(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2020 年 4 月 23 日 (23.04.2020)

WIPO | PCT

(10) 国际公布号
WO 2020/078135 A1

(51) 国际专利分类号:
G06F 1/3212 (2019.01)

(21) 国际申请号: PCT/CN2019/104292

(22) 国际申请日: 2019 年 9 月 4 日 (04.09.2019)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201811198978.8 2018年10月15日 (15.10.2018) CN

(71) 申请人: 华 为 技 术 有 限 公 司 (**HUAWEI TECHNOLOGIES CO., LTD.**) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

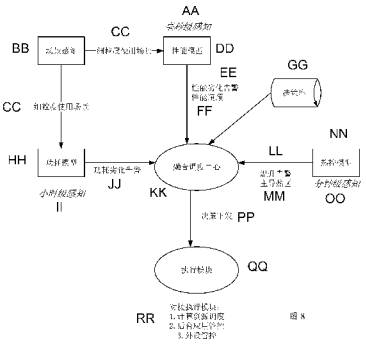
(72) 发明人: 陈秋林 (**CHEN, Qiulin**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 周振坤 (**ZHOU, Zhenkun**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 徐羽琼 (**XU, Yuqiong**); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(81) 指定国 (除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国 (除另有指明，要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,

(54) **Title:** RESOURCE SCHEDULING METHOD AND COMPUTER DEVICE

(54) 发明名称: 资源调度方法和计算机设备



AA Millisecond-level sensing
BB Scenario sensing
CC Fine-grained use scenario
DD Performance model
EE Performance alarm
FF Performance bottleneck
GG Policy database
HH Power consumption model
II Hour-level sensing
JJ Power consumption alarm
KK Integrated scheduling center
LL High temperature alarm
MM Primary heat-conducting region
NN Heat control model
OO Minute-level sensing
PP Issue policy
QQ Execution module
RR Mechanisms corresponding to execution module:
1. resource scheduling calculation;
2. background application management;
3. peripheral management

(57) **Abstract:** The present application provides a resource scheduling method and apparatus and a computer device. The method is used to determine and implement a resource management policy so as to improve overall user experience. The overall user experience is determined by means of the weights of three factors in a current usage scenario of the device: performance, temperature, and power consumption, and the user experience for each of the three factors. The present scheduling method combines the three factors of performance, temperature, and power consumption, so as to prevent "seesawing" caused by single-factor-based scheduling, thereby improving the overall user experience regarding the device.

(57) 摘要: 本申请提供一种资源调度方法、装置及计算机设备等。该方法以提高用户的整体体验为目标确定并执行资源管控策略，其中，用户的整体体验由设备当前使用场景下性能、温度和功耗三者的权重以及三者各自的用户体验确定出来。采用这种融合性能、温度和功耗三种因素的调度方法，能够避免单一因素调度可能带来的跷跷板效应，提升用户对设备的综合体验。

WO 2020/078135 A1

[见续页]

RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

资源调度方法和计算机设备

技术领域

本申请涉及计算机技术，尤其涉及一种资源管理或者说资源调度的方法、装置、计算机设备、计算机存储介质和计算机程序产品等。

背景技术

以智能手机为代表的终端设备中，随着芯片性能的高速发展，终端性能大幅提升，然而电量（相当于待机时长）和发热成为降低用户体验的核心问题。如何有效地平衡性能、功耗和发热成为提升终端设备的竞争力的核心。

以安卓（Android®）手机为例，当前是性能、功耗、热三者的管控独立进行。在性能管控方面，中央处理器（central processing unit, CPU）、图形处理器（graphic processing unit, GPU）、双倍速率（double data rate, DDR）存储器的频点仅基于历史负载进行调整。以CPU为例，当某个任务的CPU负载上升时， CPU频点上升。在功耗管控方面，针对不同的应用类型（如即时通讯类应用）对后台任务的执行时间和功耗消耗进行限制；或者根据整机剩余电量进行分级的功耗管控，例如剩余80%和20%电量对应不同的功耗管控级别。在热管控方面，根据终端设备上各个热区是否超过阈值做热源（如SOC、显示器背光灯）限制，以降低热源的频点、档位为代价限制温度上升。可见，性能、功耗、热是根据各自维度的劣化程度，独立进行调度，容易造成跷跷板效应，例如，电量充足时，优先调控性能，导致电量迅速降低，手机持续发热；电量不充足时，强制管控功耗和热，导致性能降低，影响用户体验。

发明内容

以下从多个方面介绍本申请，容易理解的是，该以下多个方面的实现方式和有益效果可互相参考。

第一方面，本申请提供一种资源调度方法，该方法包括确定设备当前的应用场景，其中，所述应用场景与用户的使用或设备的运行状态相关，运行场景通常包括以下多个维度中的任意一个或多个维度的信息：时间、空间、前台应用的类型以及应用内的使用场景；针对该应用场景，性能、功耗和温度分别有各自的权重，基于三者的权重以及三者目前的用户体验确定资源管控策略；根据所述资源管控策略调度所述设备的资源。

其中，性能、功耗和温度分别意味着三者各自的用户体验。用户体验有多种实现方法，比如对于功耗而言，用户体验的主要指标可以是待机时长，那么预估的待机时长越长，则功耗的用户体验越高；对于性能而言，用户体验的指标可以是视频的目标帧率、流畅度等，流畅度越高，用户体验越高；对于温度而言，用户体验的指标可以是手机外壳温度，手机外壳温度越高，则用户体验越低。性能、功耗和温度三者的权重分别指示三者对用户整体体验的影响程度。用户整体体验例如可以是性能、功耗和温度各自的用户体验的加权和。在确定资源管控策略的过程中，融合性能、功耗和温度三者各自的用户体验以形成用户的整体体验，以用户的整体体验为目标确定资源调度策略。

性能、功耗和温度，这三者很难融合，比如提升性能，功耗就会变高，同时温度就会上升，因此很难找到最合适的资源调度策略。而本申请提供的方法考虑到了不同的应用场景下三者对用户的重要性是不同的。针对特定的应用场景，融合性能、功耗和温度三者的权重确定合适的资源管控策略，然后根据该资源管控策略调度资源，从而实现了三种因素综合考虑的融合决策过程，有效避免了三者独立调度带来的跷跷板效应。同时，本申请提供的方法将性能、功耗和温度三个维度的用户体验满足程度的加权和作为用户整体体验，始终将用户整体体验的提升作为调度目标，有效避免了仅提升一个或两个维度的用户体验，而另一维度的体验下降过多，从而造成的用户整体体验降低的问题。

在一些实现方式中，资源管控策略的确定是周期性的，间隔时间段可以通过机器学习获得，也可以由设备的用户设置，或在设备出厂时预置在设备内部。

在一些实现方式中，资源管控策略可以由事件触发。具体的，当性能劣化程度、功耗劣化程度以及温度劣化程度中的任意一个或多个达到预设条件时，触发所述资源管控策略的确定。其中，劣化程度用于指示目标指标与实际指标之间的差距。通过事件触发资源管控策略的确定（或者说更新），能够更加及时地发现设备的问题，并尽快提供新的资源管控策略。

在一些实现方式中，性能劣化程度用于指示所述设备的性能参数值与目标性能指标之间的差距，所述目标性能指标为满足所述应用场景下的性能体验的一个或多个性能参数的值；所述功耗劣化程度用于指示预测的可用电量与最低可用电量之间的差距，所述温度劣化程度用于指示预测的外壳温度和温度阈值之间的差距。

在一些实现方式中，所述目标性能指标由以下方式确定：在预先设置的数据库中搜索与所述应用场景匹配的目标性能指标，所述数据库中存储有多种应用场景以及多种目标性能指标的对应关系。该数据库可以存在本地，也可以存在其他设备上。例如本设备是手机，数据库可以存在手机上，也可以存在云端服务器上。在其他一些实现方式中，目标性能指标也可以通过在线估计的方式获得，或通过云端服务器计算再返回结果。

不同的应用场景对应不同的目标性能指标，通过在数据库中预先存储这种对应关系，需要时即可查询，从而快速地确定当前应用场景的目标性能指标，进而提升性能劣化程度的确定速度。

在一些实现方式中，设备的性能参数值采集得到的设备当前的性能参数值。在另一一些实现方式中，是预测得到的下一个时刻的性能参数值，具体的，根据所述应用场景和所述设备当前的系统状态可以预测得到设备的性能参数值。所述系统状态指示所述设备当前的资源使用情况，例如资源供给信息和系统负载信息。以上两种方式可以都执行，之后在获得的两组性能参数值中选择一个，或者取两者的平均值，或加权处理等，最终获得一组性能参数值与目标性能参数比较，以获得性能劣化程度。

在一些实现方式中，功耗劣化程度是由所述设备的电量使用规律以及所述设备的剩余电量确定的，其中，所述电量使用规律反映历史上的充电情况和耗电情况（或者说所述电量使用规律代表所述设备电量增加和消耗的历史）。所述电量使用规律包括充电周期、用电规律和应用使用规律。其中，所述充电周期代表历史上的充电间隔，例如充电开始时刻到下一次充电开始时刻、或充电结束时刻到下一次充电结束时刻、或充电结束时刻到下一次

充电开始时刻、或充电开始时刻到下一次充电结束时刻等。所述用电规律指示历史上多个时刻以及每个时刻的剩余电量。所述应用使用规律反映的是历史上应用对电量的消耗信息，可以包括多种信息，例如常用应用、单个应用的平均使用时长、单个应用的平均使用功耗和/或应用使用时所述终端设备所在的位置等反映应用的使用时长和使用功耗的信息。

换句话说，本申请提供了一种方法，利用该方法学习用户的电量使用规律，包括充电规律和应用对电量的消耗规律等，然后根据用户的电量使用规律来判定功耗可能的劣化程度，而非仅预置一些固定的待机时长阈值。这种方式使得功耗劣化程度的确定符合用户用电习惯，从而更加准确和有效。

在一些实现方式中，所述外壳温度是根据所述设备的当前外壳温度、所述设备内多个器件各自的温度升高预测值确定的。由于每个器件对设备外壳温度提升的贡献程度不一样，所以在计算外壳温度时通常还需要考虑每个器件温度的权重，该权重用于指示对外壳温度提升的贡献程度。

