPUFreq_debugfs，可实现手动频率调节。不过该框架受限于架构特性，灵活性较差，实际应用意义不大。

针对目前申威平台上电源管理方案的不足，本文提出一种基于申威平台的运行时电源管理模型（Sw Dynamic Frequency and Core Scaling，SwDFCS）。该模型采用 Linux 的 DVFS 接口框架，实现对申威处理器调频调核硬件接口的通用抽象，二者的结合促成了在申威平台的普适性。通过在申威平台完善 CPUFreq 子系统，实现对不同核心频率的动态管理，再通过自动调核算法对系统负载进一步优化。相比于不使用模型的服务器，使用运行时电源管理模型的服务器在性能减少 2%的情况下，可降低 15%的能耗。

1　相关工作

陈道品等[13]针对当前片上系统（Soc）低功耗设计面临的挑战，提出基于多电源域和自适应调压技术的综合多层次低功耗设计方法。该方法利用 DVFS 和自适应电压缩放技术（Adaptive Voltage Scaling，AVS）进一步优化系统性能，显著减少了系统功耗冗余，从而提高系统的能效表现。王益涵等[14]针对目前计算机系统上功耗大等问题，实现了 DVFS 节能软件。该设计先使用 CPUFreq 初始化配置，再根据用户的实际需要选择是否需要调节每个 CPU 的频率。若当前负载较低，甚至可关闭非启动核。申威处理器由于硬件限制，无法在处理器上实现软件无感的核心独立调频及关核。但是动态调核算法恰好为申威实现这一设计提供了条件。

研究者们在硬件和软件方面都进行了尝试和探索，成效颇多。但是一方面，陈道品等[13]的方法需要依赖硬件支持；另一方面，由于申威平台在硬件接口上对 DVFS 支持有限，针对申威平台的电源管理工作存在很大空白。

申威处理器现有的电源管理是一个全局方案，要求所有核的频率只能同时升高或降低，无法单独控制。此外，申威目前仅支持手动频率调节，这种设计无法满足对 CPU 频率的动态管理，不符合运行时电源管理的理念，在实际使用上也存在着灵活性差、能耗高等问题。

针对上述问题，本文在申威平台上实现了首个运行时电源管理模型。一方面，可以为申威平台实现真正意义上的能耗管理；另一方面，希望通过此模型，为国产申威平台软硬件协同的电源管理研究提供参考。

2　CPUFreq 系统框架

CPUFreq 是 Linux 内核的一个子系统[15]，支持在处理器上显式设置核心频率，并提供一组模块化接口来管理 CPU 频率的变化[16]。CPUFreq 包括框架和驱动两个部分，框架属于策略部分，与体系结构无关；驱动属于机制部分，需每个架构各自实现[17]。

CPUFreq 子系统的体系架构如图 1 所示，由 CPUFreq 控制器、体系结构无关的 CPUFreq Core 和各架构专有的 CPUFreq 驱动代码组成。

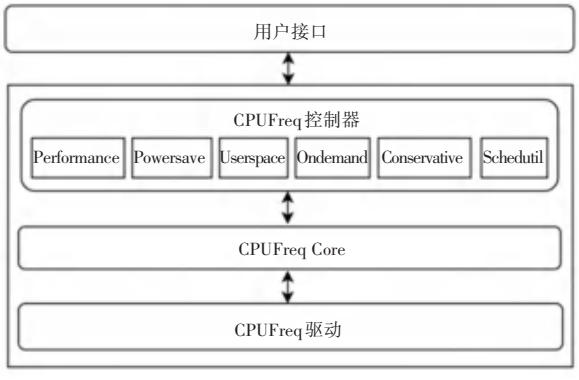

图 1　CPUFreq 子系统体系结构图

Fig. 1　CPUFreq subsystem architecture diagram

CPUFreq 驱动处于 CPUFreq 子系统的最底层，主要功能是实现 CPU 调频驱动；CPUFreq Core 位于 CPUFreq 驱动上层；CPUFreq 控制器是 CPUFreq 子系统的核心，根据当前 CPU 负载自适应调整 CPU 工作频率[18]。最上层为用户接口，若系统提供了 CPUFreq 调频驱动，用户可用对应接口写入调频策略。

