因为算法复杂度会随着核数的增长而显著增加，所以在众核处理器高性能运行时进行热管理是一个巨大的挑战。我们在这篇文章提出一个分层的动态温度管理方法来克服这个问题。新方法温度管理依然采用传统的动态电压频率调整技术和任务迁移，采用模型预测控制方法（mpc）做理论依据来实现平滑的任务控制，减少牺牲计算性能。为了扩展到众核系统，分层控制方法设计为两级。在低一级的上，核在空间上被分割成块，块内用现有的功耗分布和用mpc优化得到的功耗分布进行匹配。在高一级上，低一级没有匹配上的功耗进行全局任务迁移。如果高一级的功耗数量很大，我们采用改进的迭代最小割算法来辅助实现任务迁移的策略。最后DVFS用来调节最后没有匹配上的功耗。实验证明，新的方法比现有的其他方法有很大优势，而且在众核处理器中只有很少的性能损耗。

INTRODUCTION

高的操作温度对微处理器的可靠性有负面影响。在新一代的好性能多核众核微处理器中，正在增长的功耗密度和空间上的功耗差距带来了局部热问题，结果导致性能下降，散热开销大，和严重的可靠性问题。所以找到经济和有效地方法去解决高温度问题和同时提高多核和众核芯片的性能和可靠性仍然是一个挑战。

动态温度管理方法是一个提升芯片热相关性能的有效技术。在处理器在高性能水平运行时，动态温度管理通过控制处理器的运行行为，来保证处理器在安全温度以内。动态温度管理一般有两个方法，任务迁移和动态电压频率调整（DVFS），任务迁移方法通过交换多核或者众核处理器上任务来降低芯片的最高温度，而且也可以降低多核系统的能量损耗。所有核都在最大速度运行，所以任务迁移方法能得到更高的性能输出量。但是如果没有其他动态温度管理方法的话，这个可能会有平均温度太高的问题。

动态电压频率调整（DVFS）控制电压和操作频率来调整芯片的温度。最近，DVFS也应用于暗硅领域，DVFS可以保证芯片的温度安全，但是因为频率的变化，芯片的计算性能就要牺牲。

为了使DVFS和任务迁移更有效，控制方案通常采用基于经验的动态温度管理方法。控制器分析热模型，热传感信息等等，然后为动态温度管理技术输出引导行为，例如，DVFS应该被调整到多少频率。许多动态温度管理方法是基于传统控制方法的。但是这些方法并不太适合多核和众核热系统，因为他的复杂性。最近，模型预测方法（MPC）被引入动态温度管理。MPC利用芯片的热模型输出功率上的管理建议。因为这个方法在热行为上进行预测来得到更加有效的控制，所以mpc可以提供更有效和更精确的管理建议。对比于传统的方法，应用MPC有明显的性能提高。

DVFS和任务迁移结合MPC或许可以得到这三种方法的优点。MPC有高质量的控制，任务迁移提升更高的性能，DVFS保证温度的安全。在这方面有许多研究，大都是结合这三种方法中的两种。有的研究基于经验结合任务迁移和DVFS，有的研究结合MPC和DVFS。然而，结合MPC和任务迁移要比MPC结合DVFS更难。最近有研究结合了这三种方法。但是这个方法只能应用于多核微处理器，因为在众核处理器中集成MPC和任务迁移的将引入很大的开销。

在这篇文章中，针对高性能众核微处理器提出了一个新的分层动态温度管理方法。新的方法用模型预测控制来引导包含任务迁移和DVFS的管理过程。为了解决众核系统的执行集成MPC和任务迁移的可扩展性问题，新的方法将任务迁移分成两层，在第一层，相邻的核被分成一块，核功率的二部图匹配在块内执行来进行块内任务迁移。没有匹配的核收集起来做第二层的任务迁移计算。在第二层的迁移结果计算中引入改进的迭代最小割算法来提速升计算速度。新的分层方法对众核处理器来说，只需要很少的开销，而且高度可扩展既能保证高的处理器性能，又能保证温度不超过限制。

**2. 基于动态温度管理的模型预测控制**

在这一章我们将介绍基于动态温度管理方法的模型预测控制方法。结合mpc的热模型在2.1中介绍。2.2节介绍怎么用mpc计算期望的用于引导动态温度管理方法功耗。最后2.3节说明怎么样以mpc计算得到的功耗来引导任务迁移和DVFS。

2.1微处理器的热模型

我们都知道热系统和电路系统是相似的，我们用热阻，热容，和等效的热电流电压源来建立微处理器的热模型。类似于电路系统，l 核的微处理器热模型可以被表达为常微分方程，

3 分层动态温度管理方法

在这一章，针对高性能众核微处理器，提出了新的分层动态温度管理方法。新方法是基于模型预测控制而且采用了任务迁移和DVFS。

众核处理器上执行基于任务迁移的MPC是一个挑战，因为处理器核数非常的大，用复杂度为o3的匈牙利算法计算任务迁移的决策需要花费大量的时间，这个时间可以看成是二部图匹配的时间。为了扩展到众核处理器，新的方法将处理器分成块，在两个层次上计算任务迁移策略：块内（第一层）和块间（第二层）。当前功耗和期望功耗组成的二部图匹配，首先在第一层，也就是块内执行。在第一层内有没有匹配上的功耗，收集起来形成第二层，也就是块间。在第二层，用改进的迭代最小割算法将第二层分成“优化”块。在每个“优化”块执行二部图匹配。第二层最后没有匹配的功耗用DVFS来处理，以保证温度的绝对安全。分层算法通过降低二部图匹配规模降低了计算开销，而且匹配是并行执行的。所以，该算法可以扩展到众核系统。

3.1 第一层块内任务迁移

首先，我们将众核处理器分割成块。作为第一步，我们可以简单根据核的位置分割处理器。这一步分割不需要任何开销。我们通常将方形的块叫普通块，在边缘可能会出现矩形或者小的方形块，把这些称作边缘块。上一章说明了块内功耗匹配的执行过程，这个叫做第一层匹配。对每一块，指定一个块内的核执行匹配的计算。从整个芯片来看，第一层匹配是并行执行的。所以这里的延迟时间只是一个普通块内的匹配时间（注意，边缘块比常规快要小，也就是他们的计算时间并不计算在内）。

可以调整块内的核数以实现整个算法更小的延迟时间：如果核数很大，第一层的功耗匹配将会占用更多的时间，但是可以发现更多的匹配对，将会剩余更少的未匹配功耗到第二层，这样第二层的处理时间就会减少。

图2是一个简单的第一层划分的例子。这是一个100核微处理器，被划分为四个16核普通块（ABDE），五个核数4到8的边缘块（CFGHI）。第一层的功耗匹配将在每一个块内执行。

显然，只执行第一层的功耗匹配不足以找到所有的匹配对。例如，。。。。 对已第一层块内未匹配功耗直接执行DVFS并不是好办法，因为未匹配的功耗可能在其他块找到匹配的期望功耗。这将避免太多不必要的DVFS行为，可能最小化性能损失。所以。。。

3.2第二层块间任务迁移

在上一节，