需要说明的是，这里的“器件”也可以理解为“热区”。一个热区可以包含一个或多个器件，温度衡量可以以单个器件为最小单元，也可以以热区为最小单元。热区是哪些，通常以设备内部预先设置的为准。

在一些实现方式中，温度劣化程度的计算中用到的温度阈值是根据所述设备当前所处的环境的温度确定的。环境温度的高低会影响用户的触感温度，所以用环境温度来决定温度阈值，更能贴合设备用户实际的温度体验，从而使得温度劣化程度的计算更加准确。

在一些实现方式中，所述资源管控策略包括以下三项中的任意一项或多项：计算资源调度策略、应用管控策略以及外设管控策略。资源管控策略是全方位的。

在一些实现方式中，所述资源管控策略的确定过程利用机器学习或动态规划的方法。充分利用机器学习等数学方法加快资源管控策略的确定过程。

在一些实现方式中，通常所说的设备的应用场景一般包括时间维度、空间维度、以及应用类型维度，比如 14:00 在家看视频应用。但是应用场景还可以更加细粒度。应用场景还可以包括应用内的使用场景。应用内的使用场景包括以下两项信息中的一项或两项：用户对应用的操作行为和应用内的进程行为。例如在微信（WeChat®）上发朋友圈或看视频、在玩游戏的团战等。同样是玩游戏，可能单人作战和团队战场景对各个维度的需求是不一样的，细粒度的应用场景的划分能够使得用户体验目标的确定更加准确，从而使得确定出的资源管控策略更加有针对性，也更加有效。

在一些实现方式中，在确定资源管控策略的过程中，首先根据设备的性能瓶颈或主导热区确定候选的资源管控策略，然后再从候选的资源管控策略中确定待执行的资源管控策略。所述候选的资源管控策略能够提升所述性能瓶颈或降低所述主导热区的温度。通过这种筛选的方式可以减小资源管控策略的搜索范围，降低计算量，达到提升方法效率的目的。

在一些实现方式中，确定所述性能、功耗和温度各自的权重的因素包括以下任意一个或多个：所述设备的应用场景、所述性能、功耗和温度各自的劣化程度、以及所述性能、功耗和温度劣化判断的作用时域，作用时域例如可以是毫秒级、小时级或分钟级。

在一些实现方式，该方法中涉及的一些复杂计算，例如资源调度策略的确定过程，可通过专用处理器实现。专用处理器可作为协处理器，在主处理器的调度下执行。专用处理

器包括但不限于神经网络处理器、机器学习处理器等。另一方面，本方法中的任意步骤，即使不复杂，也可以通过专用处理器或微处理器或协处理器处理器。

第二方面，本申请提供一种资源调度方法，该方法包括：确定设备当前的应用场景；根据所述应用场景，融合性能、功耗和温度三者各自的用户体验以获得用户整体体验；根据所述用户整体体验确定并执行资源管控策略。

在一些实现方式中，根据性能、功耗和温度三者各自的用户体验以及三者在所述应用场景下分别的权重获得用户整体体验，例如三者各自的用户体验的加权和作为用户整体体验。其中，所述权重用于表示每个因素对用户整体体验的影响程度，在不同的场景下可能存在不同，例如游戏场景下用户更关注性能，所以性能的权重高于功耗和温度。

在一些实现方式中，根据性能劣化程度、功耗劣化程度、温度劣化程度分别确定所述性能的体验值、所述功耗的体验值和所述温度的体验值，其中，所述体验值用于指示用户体验，所述劣化程度用于指示目标指标与实际指标的差距，劣化程度越高，对应的体验值越低。然后根据所述性能、功耗和温度的三个体验值以及三者在所述应用场景下分别的权重确定用户整体体验。

第二方面的其他实现方式以及具体实现方式均可参考前述第一方面，在此不再赘述。

第三方面，本申请提供一种资源调度方法，该方法包括：基于用户的指令确定设备当前的应用场景；根据所述应用场景，融合性能、功耗和温度三者各自的用户体验以获得用户整体体验；根据所述用户整体体验确定并执行资源管控策略。

换句话说，本申请提供的方法可以在设备中处于默认关闭状态，由用户通过指令开启。

在一些实现方式中，设备的应用场景识别可以默认开启，但后续融合调度的方法可以由用户指令开启。进一步的，性能、功耗和温度三种模型可以分别设置开启指令，用户可以根据需求开启其中的任意一个或两个。

第三方面的其他实现方式以及具体实现方式均可参考前述第一方面或第二方面，在此不再赘述

第四方面，本申请提供一种计算机设备，例如终端设备，该设备包括处理器和存储器。所述存储器用于存储程序。所述处理器用于读取并执行所述程序并实现如前述任意一个方面或实现方式所描述的方法。

第五方面，本申请提供一种资源调度装置（或称之为资源管理和控制装置），该装置包括一个或多个模块，该一个或多个模块用于实现前述任意一个方面或实现方式所描述的方法。

第六方面，本申请提供一种计算机程序（或计算机程序产品），该计算机程序（或计算机程序产品）中存储程序，该程序再被一个或多个处理器读取并执行后实现如前述任意一个方面或实现方式所描述的方法。

第七方面，本申请提供一种计算机存储介质，可以是非易失性的。该计算机存储介质中包含程序，该程序再被一个或多个处理器读取并执行后实现如前述任意一个方面或实现方式所描述的方法。

附图说明

为了更清楚地说明本申请提供的技术方案，下面将对附图作简单地介绍。显而易见地，下面描述的附图仅仅是本申请的一些实施例。

图 1 为本申请提供的智能手机的层次划分示意图；

图 2a 为本申请提供的资源管理和控制装置的逻辑结构示意图；

图 2b 为本申请提供的资源调度方法的流程示意图；

图 3 为本申请提供的资源管理和控制装置的另一种逻辑结构示意图；

图 4 为本申请提供的性能模型的执行过程示意图；

图 5 为本申请提供的用户用电规律的一个示例；

图 6 为本申请提供的温度模型的执行过程示意图；

图 7A 和图 7B 为本申请提供的环境温度计算方法的流程示意图；

图 8 为本申请提供的资源调度方法的整体流程示意图；

图 9 为本申请提供的应用场景的维度示例；

图 10 为本申请提供的一种涉及策略筛选和反馈调整的流程示意图

图 11 为本申请提供的一种计算机设备的逻辑结构示意图；

图 12 为本申请提供的一种 NPU 的逻辑结构示意图。

具体实施方式

现有的性能管控机制，例如 CPU 的能量感知调度（energy-aware scheduling, EAS)和基于频点调节的 governor 机制，仅仅基于负载，存在性能供给过多，导致功耗和热控劣化的问题。以视频播放为例，视频播放的帧数不是固定的 60 帧，而是 24 或 25 或 30 帧。当前的性能管控机制对前台计算资源进行供给时没有考虑不同的帧数，存在资源供给过高导致功耗过高的问题。另一方面，CPU、GPU、DDR 器件调度策略各行其是，互相影响，容易导致跷跷板效应。例如，电量充足时，优先调控性能，导致电量迅速降低，手机持续发热；电量不充足时，强制管控功耗和热，导致性能降低，影响用户体验。热控策略简单直接，发现温度升高就直接采用限制频率的策略。相同的计算量，频率被限制后导致计算时间被拉长，进一步引发了 CPU 功耗增加，并引发热量增加，热控效果被抵消。例如，游戏场景中，DDR 调频策略使得 DDR 低频点，引发 CPU 负载升高，进而 CPU 频点上升，然而此时性能的瓶颈其实是内存，瓶颈导致的帧率过低的问题没有解决，反而由于 CPU 升频带来了发热问题。

有技术提出将功耗和热的管控进行联合，建立功耗和温升的对应关系模型，根据功耗预测温升。从温升预测出发，定期制定功耗预算，目的是满足智能手机温度低于温度阈值。将该功耗预算作用于热源，以达到降低功耗和控制温升的目的。该技术的本质上是通过功耗控制达到热控的目的，仅考虑了热和功耗的协同。但是这种方法很难保证用户的前台使用体验不会受到影响，依然存在为功耗或热控而降低前台性能的问题。进一步的，在前述功耗和热联合管控的基础上，有技术提出可让用户设置性能阈值，作为功耗和热的管控底线。即便如此，该技术对用户体验的考虑只是允许特定场景下用户手动标定不可接受的性

能底线，并没有把手机的整体使用体验考虑在内，例如待机时长、壳温等，容易导致手机整体使用体验降低。

以上技术提出的性能、功耗和温度的管控策略均没有针对用户使用模式差异针对性设置。不同用户使用手机的模式千差万别。例如，有些用户随时可以充电，待机时长不是体验瓶颈；有些用户长期出差在外，一天才能充一次电，对待机时长重视，不太看重游戏性能。再例如，低纬度地区环境温度高，用户对手机的发热很敏感，而高纬度地区，环境温度低，用户对发热耐受度很高。

因此，本申请提供提出了一种基于性能、功耗和温度（即发热或热）的融合资源调度方法。本申请的出发点是用户的使用场景（相当于终端设备的应用场景），而不是性能、功耗和温度的独立负载(或劣化程度)，体现了不同使用场景下体验的侧重点差异。例如，游戏场景对性能的体验要求高，可以降低待机时长的体验限制。本申请的调度决策将性能、功耗和温度结合，把用户的综合体验最优化作为调度的目标，而不是从一个或两个目标出发产生决策点，避免了管控机制彼此的冲突。

进一步的，本申请提供的方法考虑了场景和用户的差异性，不是千人一面千篇一律地执行调度。具体的，本申请基于性能、功耗和温度三方面建立用户整体体验模型，针对不同的使用场景，该模型的侧重点有所差异；同时性能、功耗和温度三者的负载分别有各自的测量模型，这些测量模型是根据用户或设备的个体特点建立的。以上结合，最终生成基于不同场景的、且具备用户差异的调度策略。

下面将以智能手机为例介绍本申请提供的融合资源调度（或管理或控制）方法。应理解的是，本申请提供的方法不仅可以应用在智能手机上，还可以应在其他类型的终端设备上。这里的终端设备包括但不限于智能手机、车载装置、个人计算机、人工智能设备、平板电脑、个人数字助理、智能穿戴式设备（例如智能手表或手环、智能眼镜）、智能语音设备（例如智能音箱等）、以及网络接入设备（例如网关等）等。另外，本申请提供的方法还可以应用在服务器等其它类型的计算机设备上。