CPUFreq 具体策略如下[19]：

（1）Performance：总是将 CPU 设于最高频率。

（2）Powersave：总是将 CPU 设于最低频率。

（3）Ondemand：实时采样，当负载低于阈值时，调节至满足当前负载需求的最低频率；当负载高于阈值时，立即提升到最高频率。

（4）Conservative：改进版 Ondemand，当负载低于阈值时，调节至满足当前负载需求的最低频率；当负载高于阈值时，逐级提升至最高频率。

（5）Userspace：将控制接口通过 sysfs 开放给用户，由用户进行自定义策略。

（6）Schedutil：根据调度器所提供的 CPU 利用率信息进行电压/频率调节，类似于 Ondemand。

3 申威平台下的运行时电源管理

3.1 传统申威电源管理方案

申威平台已形成 CPUFreq_debugfs 框架，目前仅可实现手动调节频率。图 2 展示了目前申威平台 CPUFreq_debugfs 的工作原理。

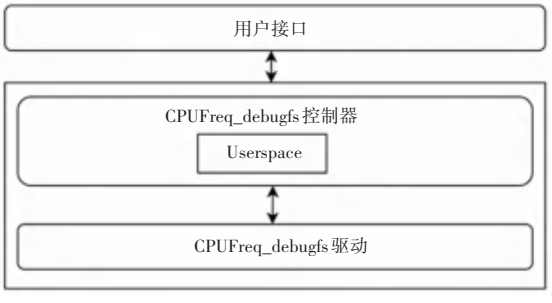

图 2 申威 CPUFreq_debugfs 工作原理

Fig. 2 Sunway CPUFreq_debugfs working principle

CPUFreq_debugfs 工作流程分为两部分：用户接口和内核驱动管理。首先用户根据接口选择申威平台下所支持的 CPU 频率，随后该频率将传给内核驱动判断。若处理器支持该频率的调节，则进入 CPU 调频驱动，由频率控制寄存器对所有核心进行频率调节。待流程完成，该频率将通过内核驱动传给用户接口，告知用户频率设置成功。

CPUFreq_debugfs 允许用户在频率范围内及时切换运行主频，但其控制缺乏灵活性，当前仅支持手动调节。目前，申威现有的电源管理是一个全局方案，意味着频率调整时所有核心将同时升高或降低，导致服务器功率不稳定，增添不必要的能耗。因此，申威处理器还需要一个局部方案来避免所有核心同升同降。

英特尔处理器拥有多个频率控制寄存器控制核心频率，但是申威处理器所需的多个频率控制寄存器均未提供。硬件的缺乏意味着如果在申威上实现运行时电源管理，所有核心频率同升同降的问题无法根本解决，相较于英特尔睿频技术，这一开销无法消除。为此，本文改进了传统申威上的电源管理方案，在申威平台上实现 CPUFreq 驱动，并采用动态调核算法弥补所有核心频率同升同降的消耗。

3.2 基于申威平台的运行时电源管理设计

基于申威平台的运行时电源管理设计主要分为两部分：动态调节频率和动态调整核数。当前申威平台以 CPUFreq_debugfs 为基本框架，因此需要实现符合申威架构规范的 CPUFreq 驱动和动态调核算法，图 3 展示了申威平台下运行时电源管理模型的基本框架。

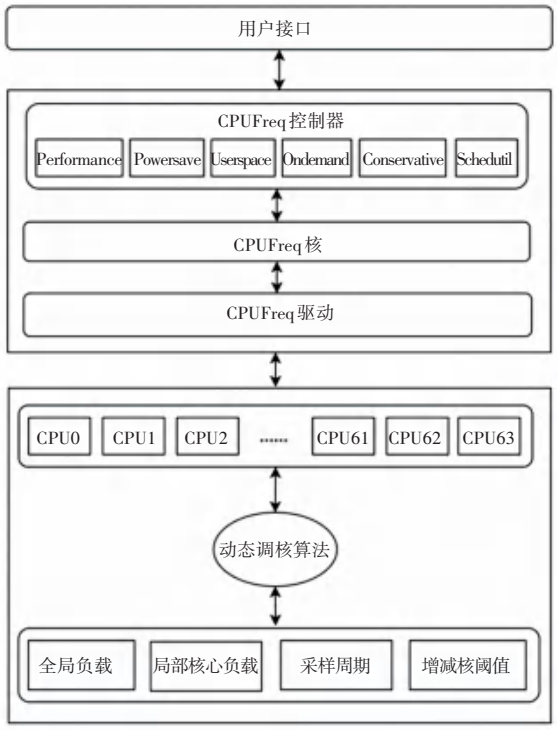

图 3 基于申威平台的运行时电源管理框架

Fig. 3 Runtime power management framework based on the Sunway platform

动态调频需实现符合申威架构规范的 CPUFreq 驱动，并实现对应函数接口。如 init、target_index 和 get 等，以完成频率的初始化、修改、获取等功能，从而实现 Performance、Powersave、Ondemand 等策略。

动态调核的核心思想是以核心全局负载为主要判断依据，同时配合每个核心的局部负载、采样周期、增减核阈值等条件来保留对应核数。

3.3 运行时电源管理在申威下的软件实现

本文在申威 3231 服务器上实现了运行时电源

管理模型。系统启动后，CPUFreq 驱动和动态调核算法将进行初始化操作。首先，调频算法会开启内核定时器，定期计算各个核心负载。当负载超过 Ondemand 策略的默认调频阈值 80%，将频率调整至最高，其余时间将根据当前负载计算频率。与此同时，调核算法将根据系统的全局负载来保留对应核数。

申威处理器使用频率控制寄存器（CLK_CTL）对核心进行频率调节，该寄存器每 7 位为一个频控域，每个频控域控制一路处理器，其中频控域的低 4 位为分频系数，提供多种频率档位。当申威处理器从最高频率 2 500 MHz 降至 200 MHz 时，有 15 个档位可供调节，频率表见表 1。

表 1　申威处理器频率表

Table 1　Sunway processor frequency table

等 级	频率	CLK_CTL[14:11]
1	1 200	0001
2	1 800	0010
3	1 900	0011
4	1 950	0100
5	2 000	0101
6	2 050	0110
7	2 100	0111
8	2 150	1000
9	2 200	1001
10	2 250	1010
11	2 300	1011
12	2 350	1100
13	2 400	1101
14	2 450	1110
15	2 500	1111

申威处理器通过索引以及外部时钟参数来计算频率，最终的频率计算公式为：

（1）当 index = 1 时，freq = CLK×6；

（2）当 index = 2 时，freq = CLK×9；

（3）当 15 ≥ index ≥ 3 时，freq = CLK×9.5 + (Index − 3)×CLK/4。

其中，CLK 为外部时钟，设为 200 MHz，freq 为处理器主频。

整个频率调整的过程分为 3 步：

（1）初始化核心可调频的范围，建立频率索引表并设置索引；

（2）通过索引计算此时需要的频率并调整；

（3）检验该频率是否合法。

在运行时电源管理模型中，调核算法与调频算法相继执行。调核算法主要通过内核调度器来判断核心的全局负载，类似于调频算法中的 Ondemand 策略，最终将选择一个与当前全局负载相匹配的核心数。假设某一核心满负荷运行的负载是 100%，则负载水平的变化就是 0～100%，整个机器的全局负载是（0～100%）×核数。

以 64 核申威 3231 双路服务器为例，核心数最低设为 16 核，则负载变化的范围是 1 600%～6 400%。由于调核算法与调频算法的思想类似，因此本文将调核阈值设为 80%，见表 2。

表 2　动态调核增、减核条件

Table 2　Dynamic reconciliation of the increase and decrease of nuclear conditions

当前核数	增核条件	减核条件
64	无（最大值）	负载 < 6 380%
63	负载 > 6 280%	负载 < 6 280%
62	负载 > 6 180%	负载 < 6 180%
…	…	…
17	负载 > 1 680%	负载 < 1 680%
16	负载 > 1 580%	无（最小值）