如图 1 所示，为本实施例提供的智能手机 100 的逻辑结构示意图。智能手机 100 包括三层：应用 110、操作系统 120 以及硬件 130。应用 110 包括各种应用，例如游戏、微信、YouTube 视频、浏览器等。硬件 130 包括中央处理器（central processing unit, CPU）、应用处理器（application processor, AP）、图形处理器（graphic processing unit, GPU）、存储器（例如 DDR）、输入设备、显示设备（例如触摸屏是输入与显示合一的设备）、各种类型的传感器等。有的未在图中示出。更多硬件可参考后续实施例的描述。

操作系统 120（operating system，简称 OS）是管理和控制计算机硬件与软件资源的计算机程序，是直接运行在“裸机”上的系统软件，应用 110 内包含的各种应用软件通常都需要在操作系统的支持下才能运行。操作系统 120 是用户和计算机的接口，同时也是硬件 130 和应用 110 的接口。操作系统 120 的功能包括管理硬件 130 内的各种硬件、应用 110 内的各种软件、以及各种数据资源，控制程序的运行，改善人机界面，为应用提供支持，让计算机系统所有资源最大限度地发挥作用。由此可见，资源管理和控制机制是操作系统 120 的核心（本实施例中将实现该机制的软件程序称之为资源管理和控制装置）。本实施例提出的资源调度方法正是运行在操作系统 120 内部的一种方法。通过该方法感知上层的应用场

景，监控底层资源运行状况，并融合性能、功耗、温度三个因素确定资源的调度策略，以提升智能手机 100 的整体体验。

图 2a 和图 2b 分别示出了本实施例提供的资源调度方法的一种模块划分示意图和方法流程的概要示意图。如图 2a 所示，本实施例提供的资源调度方法运行在操作系统 120 内部，主要包括三个模块：场景感知模块 121、资源监控和决策模块 122 以及执行模块 123。这三个模块分别执行图 2b 中示出的三个步骤 S201-S203。

S201、确定手机当前的应用场景。

应用场景通常包括以下一个或多个维度的信息：时间、空间、前台应用的类型以及应用内的使用场景。例如，“在家看视频”这个场景包括空间和前台应用的类型两个维度。

手机的应用场景可以理解为用户使用手机的使用场景，比如开会场景、睡眠场景、工作场景、家场景、噪音场景、开车场景、跑步场景等。

手机的应用场景可以是单维度的，也可以是多维度的，例如用户在噪音环境下看视频。进一步的，若用户正在使用某个应用，可以在该应用内继续细分不同的场景，例如正在观看 1080p 或 720p 的视频，或者正在玩某个游戏中的团队战役等。

需要说明的是，手机的动作或运动状态可以划归到“空间”维度。举例说明，跑步场景的前台应用类型可能为某个健康应用或某个音乐应用，同时手机上的感应装置感应到手机在规律性运动，即“跑步场景”包含前台应用的类型和空间这两个维度的信息。

S202、基于所述应用场景下性能、功耗和温度三者的权重确定资源管控策略。

确定资源管控策略的触发条件可以是性能劣化程度、功耗劣化程度以及温度劣化程度中的任意一个或多个达到预设条件。

性能劣化程度用于指示所述设备的性能参数值与目标性能指标之间的差距，所述目标性能指标为满足所述应用场景下的性能体验的一个或多个性能参数的值。功耗劣化程度用于指示预测的可用电量与最低可用电量之间的差距。温度劣化程度用于指示预测的外壳温度和温度阈值之间的差距。

需要说明的是，本实施例中所说的两项的“差距”指该两项的区别。“差距”的实现方式有多种，例如可以是这两项的差值、这两项的比值，或在差值或比值的基础上再增加其他的因素后获得的值；或者不是具体的值，是一种表现趋势的函数，例如性能参数值的发展与目标性能指标相违背，比如性能将要下降，但目标性能指标为更高值。换句话说，只要能够表征出劣化程度即可，本实施对具体实现方式不做限定。

这里的性能、功耗和温度可以分别理解为性能体验、待机时长（即功耗体验）、外壳温度（即温度体验）。根据以上三者分别的权重确定资源管控策略。该权重和前一步骤确定的手机应用场景相关，不同应用场景下，三者的权重通常不同，例如游戏场景下性能体验的权重高于温度和功耗。

具体的，首先根据前一步骤确定的应用场景确定目标性能指标。然后计算所述目标性能指标与手机当前配置的性能参数值之间的差距以确定性能劣化程度，并根据所述性能劣化程度确定是否生成性能劣化告警。根据所述应用场景、电量使用习惯以及所述终端设备的当前剩余电量确定功耗劣化程度，并根据所述功耗劣化程度确定是否生成功耗劣化告警。根据采集到的热区温度预测所述终端设备的外壳温度，根据所述外壳温度确定温度劣化程

度，并根据所述温度劣化程度确定是否生成温度升高告警。当所述性能劣化告警、功耗劣化告警以及所述温度升高告警中的任意一个或多个被生成时，说明当前的资源管控出不能满足手机整体的用户体验，因此开始资源管控策略的确定。

在其他实施例中，用户可以设置资源管控周期，周期性触发策略确定过程。

S203、根据所述资源管控策略调度所述设备的资源。所述资源管控策略包括以下三项中的任意一项或多项：计算资源调度策略、应用管控策略以及外设管控策略。

在一些实施例中，前一步骤确定的资源管控策略为满足所述应用场景下用户体验最优的资源管控策略。用户体验由性能、待机时长和温度三者的满足程度以及三者分别在所述应用场景下的权重通过一定的数学变换获得。

确定资源管控策略的过程可以理解为求解用户体验最优的数学过程，例如机器学习或动态规划的方法。在其它一些实施例中，“最优”未必理解为理论上或数学上的最大值。用户体验作为目标体验，该值可以由用户设置，或者该值虽不是数学上的最大值，但可能是当前场景下最合适的一个值。

由以上可见，通过本实施例提供的方法，手机可以感知用户所处的使用场景，根据使用场景确定目标资源需求（相当于目标性能指标），并计算性能需求和当前资源配置之间的差距，以获得性能劣化数据，同时结合功耗和热这两方面产生的劣化数据，基于性能、功耗和热这三方面在当前使用场景下的权重产生调度决策并下发，以达到同时保证前台性能、待机时长和手机壳温处于稳态的目的。

与现有技术相比，本申请提供的方法在识别用户使用场景的前提下，将性能、功耗和热三者按照当前场景下的权重结合起来，以三者结合的用户综合体验最优化作为调度的目标，而不是从一个或两个目标出发产生决策点，避免了管控机制彼此的冲突，体现了不同使用场景下三者的侧重点，从而生成更加有效的资源调控策略。

图 3 为本实施例提供的资源管理和控制装置进一步的模块划分示意图，尤其是对于资源监控和决策模块进行了更加详细的描述。如图所示，该装置包括场景感知模块 210（相当于场景感知模块 121）、执行模块 260（相当于执行模块 123）、以及 220-250 四个模块。

场景感知模块 210 用于识别终端的应用场景。该应用场景由一个或多个维度的信息标识。例如用户使用情景（家/办公室、白天/黑夜、工作日/休息日）、应用类型（如视频类、游戏类等）、应用内细粒度使用场景（如在看 1080p 高清视频）等。

性能模型 220 用于预测性能是否会劣化或性能劣化程度。具体的，性能模型 220 用于感知不同应用场景下的目标性能目标，结合当前的系统状态（包括资源供给状况和系统负载情况），预测和判断用户性能维度的体验是否会劣化以及劣化的程度。进一步的，性能模型 220 会根据芯片上报的 PMU (performance monitor unit)信息，分析和识别出造成性能劣化的主要瓶颈。

需要说明的是，性能模型的预测通常是毫秒级别。

功耗模型 230 用于预测功耗劣化程度。在一些实施例中，根据当前应用、手机剩余电量和学习出的用户用电习惯，预测未来的一段时间内（例如 1 小时）是否满足待机时长的约束，若不满足约束则给出功耗劣化的程度。功耗劣化程度可以用待机时长或者电量劣化程度来表示。

需要说明的是，功耗模型的预测通常是小时级别。

热控模型 240 用于预测温度劣化程度。在一些实施例中，根据热稳态算法和壳温预测算法，判断未来的一段时间（例如 8 分钟）是否会出现手机温度超过阈值，引起严重发热的现象，若存在会发出只是。进一步的，热模型会根据手机上的热区的温度和热区对最终外壳温度的影响程度，来识别导致发热的主导热区。

需哟啊说明的是，热控模型的预测通常是分钟级别。

决策模块 250 用于基于当前应用场景下性能、功耗和温度三者的权重确定资源管控策略。在一些实施例中，根据来自于性能模型、功耗模型和热模型的告警信息触发融合调度决策，综合考虑上述三个维度的需求、现状、以及当前应用场景下三个维度的权重来动态决定用户综合体验最佳的最终方案。

执行模块 260 用于根据上述确定的资源管控策略实施资源管控。资源管控通常涉及多种资源的多项管控措施，因此调度管控模块可以理解为多个分布在系统的不同地方的功能单元的组合。资源管控包括对实时资源、后台进程和外设等的调度。

下面分别从场景感知、性能模型、功耗模型、热控模型、融合调度这几个角度详述本申请的多种实现方式。下面的各个方法可以理解为前述相应的功能模块的具体实现。容易理解的是，一个功能模块可以包含多个子模块，功能模块的具体实现可以是这多个子模块协作完成的。

(1)场景感知（对应场景感知模块 210）

场景感知是多种识别方法的综合，输出用户的具体使用场景，如当前用户在家里在看清晰度 1080p 的硬解码视频。场景感知包括用户时空感知、应用类型感知、以及应用内场景识别三个方面。

用户时空感知又可以包括时间维度和空间维度。时间维度例如节假日/工作日、白天/黑夜。通过获取当前的系统时间或从系统的日历中查询以区分时间。空间维度例如出差/旅游、家/办公室。通过采集得到的 GPS 位置信息进行聚类，按照不同时间的最频繁出现的地理位置点识别家和办公室位置；结合当前的时间来确定是否出差或旅游等。