由上表可得，当系统计算出全局负载，保留的核心数目也随之确定。

在动态调核过程中，为了减少开销，本文使用了 3 种方法：

（1）在关核之前判断，选择当前负载最小的核心关闭，以减少进程间迁移。

（2）强化增、减核条件，连续 2 次收到增核或者减核的请求后再执行操作。

（3）拉长系统的采样周期，以秒为采样周期，减少采样频率。

申威平台运行时电源管理模型设计如图 4 所示。

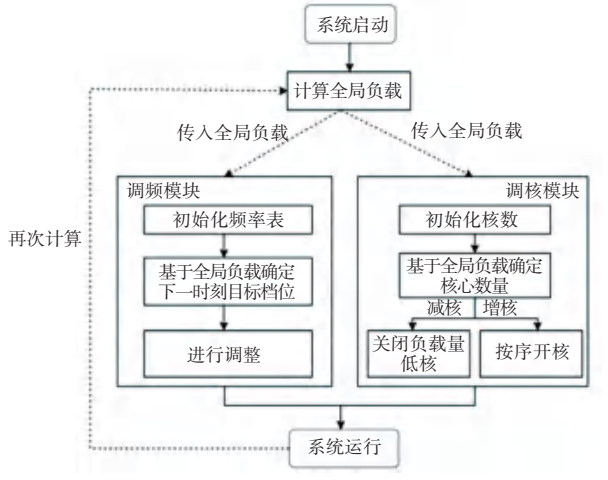

图 4　基于申威平台的运行时电源管理模型设计图

Fig. 4　Design of runtime power management model based on Sunway platform

4 测试与评估

为验证申威平台运行时电源管理模型的调频与调核效果,本节基于申威 3231 服务器从性能与功耗层面进行了验证评估。测试机器采用申威 3231 双路服务器,该服务器有 64 个核心,默认频率为 2.4 GHz,搭载深度操作系统 Uos 20.02,具体参数见表 3。

表 3 实验机器参数配置

Table 3 Experimental machine parameters configuration

软硬件	处理器配置
Architecture	Sw_64
CPU	Sunway 3231 @ 2.4 GHz
核数	64
CPU max MHz	2 500
CPU min MHz	1 200
内存/GB	512
Linux 内核	4.19.180

测试主要对比未采用运行时电源管理模型(默认频率为 2.4 GHz)的服务器与采用运行时电源管理模型(最高频率 2.5 GHz,最低 200 MHz)的服务器。

主要从两个方面进行评估:首先,通过运行 UnixBench 测试程序,获取这两种模型下的性能评分,并比较性能差异;接着,比较在运行 UnixBench 测试程序期间的功耗差异。

4.1 性能测试

本节主要采用性能测试工具 Unixbench 测试两种不同模型下申威 3231 服务器的性能。UnixBench 是一个广泛使用的开源基准测试套件,旨在评估操作系统下的系统性能[20]。该测试软件可以测量多种不同的系统参数,包括 CPU 性能、内存带宽、文件系统性能等等。通过运行 UnixBench,用户可以获得有关其系统性能的详细报告,包括每个测试项目的分数以及总体得分。

UnixBench 测试默认包含单路和 64 路并行测试,在该实验环境中,频率控制器可控制 64 个核心。当处理器使用运行时电源管理模型并处于满负载状态时,处理器频率会被自动调节至最高频率,同时电压也会被调节至最高电压。因此,在该环境下,所有 64 个处理器核的频率均可被调节至最高频率,但考虑到大部分使用场景都不是满负载,因此为了使结果更加符合实际场景,只采用 32 路并行测试结果进行分析。申威 3231 服务器在两种模型下的 Unixbench 性能测试结果见表 4。

从综合分数 Score 看,使用运行时电源管理模型的性能损失有限(约为 2%),有些测试项(如 Fstime、Fsdisk)甚至比未使用模型的机器更好。这是因为采用 32 路并行测试时,系统的负载只有一半左右,未使用模型的机器始终保持 2.4 GHz 和 64 核心运行,而使用运行时电源管理模型的机器在调频调核的过程中会产生性能损耗,因此在总分上会有损失。但是,Fstime 和 Fsdisk 这些测试项会对文件系统和磁盘 I/O 性能进行密集访问,导致系统负载显著增加,此时使用运行时电源管理模型的机器会动态提升频率并启动更多的核心响应,因此性能更好。