应用类型感知即感知用户在前台使用的是什么应用。可以在手机系统中预置多种应用类型的列表，例如国内使用人数排名前 10000 名的应用类型，然后根据当前的前台应用查询列表以确定该前台应用对应的应用类型，比如微信（WeChat）对应即时通讯类应用。在另一些实现方式中，手机也可以将当前应用的标识通过网络发送给云端服务器，由云端服务器识别并返回识别结果给手机。

应用内场景识别即识别应用内更细粒度的场景。应用内场景维度包括用户对应用的操作行为（简称用户行为）和应用内的进程的行为（简称进程行为）。用户的行为例如播放不同清晰度的视频、进行视频聊天等属于用户的行为，游戏的启动、放大招、选英雄等属于应用内进程的行为。用户行为可通过打点上报各种事件以及在数据流中执行元数据识别等方式获取。进程行为可以通过应用的绘制信息或预先嵌入在应用中的 SDK 主动向系统上报等方式识别。

(2) 性能模型 （对应性能模型 220）

性能模型主要用于识别不同场景下的性能目标，并结合当前的系统状态（包括资源供给状况和系统负载情况），预测和判断用户性能维度的体验是否会劣化（毫秒级别），以及劣化的程度。进一步的，性能模型会根据当前使用场景和系统状态，分析和识别出造成性能劣化的主要瓶颈。

图 3 为场景识别以及确定性能劣化告警的方法流程示意图。该方法大致分为以下 a)、b)和 c）三部分。

a）识别场景和对应的目标性能指标

通过各种方式实现场景识别的过程，前述（1）中已经介绍了，在此不再赘述，可参考图 4。识别场景（S310）之后，在预先设置的数据库 301 中搜索与所述场景匹配的目标性能指标（S320）。数据库 301 存储有多种应用场景以及多种目标性能指标的对应关系。该数据库可以是离线生成并预先配置的。

具体的，该数据库中包括不同场景(记做 Si, i=0…n)，以及不同场景对应的目标性能指标 Ti。Ti 包括：目标帧率、丢帧率、低帧率占比（低于目标帧率的帧数占总帧数的百分比）、以及流畅度（帧率波动情况）等。

在一些实施例中，数据库中的场景划分和体验目标可以是人工参与制定的。在另一些实施例中，数据库中的场景划分和目标性能指标是从通过对上报的大数据进行聚类发现的，例如，视频播放场景细分为不同的帧率（24/25/30 帧）和软硬解码场景，分别对应不同的体验目标。

在系统运行时，场景感知模块 210 可以用于识别场景 S_i，然后性能模型 220 查询数据库 301 以获得该场景 S_i 的目标性能指标 T_i。例如，场景感知模块 210 通过视频流数据的解码方式和帧头部信息确定是硬解码 30 帧场景，该场景输入性能模型 220，性能模型 220 确定对应的体验目标可能为 Ti = {平均 30 帧，丢帧率低于 30%，低帧率占比不高于 20%，流畅度在 0.8 以内}。

b）确定系统当前状态下的性能指标

根据当前的应用场景和系统状态，预测下个采样周期（毫秒或秒级）的帧率变化并通过计算获得其他维度的性能指标(S330)。其中，系统状态包括资源供给状况和系统负载情况。其他维度的性能指标，例如丢帧率、低帧率占比以及流畅度，可基于当前帧率变化和相应的计算公式计算获得。计算公式如下：

$$\mathcal{F}^{i+1} = \mathcal{F}^{i} + \Delta\mathcal{F} \qquad (1)$$

其中，$\mathcal{F}^{i+1}$是下个采样周期的帧率，$\mathcal{F}^{i}$是当前采样周期采集到的帧率，$\Delta\mathcal{F}$是预测的帧率变化，其具体计算方法为：

$$\Delta\mathcal{F} = \beta_{overload} \times \sum_{r\in R} \frac{r_{cur} - r_{within\_scene}}{|r_{cur}|} \times \alpha_{app} \qquad (2)$$

其中，r 表示资源的某个维度，R 表示所有资源的集合，r_{cur}表示当前资源供给状况，

r_{within_scene}表示在这个应用内场景下资源的需求情况，$\frac{r_{cur}-r_{within_scene}}{|r_{cur}|}$表示进行归一化操作，防止维度之间的值域相差过大引起的维度消失情况，α_{app}表示不同应用在不同资源上的权重系数，$\beta_{overload}$表示系统负载对应的权重系数。

其中α_{app}和$\beta_{overload}$可以通过如下方式获得：第一种，离线分析后预置在手机中；第二种，在线分析，例如统计应用在运行时的各个资源的使用情况，分析这个应用对不同资源的侧重/需求，基于此来学习各个维度的权重。

除了上述预测的方式可以获得系统当前状态下的性能指标之外，还可以通过数据采集的方式获得。参考图 3，采集一段时间的帧率，并获得，并通过采集或计算的方式获得平均帧率、丢帧率、低帧率占比、流畅度等其他维度的性能指标（S340）。

需要说明的是，预测和采集的数据可以择其一使用，也可以对这两种方式获得的数据做一些处理后使用，例如取二者的平均值。

c)用户性能体验差距的评估

根据目标性能指标和当前实际反馈的性能指标，可以计算当前性能距离目标性能的差距，即性能劣化程度（S350）。这里，实际反馈的性能指标可以通过上面提到的预测方式所获取的，也可以通过实时采集的方式所获取的。

性能劣化程度的计算公式如下：

$$\Delta T = \sum \alpha_i * \frac{T_{target\,i} - T_{out\,i}}{T_{target\,i}} \quad (3)$$

其中，α_i为第 i 项指标在不同场景下的劣化权重。$\sum \alpha_i = 1$。例如，游戏场景中平均帧率劣化对应的权重就比视频场景下平均帧率的权重大。$T_{target\,i}$表示第 i 项指标的目标值，如游戏中对战场景的平均帧率的目标性能指标是 57 帧。$T_{out\,i}$表示第 i 项指标的实际值。

当ΔT超过阈值时（如超出一定的百分比），视为性能发生劣化，需要将性能劣化告警事件上报给决策模块 250。

需要说明的是，不同的性能指标有不同的要求，有的量化后数值越高越好，有的量化后的数值越低越好，所以公式$\frac{T_{target\,i} - T_{out\,i}}{T_{target\,i}}$是可以根据具体情况变化的，此属于常见的数学方法，本实施例在此不一一列举，具体示例可参考后续描述。

需要说明的是，以上步骤除场景识别由场景感知模块 210 执行之外，其他的步骤均可由性能模型 220 执行。

(3)功耗模型 (对应功耗模型 230)

功耗模型用于学习用户的电量使用习惯，并根据当前应用场景和当前的应用对待机时长或者可用电量的影响程度进行评估。对于确定的用户，待机时长和可用电量两个变量之间可相互转换，故下文将仅以可用电量作为例子来说明。功耗模型的具体实现大致包括以下 a)、 b) 和 c)三个部分。

a)采集和记录用户的电量使用信息

本步骤会在学习用户使用习惯之前执行，采集和记录的信息包括但不限于用户使用的应用、使用时长、充电开始时刻和充电结束时刻、充电地点、应用平均功耗等信息，并将其进行持久化存储，其存储形式为文件或数据库。

b)分析和学习用户的电量使用规律

本步骤以步骤 a)中持久化保存的电量使用信息作为输入，通过机器学习或者统计学的方法来学习用户的电量使用规律，其中电量使用规律包括充电周期、用电规律和应用使用规律。

具体的，以充电开始时刻和充电结束时刻作为输入，学习出充电周期。不同用户充电周期可能不同，例如有的人一天一充，有的人两天一充。统计用户在每个时间点的剩余电量，学习出基于时间点的历史剩余电量，作为用电规律。以用户使用的应用、使用时长、应用平均功耗、用户位置等作为输入，学习出在不同位置和时间段的用户常用应用，以及应用的平均使用时长和平均使用功耗等信息作为应用使用规律。

c)基于使用习惯的功耗划分

以步骤 b)中学习出的充电周期和用电规律作为输入，通过时间段划分和统计学方法，输出每个时间段及这个时间段对应的最低可用电量。

时间段以充电周期作为划分基础，而不是固定的一天。不同的用户充电周期有较大差异，例如有的人一天一充，即充电周期时长为 24 小时；有的人两天一充，即充电周期时长为 48 小时。

在一些实现方式中，时间段的划分方法可以是固定时间片，例如每 3 小时作为一个时间段。在另一些实现方式中，时间段的划分可以基于语义，例如早晨、上午、中午、下午、傍晚、晚上、半夜。在另一些实现方式中，时间段可以使用动态划分方法。

时间段的动态划分方法举例如下：先使用预置的时间段的下限个数，如 3 个，对充电周期进行等距划分。确定每个时间段的持续时长，若时长超过阈值，例如 3 小时，则在这个时间段进行插值，保证每个时间段的持续时长不超过时长阈值。评估每个时间段的电量变化或待机时长变化，若一个时间段的电量变化或待机时长变化超过阈值，例如上一个时间段最低可用电量为 80%，后一个时间段最低可用电量为 20%，电量变化为 60%，超过阈值 20%，则在这个时间段进行插值，保证每个时间段的电量变化不超过电量阈值。

图 5 为划分好的时间段的示意图，其中每个时间段的最低可用电量计算方法如下：

$$E_i^{th} = \frac{1}{N}\sum_N E_{ji}^{history} + \gamma(\text{std}, \text{loction}, \text{scene}) \qquad (4)$$

其中，E_i^{th}表示第 i 个时间段的最低可用电量，N 表示过去发生的次数，$E_{ji}^{history}$表示第 j 个历史上该时间段的剩余电量，$\gamma(\cdot)$表示弹性函数，即允许最低可用电量可以在$\gamma(\text{std}, \text{location}, \text{scene})$所代表的值区间内进行波动，它以历史统计的标准差 std、当前位置 location 和当应用场景 scene 作为变量。

d)根据当前应用场景的实时功耗劣化评估。

本步骤在步骤 c)完成后执行。本步骤是周期性执行或者用户使用的应用发生变化时被触发执行。本步骤的输入是当前电量、步骤 b)中学习出的应用使用规律和步骤 c)计算得到的时间段划分信息，输出是功耗劣化评估和劣化程度。

具体的，根据应用使用规律和当前的前台应用，预测出该前台应用的使用时长以及下一个时间段所预估计的电量消耗ΔE，然后根据下一个时间段需要满足的最低可用电量 E_i^{th}，评估是否出现功耗劣化。如果当前剩余电量E_{cur}减去预估计的电量消耗ΔE低于下一个时间段需要满足的最低可用电量 E_i^{th}，则评估会出现功耗劣化，并将告警发送给融合调度模块进行决策。劣化程度ξ的评估可通过下面的公示计算所得：

$$\xi = \frac{E_i^{th} - (E_{cur} - \Delta E)}{E_i^{th}} \qquad (5)$$

(4)热控模型 (对应热控模型 240)

根据热稳态算法和壳温预测算法，判断将来的几分钟内是否会出现温度升高预警（手机外壳温度超过阈值）。同时，热控模型会根据热区温度和热区对最终温度的影响程度，来识别导致发热的主导热区。参考图 6，热控模型的实现主要分为以下 a)-e)几个部分。

a)散热能力识别

手机的布局，包括手机的结构布局和手机的外壳材质，会影响手机本身的散热能力。借助函数逼近方法，例如最小二乘方法，离线学习出多种使用场景下的该手机的散热能力曲线，并在设备出厂时提前进行预配置。

b)壳温预测

壳温就是手机的外壳温度。所述外壳温度是根据手机当前的外壳温度、手机内多个器件各自的温度升高预测值、以及每个器件温度的权重确定的。器件的温度升高预测值根据该器件的温升函数和该器件的当前资源配置确定，其中，所述温升函数可以预置在手机中。

具体的，根据采集的当前的热区温度，以及热区本身对壳温的影响程度来预测未来短期内 t 时刻（分钟级预测）的手机壳温，具体计算公式为：

$$H(t) = T_{now} + \sum \alpha_i * F_i(R_i, t) \qquad (6)$$

其中α_i是各热区温度的权重，与硬件设备的布局等相关。T_{now}是当前时刻对热区采样的温度值。各热区在未来 T 时刻的温升温度可用$F_i(R_i, t)$来度量。其中R_i是当前的系统资源配置（或供给）情况，t 是持续时间。F_i是第 i 个热区器件（如 CPU 或 GPU 或 DDR）的温升函数，其与产品的硬件配置和布局有关，一般是多个因素线性拟合的结果，为固定函数。

DDR 即双倍速率，DDR SDRAM 即双倍速率同步动态随机存储器，本领域习惯称为 DDR，其中，SDRAM 是 synchronous dynamic random access memory 的缩写，即同步动态随机存取存储器。

需要说明的是，一个“热区”可以仅包含一个器件，也可以包含多个相同类型或不同类型的器件。

c)主导热区识别

计算每个热区对壳温上升所贡献的比例并进行排序，选取排序在前的N个热区作为主导热区，贡献比例的计算方法为：

$$\rho_j = \frac{\alpha_j * F_j(R_j,t)}{\sum \alpha_i * F_i(R_i,t)} \quad (7)$$

其中ρ_j是热区 j 对壳温所贡献的比例，$\alpha_j * F_j(R_j,t)$是热区 j 产生的壳温上升量，$\sum \alpha_i * F_i(R_i,t)$是所有热区产生的壳温上升量。

识别出主导热区，可以为决策模块250的调度决策提供输入。

d) 环境温度识别

环境温度（简称环温）计算的输入包含：NTC节点采集的温度数据、各器件的功耗数据以及温箱环境温度数据进行模拟训练，计算得到环境温度模型的拟合参数。流程如图7A所示。首先将当前系统状态及各个NTC节点采集的存入状态历史表(S701)，然后判断是否需要更新环境温度(S702和S703).如果与上次环温输出的间隔时间没有超过间隔阈值(S702)，则不更新，反之则继续判断是否处于充电状态(S703)，如果处于充电状态则不更新。如果与上次环温输出的间隔时间已经超过间隔阈值且不处于充电状态，那么对状态历史表中的温度数据进行过滤（1），然后根据过滤之后的数据计算最新的环境温度（2），计算结果可能需要一些过滤处理（3），过滤处理后将最终的结果加入环温历史表(S704)。

NTC全称是thermistor temperature sensor。NTC节点就是指温度采集传感器，用于采集各个热区的温度。

环境温度的计算有一定的周期，当距离上次计算时间超过阈值，并且当前手机不处于充电状态时，执行最新环境温度计算，并将计算结果加入环境温度历史表。

如图7B所示，环境温度计算采用类Kalman滤波的计算方法，具体的，可以采用线性时不变降阶模型（linear and time-invariant reduced-order model, LTI-ROM）来预测环境温度，该模型的输入是多个功耗P1-P5以及采集的温度值。LTI-ROM模型输出的是下一个时刻的环境温度预测值，该预测值被输入校正模型，由校正模型结合环境温度的实际值对预测值进行校正，预测值和实际值一致的时候输出环境温度，不一致时反向调节LTI-ROM模型的参数。

e)温升告警决策

热控模型中，温升告警的判断条件可以包括以下两个：短期内的最大温度不能超过阈值，同时这段时间内温度波动范围不能超过阈值。

用公式描述：

$$\forall t \in [0,T)\ H(T) \triangleq \begin{cases} Max(H(T)) < T_{thresold} \\ Max(H(T)) - Min(H(T)) < \varepsilon \end{cases} \quad (8)$$

其中时间段 T 的整机温度用 H(T)描述，$T_{thresold}$是温度上限阈值，ε是温度波动阈值。上述公式的意义是时间段 T 内的最高温度小于$T_{thresold}$，同时最高温度和最低温度的差值小于ε。在其他实施例中，“小于”也可以是“小于或等于”。

温度阈值$T_{thresold}$和/或ε是可以根据环境温度调整的。例如环境温度很低，那么$T_{thresold}$可以设置的高一些，因为此时人对手机外壳温度的容忍程度比较高；相反，环境温度很高，那么$T_{thresold}$可以设置的低一些，因为此时人对手机外壳温度的容忍程度比较低。

在其他实施例中，上述两个条件并非缺一不可，可以选择其中任意一个作为温升告警的判断条件。

5)融合调度模型

当有性能劣化告警、功耗劣化告警或者温升告警中的一个或者多个事件出现时，会触发融合调度决策，并根据融合调度模型决定用户整体体验最佳的最终方案，对实时资源、后台应用和外设等进行管控和调度。

在当前状态X＝｛手机使用场景 * 性能劣化程度 * 功耗劣化程度(相当于待机时长劣化程度)* 温度劣化程度｝的前提下，在可用的管控策略空间A中搜索最优的策略a，使得短期内用户的整体体验R最优。其中策略a的组成可以为{ CPU/GPU/DDR 档位, 外设管控档位, 应用管控策略}。用户的整体体验R是以下因素的加权值：

$$R = \sum \alpha_i * S_i \quad (9)$$

其中S包括以下维度：性能体验目标满足程度（S_1）、待机时长(S_2)以及壳温体验(S_3)。换句话说，整体体验R与性能、功耗以及外壳温度三个因素相关。

α_i是每个因素的权重，该权重在不同的终端应用场景下（或不同的用户）有不同。例如，游戏场景中性能体验目标满足程度的权重就要高于阅读电子书场景。搜索的目标就是寻找最大化R的策略a，即$max_{a \in A}R$。

在具体实施中，可将融合调度决策的选择过程抽象化为一个强化学习或马尔科夫决策过程。以强化学习为例，

- 离线状态下，根据走访、用户体验，确定各种场景下体验目标权重α的选择；
- 将融合调度决策选择过程抽象为四元组E＝<X, A, P, R>。其中环境X表示当前的状态，即{使用场景、性能/功耗/热劣化程度} 。A表示当前可选择的策略，如资源供给调整、外设管控策略、应用管控策略构成的策略组合。P是离线环境下学习得到的状态转移概率矩阵。R是在X状态下作出A策略的奖赏，即用户整体体验的优化/劣化变动；
- 通过离线场景学习的P和R，在运行时根据状态X，采取相应的策略A。

同时，融合调度决策会根据运行时的系统和用户的反馈来学习和更新后续的策略选择。涉及的反馈包括：

- 性能维度:丢帧、卡顿、外设主动调节反馈（调节背光、音量）、后续使用的应用等；
- 功耗维度：电池耗尽的次数、某个时间段功耗是否超标等；
- 温度维度：电池发热次数、电池温度是否有效下降等。

下面结合游戏场景的一个实例以及前述方法本申请提供的融合调度方法。

如图8所示，该方法主要分为场景感知、性能模型、功耗模型、热控模型四个感知模块，融合调度中心（含决策库）一个决策模块和一个对接外部调度机制的执行模块。以下是各个流程的具体描述。场景感知负责识别细粒度的使用场景。性能模型是毫秒级的感知模块，负责根据使用场景确定目标性能需求，以及确定当前性能指标，然后根据两者确定性能劣化程度和性能瓶颈。功耗模型是小时级别的感知模块，负责判断功耗（即待机时长）劣化程度并适时输出预警。热控模型是分钟级别的感知模块，用于判断温度升高程度以及主导热区。融合调度中心接收到性能模型、功耗模型、热控模型这三个感知模块的输入，然后做出整体性能最优的调度策略，并下发到执行模块。调度策略可以包括计算资源调度、后台应用管控以及外设管控等。决策库用于存储一些预置的调度策略，以供融合调度中心在其中搜索以获得整体性能最优的调度策略。