表 4 申威 3231 服务器两种情况下运行 Unixbench 测试结果

Table 4 Unixbench test results of Sunway 3231 server running under two scenarios

测试项	未采用模型(分数)	采用模型(分数)	测试项功能
Dhrystone	23 839	23 393	衡量 CPU 整数运算性能
Whetstone	17 707	17 634	衡量 CPU 浮点运算性能
Execl	5 778	5 694	评估操作系统创建和执行新进程的性能
Fstime	607	634	评估文件系统性能,包括文件创建、删除、读取和写入操作
Fsbuffer	373	395	评估文件系统在处理不同大小缓冲区时的性能表现
Fsdisk	1 429	1 466	评估磁盘 I/O 性能,包括文件的顺序读取和写入
Pipe	20 048	19 337	评估管道通信性能
Context1	8 942	8 371	评估单进程上下文切换性能
Spawn	4 218	4 200	评估操作系统在创建新进程方面的性能
Shell1	12 048	11 751	评估同时执行一个 shell 脚本的性能
Shell8	9 449	7 838	评估同时执行 8 个 shell 脚本的性能
Syscall	1 064	1 049	评估系统调用的开销性能
Score	4 518	4 427	整体综合分数

4.2　功耗测试

本节将对比两种不同模型下机器在运行 Unixbench 时的功耗差异，使用 EPM-P E4417A 功率计测量服务器功耗，并计算出平均功耗，功耗测试结果见表 5。

由表 5 可知，使用运行时电源管理模型后，由于系统在空闲状态下一直处于最低频 1 200 MHz，且只有 16 个核心在运行，因此空闲状态下功耗最低；与没有开启模型，始终运行在 2 400 MHz、64 核运行的实验环境相比，功耗只有后者的 40%。

在运行 Unixbench 测试程序期间，系统负载会逐渐上升，但本实验设置的负载为 32 路并行测试，因此系统很少出现满负载情况。在此实验下，除了 Fstime、Fsbuffer 和 Fsdisk 会对文件系统和磁盘 I/O 性能进行密集访问的测试项，使用模型的机器中大部分测试项的功耗都小于未使用模型机器。从表 5 最后一行的平均功耗看，使用模型的机器消耗的功耗比未使用模型的机器降低约 15%。

表 5　申威 3231 服务器两种情况下运行 Unixbench 测试结果

Table 5　Unixbench test results of Sunway 3231 server running under two scenarios

测试项	未采用模型功耗/W	采用模型功耗/W	测试项功能
Dhrystone	78.92	67.25	衡量 CPU 整数运算性能
Whetstone	73.51	66.02	衡量 CPU 浮点运算性能
Execl	74.28	64.85	评估操作系统创建和执行新进程的性能
Fstime	73.68	76.86	评估文件系统性能，包括文件创建、删除、读取和写入操作
Fsbuffer	74.87	77.02	评估文件系统在处理不同大小缓冲区时的性能表现
Fsdisk	75.06	77.56	评估磁盘 I/O 性能，包括文件的顺序读取和写入
Pipe	75.36	65.09	评估管道通信性能
Context1	74.08	62.49	评估单进程上下文切换性能
Spawn	72.98	52.48	评估操作系统在创建新进程方面的性能
Shell1	73.24	58.64	评估同时执行一个 shell 脚本的性能
Shell8	74.11	61.39	评估同时执行 8 个 shell 脚本的性能
Syscall	75.15	63.75	评估系统调用的开销性能
Idle	65.50	26.42	空闲状态下
Average	73.90	62.29	平均功率

4.3　测试小结

综上，申威 3231 服务器在使用运行时电源管理模型后，与未采用该模型的机器相比，性能下降约 2%，但空闲时能耗降低 60%，平均能耗降低 15%。由此得出，基于申威平台的运行时电源管理模型尽管会损失一点性能，但能耗的降低效果更加明显。在实际情况中，服务器很少长时间满负载运行，因此该模型在正常情况下可达到理想的节能目标。

5　结束语

当前申威处理器无法根据当前负载动态调节频率，无法满足绿色节能需求，在笔记本、嵌入式等功耗敏感场景下应用受限。针对此需求。本文提出了基于申威平台的运行时电源管理模型 SwDFCS，并在申威 3231 双路服务器上实现，解决了申威平台只支持手动调节频率的不足。经测试结果表明，SwDFCS 整体性能良好，在性能下降约 2% 的同时可以节省 15%的能耗，这一工作为国产申威平台电源管理的下一步研究提供了有价值的参考。

进一步的研究可从以下方面考虑：

（1）申威处理器需要完善硬件支持，提供多个频率控制器以及电压调节器，以便在灵活调节频率的同时为处理器提供适当电压；

（2）软件算法还可进一步优化，结合其他参数来实施调频调压策略，未来软硬件相结合的申威平台的运行时电源管理模型将更加出色。

参考文献

[1] 成都申威科技有限责任公司. 申威处理器[EB/OL]. (2021-02-07). http://www.swcpu.cn

[2] GAO J, ZHENG F, QI F, et al. Sunway supercomputer architecture towards exascale computing: Analysis and practice [J]. Science China Information Sciences, 2021, 64(4): 141101.