调度策略的执行过程并非本申请的重点，所以除执行模块之外，下面对剩余的五个流程分别进行详细的描述。

流程一. 用户使用场景感知

对用户的使用场景（即手机的应用场景）进行感知，作为性能模型的基础输入，用于识别用户前台的性能需求。

如图9所示，用户的使用场景包括几个维度，分别由以下数据来源提供。

- 用户的时间维度和空间维度，包括当前时间、当前用户所在位置类型。时间维度可通过系统日历进行查询，例如节假日和工作日。空间维度包括家/办公室/其他，可通过通过位置聚合服务获得，例如某些手机提供了情景智能服务，则可以通过查询情景智能服务接口获取。
- 应用类型维度，如即时通讯类应用、邮件类应用或游戏类应用。应用类型一般表示应用的核心用途。例如微信（WeChat）的类型就是即时通讯类应用。数据的来源可以是查询应用类型数据库（数据库设置在端侧或云侧），或在端侧动态识别，亦或从云端查询。
- 应用内场景维度，如微信内可以在朋友圈里看视频，则应用内场景就是视频播放场景。场景可根据框架（Framework）中打点上报的应用行为事件和前台应用关联信息进行识别。例如，根据打点上报的全屏事件以及视频播放事件（或视频类型应用和发声事件）可以识别视频播放场景。对于非游戏，还可通过应用活动（Activity）和场景关联，通过前台的Activity识别不同的应用内场景。例如，ActivityManagerService

上报com.tencent.mm/com.tencent.mm.plugin.sns.ui.SnsTimeLineUI这个Activity在前台，则可知用户当前正在使用微信应用的朋友圈功能。

- 应用内场景参数维度，如优酷（Youku）内看视频区分全屏/非全屏、软解视频/硬解视频、360p、720p/1080p等不同的清晰度、是否有用户字幕显示等。这部分数据可通过在框架中打点获取，或集成了特定sdk的应用主动发送相关的参数。

用户的使用场景感知结果，封装成scene数据类型表示的细粒度使用场景，传递给性能模型、功耗模型，作为这两种模型的输入。

以游戏为例，假设用户当前在玩G游戏，游戏的帧率有30、40、60帧的不同场景，则用户场景感知输出的scene信息为：<节假日，晚上，家里，G游戏，团战，60帧>。其中“G”指游戏应用的名称，“团战”指的是该游戏中的一种多人战争场景。

流程二. 性能模型

性能模型用于发现性能体验的劣化程度，发出性能劣化感知信息和性能瓶颈，通知融合调度中心。对于游戏场景，性能体验指标通常包括以下维度：平均帧率、低帧率占比以及抖动率信息。其他场景下，性能体验指标可能与游戏场景相同，也可能不同，本实施例对此不做限定。

在性能模型中按照以下顺序执行1-4的顺序执行，最后输出性能劣化程度和性能瓶颈。

1. 识别场景体验目标

根据Scene=<节假日，晚上，家里，G游戏，团战，60帧>，在数据库中进行查询，获得当前场景对应的性能指标。

具体的，性能指标可以根据手机的分级进行区分。例如旗舰机、中端机、低端机分别对应性能目标A、B和C。三种手机级别在数据库中可以实现为三张表，每张表包含该手机级别的场景和性能指标的对应关系；也可以是一张表，由手机级别字段进行区分。

延续前述例子，从框架层上报信息得知运行环境为旗舰机，分辨率为1080p，则确定当前Scene对应的性能目标A就是<平均帧率57，低帧率占比0.5%，抖动率0.2%>。

如果是中端机，则对应的性能目标B为<平均帧率56，低帧率占比1%，抖动率3%>。如果是低端机，则对应的性能目标C为<平均帧率55，低帧率占比1%，抖动率1%>

2. 采集前台应用性能

通过在Surfaceflinger中进行打点，采集应用当前的实际性能指标。除了当前的实际性能指标，性能模型根据系统当前的负载和SOC资源供给状况，可进行短期内性能指标变化预测。Surfaceflinger是安卓系统框架层的一个组件。SOC全称是system on a chip，可认为是芯片组，集成处理器的各个核，本实施例中包括4个大核和4个小核。在本实施中，GPU、DDR也在SOC上。

根据采集或预测得到的实际性能指标与性能目标的差距，可以发现性能劣化。对这种性能劣化进行量化，产生不同级别的性能劣化告警。

3. 性能瓶颈发现

性能瓶颈发现主要依靠 SOC 上的性能监控单元（performance monitor unit，PMU）上报事件进行识别，为已有成熟方案，此处不进行赘述。

4. 量化场景劣化

性能的劣化量化方式为对三项体验指标的加权，其量化方式可参见前述公式(3)。

延续前述例子，在 G 游戏团战场景中，三种指标的劣化程度权重为：平均帧率 0.5、低帧率 0.3、抖动率 0.2，对应公式(3)中的α_i。换句话说，平均帧率在 G 游戏团战场景中的权重最大；低帧率次之；抖动率的权重最小。

每个维度的劣化程度的量化定义为当前性能指标$T_{out\,i}$距离性能目标$T_{target\,i}$的百分比差距。例如性能目标 A 中的平均帧率目标为 57 帧，当前平均帧率为 50 帧，则平均帧率劣化值为(57-50)/57=0.12。

例如，低帧率劣化值为（0.3-0.5）/0.5=-0.4。其中，目标值为 0.5，当前值为 0.3，当前值优于目标值，所以劣化值是负数，此时可以强制设置劣化值为 0。类似的方法，抖动率劣化值为（0.3-0.2）/0.2=0.5，其中目标值是 0.2，当前值是 0.3，劣化值是 0.5。

根据公式(3)$\Delta T = 0.5 * 0.12 + 0.3 * 0 + 0.2 * 0.5 = 0.16$。不同的$\Delta T$值对应不同级别的性能劣化告警，将性能劣化告警上报到融合调度中心。具体的，可以直接上报 0.16，也可以将0.16转化成性能劣化告警的级别,例如对应的级别为0.3,将0.3上报到融合调度中心。

流程二. 性能模型

功耗模型在本实施例也需要考虑当前的使用场景，因为不同使用场景下的功耗可能是不一样的，比如本实施中的游戏场景，用电量的消耗就要快于聊天场景。

功耗模型的运作过程主要包括以下 1 和 2 两步。

1.学习用户的充电习惯和用电习惯，判断剩余待机时间。

功耗模型在后台收集电量使用信息,电量使用信息分为两部分:充电习惯和用电习惯。

a)充电习惯

统计当前时间和空间类型（工作日/休息日，家/办公室/其他，来自于场景识别模块）中，连续两次从80%以下充电到100%的时间之差，作为用户的充电记录。

对过去 n 次同类时空类型的充电记录进行统计学习，例如取 n=7 即最近 7 次充电记录作为窗口，计算窗口内充电开始时间段的方差。例如，过去 7 次充电时间分别为 9 点、10 点、9 点、9 点 30、8 点 30、9 点、10 点半，则对应的方差为 0.41。

如果充电记录的方差小于一定阈值，则认为用户的充电习惯稳定，可得到用户在当前时空类型下距离下一次充电的时间预测。如果充电记录方差比较大，则使用默认充电时长

（12 小时）作为预测，计算距离下一次充电的时间间隔。

b)用电习惯

将一天划分为 k 个时间片，如取 k=8，每个时间片为 3 小时。统计在当前时间片内用户常用的应用、应用单位时间内耗电量，并加权得到该时间片内用户功耗开销，作为用户的用电习惯。

需要说明的是，以上充电习惯和用电习惯仅是举例说明，在本申请的基础上，还可以根据需求获取其他的信息作为功耗模型的输入。

2.功耗劣化量化

周期性地或在前台应用发生切换时，进行功耗劣化评估。功耗劣化评估的方式是计算当前应用对本时间片结束时待机时长的影响程度。

例如，在 15:00，用户开始玩 G 游戏，该用户的充电习惯显示用户可能在 21:00 开始充电，当前电量剩余 3000mA。功耗模型需要预留 20%的电量避免进入省电模式，因此当前可分配电量为 2800mA。15:00 – 18:00、18:00-21:00 两个时间片中，用户历史用电比例为 3:7，则两个时间片分别分配 840mA 和 1960mA。当用户 15:50 结束游戏后，发现游戏消耗了 300mA 电量，按照用电习惯预估当前时间片内还要消耗 800mA，则当前时间片功耗劣化程度为 1960-(2800-300-800)/1960 = 0.1326。功耗劣化触发功耗报警，将劣化报警发送给融合决策模块。

流程四. 热控模型

出厂前，手机中会预置两部分只与器件型号、整机设备布局有关的信息：

a. 热区器件的温升函数$F_i(R_i, t)$，表示档位在 R_i、持续时间 t 之后该器件产生的温升；
b. 整机散热能力曲线。

在手机使用过程中，在不充电状态时通过卡尔曼滤波的方式对环境温度进行估计，具体方法可参考前述图 7A 和图 7B。

在手机使用过程中，定时对热区温度进行采集，并根据采集的热区温度和热区的温升预测函数计算壳温的预测值。然后根据壳温的预测和环境温度产生温升预警。

例如，当前环境温度为 26 度。用户在玩 G 游戏的过程中，CPU 高负荷高频点运行。通过各热区的温度采集和热区温度预测，发现当前壳温 38 度，将在 5 分钟内超过 42 度，短期内的最高温度（42 度）超过了阈值（假设阈值是 40 度），则产生温升预警。

报送到融合调度中心的是量化后的温度劣化结果，将类似<短期温升，当前 38 度，5 分钟内超过 42 度>的预警信息转化为一个温度劣化值。这种转换可以是按公式进行，也可以根据规则进行。例如，将温度分为几个档位：38,42,45,48,53，在 5 分钟内温升超过对应档位分别对应温度劣化值 0.2,**0.3**,0.5,0.8,1.2，十分钟内温升超过对应档位对应温度劣化值 0.1,0.2,0.3,0.4,0.6。本实施例中温度劣化值输出为 0.3。

主导热区的发现是在壳温计算的过程中进行的，将对壳温升高贡献最大的热区作为主

导热区（例如 GPU），发送到融合调度中心，供决策使用。

流程五. 融合调度

当有性能劣化告警、功耗劣化告警或者温升告警中的一个或者多个事件出现时，会触发融合调度决策，即寻找最大化用户体验 R 的调度策略，请参考前述公式（9）。具体的α_i和S_i通过如下方式获得。

- 性能、功耗、热在当前应用场景下的权重α_i

每个用户体验维度的权重系数α针对不同的应用场景、不同的用户均有所差异，可以由以下多个因素综合决定：

$$\alpha = \gamma(\text{scene}, \text{time\_domain}, \text{deterioration})\quad(10)$$

其中，γ(.)是权重计算公式，根据多个因素计算出三个维度各自的权重系数，多个因素包括：

a) scene是终端应用场景，例如游戏场景、视频场景、阅读场景等。不同场景下各个维度重要性有差异，例如游戏场景下，性能维度的权重要高于功耗和热维度的权重；本实施例中是：<节假日，晚上，家里，G 游戏，团战，60 帧>；
b) time_domain是该体验维度的作用时域，例如是秒级、分钟级别、小时级别，体验维度的作用时域越小则相对的权重会越大；
c) deterioration 是该维度的劣化程度，劣化程度越严重则该维度的权重越大。

举例而言，在<游戏场景，在家>的条件下，当<性能劣化，功耗劣化，温升劣化>为<0.3,0.3095,0.3>的时候，最终计算的<性能维度权重，功耗维度的权重，温升维度的权重>为<0.5,0.3,0.2>。α的计算公式γ可以是预置的待查询数据，也可以是根据公式实时计算的数据。

- 性能、功耗、热的体验（S_i）的量化

性能维度的用户体验：根据前台用户体验和后台用户体验来衡量。其中，前台用户体验由目标帧率、丢帧率、低帧率占比和流畅度来决定；后台用户体验由后台应用在后续使用的热启动比例来衡量。性能维度的用户体验的量化方式为平均帧率、低帧率占比、帧率抖动三个指标的满足程度，计算方式为 1 减去性能劣化程度 1-ΔT，即性能模型中的性能劣化越小则用户体验越好，最大为 1。

功耗维度的用户体验：根据待机时长来衡量。当待机时长未达到设置的上限时，待机时长越大则功耗维度的用户体验越佳；达到上限后，待机时长的增加不影响功耗维度的用户体验。例如，待机时长的上限可以为用户充电周期的 1.2 倍。功耗维度的用户体验量化方式为预测的待机时长和待机时长的上限的比值，最大为 1。

温升维度的用户体验：当后续温度不会超过阈值时（即用户感知手机不发烫），温升维

度的用户体验根据温升速度的最大值和方差来衡量，最大值越小且方差越小则温升体验最佳（控制温升速度以最平缓的速度上升）。热维度的用户体验量化方式为 1 减去温升劣化程度。最大为 1。温升劣化程度的量化方式见“流程四.热控模型”。

- 最优调度策略搜索

图 10 为一种最优调度策略的确定过程的示例。如图所示，在可用的管控策略空间中寻找最优的策略的方法可以分为以下几步。

S901、借助性能瓶颈、主导热区来确定策略搜索的方向，针对性地选择出候选的资源调度策略集合。

性能瓶颈和主导热区分别由性能模型和热控模型所提供，能帮助减少策略的搜索空间，减少候选的资源调度策略的数目。例如在 G 游戏场景，假设 G 游戏的性能瓶颈是 DDR，当发生性能劣化时应提升 DDR 资源来提升性能体验；假设发现主导热区是 GPU，可针对性地降低 GPU 的频点来实现快速地控制温升。更具体地分析，当多个维度之间存在冲突时，例如主导热区和性能瓶颈都是大核资源，那么大核资源频点上升和下降都是可能的搜索方向，由后面的用户整体体验来决定。

除了性能瓶颈和主导热区之外，资源功耗表信息也可以作为筛选资源调度策略的输入。资源功耗表信息由功耗模型提供。资源功耗表中包括多种资源及其对应的耗电信息，例如，计算资源，如 CPU 和 GPU 等在什么档位，单位时间耗多少电；外设，如屏幕、声音等在什么档位，单位时间耗多少电；不同应用在后台单位时间平均耗多少电。

S902、针对每种候选的资源调度策略，结合性能、功耗、温度这三个维度的用户体验评估方法，来预估这三个维度分别的用户体验。各维度的用户体验评估方法详见前一部分“性能、功耗、热的体验（S_i）的量化”。

S903、根据各个维度在当前应用场景下的权重系数，计算每种资源调度策略的整体用户体验，并进行排序选择得分最高的策略，作为最优的资源调度策略。计算公式为 $R = \sum \alpha_i * S_i$（相当于公式（9）），其中权重系数的衡量方法详见前一部分“性能、功耗、热在当前应用场景下的权重α_i”。

S904、下发并执行最优的资源调度策略，并通过实时反馈（S905）来获取采取该策略后的各个体验维度（包括性能、功耗、温度）的真实变化情况，并借此来调整上述步骤中的计算参数。例如，假设根据之前的计算最优策略实施后性能维度不劣化，但策略执行后实时监控发现还存在丢帧导致性能体验有所下降，则反馈机制会将信息反馈到性能维度的用户体验评估计算处，更新各个参数的权重。进一步的，也会将信息反馈到性能模型的性能瓶颈分析处，重新学习该场景下的性能瓶颈。

需要说明的是，在以上实施例中说明了一些数据量化或数据处理方式，都是为了后续的数学模型能够顺利计算，但是本申请并不局限于这些实施例。在具体实现过程中，本领域技术人员容易根据本申请提出的方法想到其他的数学处理方法。另外，具体计算过程中可能涉及到误差。

图 11 为一种智能手机的设备结构示意图。图 1 为智能手机系统架构的分层示意图，而图 11 尽可能地从硬件角度描绘手机的结构。图 1 中的硬件 130 包括的硬件及其连接关系可以参考图 11。 图 1 中操作系统 120 和应用 110，以软件程序的形式存储在图 11 所示的存储器 870 内部。

如图 11 所示，该手机包括通信模块 810、传感器 820、用户输入模块 830、输出模块 840、处理器 850、音视频输入模块 860、存储器 870 以及电源 880。进一步的，本实施例提供的手机还可以包括神经网络处理单元（neural-network processing unit, NPU）890。

通信模块 810 可以包括至少一个能使该计算机系统与通信系统或其他计算机系统之间进行通信的模块。例如，通信模块 810 可以包括有线网络接口，广播接收模块、移动通信模块、无线因特网模块、局域通信模块和位置 (或定位）信息模块等其中的一个或多个。这多种模块均在现有技术中有多种实现，本申请不一一描述。

传感器 820 可以感测系统的当前状态，诸如打开/闭合状态、位置、与用户是否有接触、方向、和加速/减速，并且传感器 820 可以生成用于控制系统的操作的感测信号。

用户输入模块 830，用于接收输入的数字信息、字符信息或接触式触摸操作/非接触式手势，以及接收与系统的用户设置以及功能控制有关的信号输入等。用户输入模块 830 包括触控面板和/或其他输入设备。

输出模块 840 包括显示面板，用于显示由用户输入的信息、提供给用户的信息或系统的各种菜单界面等。可选的，可以采用液晶显示器（liquid crystal display, LCD）或有机发光二极管（organic light-emitting diode, OLED）等形式来配置显示面板。在其他一些实施例中，触控面板可覆盖显示面板上，形成触摸显示屏。另外，输出模块 840 还可以包括音频输出模块、告警器以及触觉模块等。

音视频输入模块 860，用于输入音频信号或视频信号。音视频输入模块 860 可以包括摄像头和麦克风。

电源 880 可以在处理器 850 的控制下接收外部电力和内部电力， 并且提供系统的各个组件的操作所需的电力。

虽然图示中将处理器 850 描绘成一个组件的样子，但 850 可以表示多个处理器，例如，处理器 850 可以包括一个中央处理器 (central processing unit, CPU) 和一个图形处理器 (graphic processing unit, GPU)。CPU 一般具有多个核，属于多核处理器。这多个核可以集成在同一块芯片上，也可以各自为独立的芯片。

存储器 870 存储计算机程序，该计算机程序包括操作系统程序 872 和应用程序 871 等。

典型的操作系统如微软公司的 Windows®，苹果公司的 iOS®、谷歌公司开发的基于 Linux® 的安卓（Android® ）系统等。本申请前述任意一个实施例提供的方法可以通过软件的方式实现，可以认为是操作系统程序 872 的具体实现，作为操作系统程序 872 的一部分部署在智能手机上，当处理器 850 读取操作系统程序 872 并运行之后，本申请提供的方法操作系统程序 872 的一个功能可适时运行起来。具体的，本申请提供

的功能可以默认开启，也可以在用户的指令下开启。

需要说明的是，操作系统程序 872 相当于图 1 中的操作系统 120；应用程序 871 相当于图 1 中的应用 110。

存储器 870 可以是以下类型中的一种或多种：闪速（flash）存储器、硬盘类型存储器、微型多媒体卡型存储器、卡式存储器（例如 SD 或 XD 存储器）、随机存取存储器（random access memory, RAM）（例如前述实施例中提到的 DDR SDRAM）、只读存储器（read only memory, ROM）、电可擦除可编程只读存储器（electrically erasable programmable read-only memory, EEPROM）、可编程只读存储器（programmable ROM, PROM）、回滚保护存储块（replay protected memory block, RPMB）、磁存储器、磁盘或光盘。

在其他一些实施例中，存储器 870 也可以是因特网上的网络存储设备，系统可以对在因特网上的存储器 870 执行更新或读取等操作。

处理器 850 可以指示一个或多个处理单元，例如，处理器 850 可以包括一个或多个中央处理器，或者包括一个中央处理器和一个图形处理器，或者包括一个应用处理器和一个协处理器（例如微控制单元或神经网络处理器）。当处理器 850 包括多个处理单元时，这多个处理单元可以集成在同一块芯片上，也可以各自为独立的芯片。一个处理器可以包括一个或多个物理核。

存储器 870 还存储有除计算机程序之外的其他数据 873。

NPU 890 作为协处理器挂载到主处理器 850 上，用于执行主处理器 850 给它分配的任务。本申请实施例中，NPU890 可以在主处理器 850 的调度下实现管控决策过程中涉及的部分复杂算法。在其它一些实施例中，NPU890 也可以作为主处理器执行本申请提供的方法全部流程，或调用它的协处理器共同实现本申请提供的方法。

需要说明的是，图中示出的各个模块的连接关系仅为一种示例，本申请任意实施例提供的方法也可以应用在其它连接方式的智能手机中，例如所有模块或部分模块通过总线连接以实现通信。

图 12 为一种 NPU900 的结构示意图，该 NPU900 相当于图 11 中的 NPU890。如图所示，NPU900 与主处理器和外部存储器相连，这里的主处理器相当于图 11 中的处理器 850。NPU900 的核心部分为运算电路 903，通过控制器 904 控制运算电路 903 提取存储器中的数据并进行数学运算。