[3] XIAO G, LI K, CHEN Y, et al. CASpMV: A customized and accelerative SPMV framework for the sunway TaihuLight [J].

IEEE Transactions on Parallel and Distributed Systems, 2019, 32(1): 131-146.
[4] 张运德. 异构服务器智能运维管理技术设计与实现[J]. 信息工程大学学报, 2021, 22(6):683-687.
[5] 郜晨, 何升, 杭骁骞. 基于申威 NMII 的锁死故障监测与诊断[J]. 计算机应用研究, 2024, 41(4): 110-119.
[6] MASCITTI A, CUCINOTTA T. Dynamic partitioned scheduling of real-time dag tasks on arm big. little architectures[J]. Journal of Systems and Software,2021,173(5):110886.
[7] MITTAL S. A survey of architectural techniques for DRAM power management [J]. International Journal of High Performance Systems Architecture, 2012, 4(2): 110-119.
[8] LU Y H, ŠIMUNIĆ T, MICHELI G D. Software controlled power management [C]//Proceedings of the Seventh International Workshop on Hardware/Software Codesign. Piscataway, NJ: IEEE, 1999: 157-161.
[9] ROTEM E, NAVEH A, RAJWAN D, et al. Power management architecture of the 2nd generation Intel © Core microarchitecture, formerly codenamed Sandy Bridge [C]//Proceedings of 2011 IEEE Hot Chips 23 Symposium (HCS). Piscataway, NJ: IEEE, 2011: 1-33.
[10] CHUNG H, KANG M, CHO H D. Heterogeneous multi-processing solution of Exynos 5 Octa with ARM big[R]. Samsung White Paper, 2012.
[11] KUMAR H, CHAWLA N, MUKHOPADHYAY S. BiasP: A DVFS based exploit to undermine resource allocation fairness in linux platforms[C]//Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design. Piscataway, NJ: IEEE, 2020: 223-228.
[12] SYDIR J, LI B, MERCATI P, et al. DPM-NFV: Dynamic power management framework for 5g user plane function using bayesian optimization[C]//Proceedings of Globecom 2022-2022 IEEE Global Communications Conference. Piscataway, NJ: IEEE, 2022: 4099-4105.
[13] 陈道品, 武利会, 罗春风, 等. 基于多电源域和自适应调压的 SoC 低功耗研究[J]. 能源与环保, 2022,44(1):183188. DOI: 10. 19389/j. cnki. 1003-0506. 2022. 01. 030
[14] 王益涵, 王凯林, 孙宪坤, 等. 基于 CPUfreq 的 DVFS 节能技术的研究与实现[J]. 计算机测量与控制, 2016, 24(2): 151-154.
[15] 周向军. 基于物联网的智慧电源管理系统设计与应用[J]. 长春大学学报, 2021, 31(4): 27-32.
[16] 赵婉芳. 基于 linux 的 Cpufreq 动态电源管理模块研究[J]. 现代工业经济和信化,2018,8(5):72-73. DOI:10. 16525/j. cnki. 141362/n. 2018. 05. 29
[17] 陈华才. 用“芯”探核:基于龙芯的 Linux 内核探索解析[M]. 北京:人民邮电出版社, 2020: 473-476.
[18] FERREIR A, LIMA J V , RAÏS I, et al. Performance and energy analysis of openmp runtime systems with dense linear algebra algorithms[J]. The International Journal of High Performance Computing Applications, 2019, 33(3): 431-443.
[19] MAHMOUDI M, AVOKH A, BAREKATAIN B. SDN-DVFS: An enhanced QoS-aware load-balancing method in software defined networks[J]. Cluster Computing, 2022, 25(2): 1237-1262.
[20] 惠康,王泽胜,张士宗,等. 通用 CPU 性能基准测试研究综述[J]. 电子学报, 2023, 51(1): 246-247.