在一些实现中，运算电路 903 内部包括多个处理引擎（process engine, PE）。在一些实现中，运算电路 903 是二维脉动阵列。运算电路 903 还可以是一维脉动阵列或者能够执行例如乘法和加法这样的数学运算的其它电子线路。在另一些实现中，运算电路 903 是通用的矩阵处理器。

举例来说，假设有输入矩阵 A，权重矩阵 B，输出矩阵 C。 运算电路 903 从权重存储器 902 中取矩阵 B 相应的数据，并缓存在运算电路 903 的每一个 PE 上。运算电路 903 从输入存储器 901 中取矩阵 A 数据与矩阵 B 进行矩阵运算，得到的矩阵的部分结果或最终结果，保存在累加器（accumulator ）908 中。

统一存储器 906 用于存放输入数据以及输出数据。权重数据直接通过存储单元访问控制器 905（例如 direct memory access controller，DMAC）被搬运到权重存储器 902 中。输

入数据也通过存储单元访问控制器 905 被搬运到统一存储器 906 中。

总线接口单元 910（bus interface unit， BIU）用于 AXI（advanced extensible interface）总线与存储单元访问控制器 905 和取指存储器 909（ instruction fetch buffer）的交互。

总线接口单元 910 用于取指存储器 909 从外部存储器获取指令，还用于存储单元访问控制器 905 从外部存储器获取输入矩阵 A 或者权重矩阵 B 的原数据。

存储单元访问控制器 905 主要用于将外部存储器中的输入数据搬运到统一存储器 906 或将权重数据搬运到权重存储器 902 中或将输入数据数据搬运到输入存储器 901 中。

向量计算单元 907 通常包括多个运算处理单元，在需要的情况下，对运算电路 903 的输出做进一步处理，如向量乘、向量加、指数运算、对数运算、和/或大小比较等等。

在一些实现中，向量计算单元 907 能将经处理的向量存储到统一存储器 906 中。例如，向量计算单元 907 可以将非线性函数应用到运算电路 903 的输出，例如累加值的向量，用以生成激活值。在一些实现中，向量计算单元 907 生成归一化的值、合并值，或二者均有。在一些实现中，经处理的向量能够用作运算电路 903 的激活输入。

与控制器 904 连接的取指存储器 909 用于存储控制器 904 使用的指令。

统一存储器 906，输入存储器 901，权重存储器 902 以及取指存储器 909 均为 On-Chip 存储器。图中的外部存储器与该 NPU 硬件架构独立。

本实施例提供的方法中存在多个模型和多种算法，其中涉及到的复杂算法，例如搜索最优调度策略的算法可以由 NPU900 执行，然后将执行结果返回给主处理器，从而提高方法的运行效率。

需要说明的是，NPU 仅是举例，本申请提供的方案中的部分算法可以由其他类型的专用处理器处理。由于专用处理器在处理特定算法上的处理效率高于普通处理器，所以利用专用处理器处理方案中的部分算法，能够有效提升方案的整体运行效率，从而尽快找到合适的资源管控策略，并实现该资源管控策略。

需要说明的是，本实施例提供的方法也可以应用于非终端的计算机设备，例如云端服务器。

需要说明的是，以上实施例多以人脸识别方案为例介绍，但本申请提出的方法显然可以应用于除人脸识别之外的其它方案，本领域技术人员根据本申请提供的实现方式容易想到其它方案的类似实现方式。

需要说明的是，前述实施例中提出模块或单元的划分仅作为一种示例性的示出，所描述的各个模块的功能仅是举例说明，本申请并不以此为限。本领域普通技术人员可以根据需求合并其中两个或更多模块的功能，或者将一个模块的功能拆分从而获得更多更细粒度的模块，以及其他变形方式。

以上描述的各个实施例之间相同或相似的部分可相互参考。本申请中的“多个”若无特殊说明，指两个或两个以上，或“至少两个”。本申请中的“A/B”包括三种情况：“A”、“B”和“A 和 B”。

以上所描述的装置实施例仅仅是示意性的，其中所述作为分离部件说明的模块可以是或者也可以不是物理上分开的，作为模块显示的部件可以是或者也可以不是物理模块，即可以位于一个地方，或者也可以分布到多个网络模块上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。另外，本申请提供的装置实施例附图中，模块之间的连接关系表示它们之间具有通信连接，具体可以实现为一条或多条通信总线或信号线。本领域普通技术人员在不付出创造性劳动的情况下，即可以理解并实施。

以上所述，仅为本申请的一些具体实施方式，但本申请的保护范围并不局限于此。

权 利 要 求

1. 一种资源调度方法，其特征在于，包括：

确定设备当前的应用场景；

基于性能、功耗和温度在所述应用场景下分别的权重确定资源管控策略；

根据所述资源管控策略调度所述设备的资源。

2. 根据权利要求 1 所述的方法，其特征在于，所述基于性能、功耗和温度在所述应用场景下分别的权重确定资源管控策略，包括：

当性能劣化程度、功耗劣化程度以及温度劣化程度中的任意一个或多个达到预设条件时，基于性能、功耗和温度在所述应用场景下分别的权重确定资源管控策略；

其中：所述性能劣化程度用于指示所述设备的性能参数值与目标性能指标之间的差距，所述目标性能指标为满足所述应用场景下的性能体验的一个或多个性能参数的值；所述功耗劣化程度用于指示预测的可用电量与最低可用电量之间的差距，所述温度劣化程度用于指示预测的外壳温度和温度阈值之间的差距。

3. 根据权利要求 2 所述的方法，其特征在于，所述目标性能指标与所述应用场景具有对应关系。

4. 根据权利要求 3 所述的方法，其特征在于，还包括：对采集到的历史场景数据进行聚类以获得多个应用场景；为每个应用场景设置对应的目标性能指标。

5. 根据权利要求 2-4 任意一项所述的方法，其特征在于，所述设备的性能参数值是根据所述应用场景和所述设备当前的系统状态预测得到的，所述系统状态指示所述设备当前的资源使用情况。

6. 根据权利要求 5 所述的方法，其特征在于，所述系统状态包括资源供给信息和系统负载信息。

7. 根据权利要求 2-4 任意一项所述的方法，其特征在于，所述设备的性能参数值为采集得到的性能参数值。

8. 根据权利要求 2-7 任意一项所述的方法，其特征在于，所述功耗劣化程度为基于所述设备的电量使用规律以及所述设备的剩余电量确定的，其中，所述电量使用规律反映所述设备历史的充电和耗电情况。

9. 根据权利要求 8 所述的方法，其特征在于，所述预测的可用电量通过如下方式预测：

根据所述电量使用规律和当前的前台应用预测电量消耗，并获得所述预测的可用电量，所述预测的可用电量为所述剩余电量与所述电量消耗的差。

10. 根据权利要求 9 所述的方法，其特征在于，所述电量使用规律包括充电周期、用电规律和应用使用规律，其中所述充电周期代表历史上的充电间隔，所述用电规律代表历史上多个时刻以及每个时刻的剩余电量，所述应用使用规律代表历史上应用对电量的消耗信息；

所述最低可用电量是根据所述充电周期和所述用电规律确定的；所述电量消耗是根据所述应用使用规律和所述前台应用确定的。

11. 根据权利要求 10 所述的方法，其特征在于，所述充电周期为充电结束时刻到下次

充电开始时刻之间的时间段；所述用电规律包括历史上的多个时刻以及每个时刻对应的剩余电量；所述应用使用规律包括反映应用的使用时长和使用功耗的信息。

12. 根据权利要求 2-11 任意一项所述的方法，其特征在于，所述外壳温度是根据所述设备的当前外壳温度、所述设备内多个器件各自的温度升高预测值、以及每个器件温度的权重预测的。

13. 根据权利要求 12 所述的方法，其特征在于，所述温度升高预测值根据器件的温升函数和所述器件的当前资源配置确定。

14. 根据权利要求 2-13 任意一项所述的方法，其特征在于，所述温度阈值是根据所述设备当前所处的环境的温度确定的。

15. 根据权利要求 2-14 任意一项所述的方法，其特征在于，所述温度劣化程度包括预测的一段时间内的外壳温度的最大值和第一阈值的第一差距以及温度波动值与第二阈值的第二差距，其中所述温度波动值为所述最大值与该段时间内的最小值之差。

16. 根据权利要求 1-15 任意一项所述的方法，其特征在于，所述资源管控策略包括以下三项中的任意一项或多项：计算资源调度策略、应用管控策略以及外设管控策略。

17. 根据权利要求 1-16 任意一项所述的方法，其特征在于，所述资源管控策略为使得所述应用场景下用户整体体验最优的资源管控策略，其中所述用户整体体验根据性能、功耗和温度三者的用户体验以及三者分别在所述应用场景下的所述权重获得。

18. 根据权利要求 1-17 任意一项所述的方法，其特征在于，所述资源管控策略的确定过程利用机器学习或动态规划的方法。

19. 根据权利要求 2-18 任意一项所述的方法，其特征在于，所述目标性能指标包括以下四项参数中的任意一项或多项的值：目标帧率、丢帧率、低帧率占比和流畅度；其中，所述低帧率占比为低于目标帧率的帧数占总帧数的百分比。

20. 根据权利要求 1-19 任意一项所述的方法，其特征在于，所述应用场景包括应用内的使用场景。

21. 根据权利要求 1-20 任意一项所述的方法，其特征在于，所述确定资源管控策略包括：

根据设备的性能瓶颈或主导热区确定候选的资源管控策略，所述候选的资源管控策略能够提升所述性能瓶颈或降低所述主导热区的温度；

从所述候选的资源管控策略中，基于所述性能、功耗和温度分别的权重搜索出所述资源管控策略。

22. 根据权利要求 1-21 任意一项所述的方法，其特征在于，确定所述性能、功耗和温度各自的权重的因素包括以下任意一个或多个：所述设备的应用场景、所述性能、功耗和温度各自的劣化程度、以及所述性能、功耗和温度劣化判断的作用时域。

23. 根据权利要求 1-22 任意一项所述的方法，其特征在于，所述基于性能、功耗和温度在所述应用场景下分别的权重确定资源管控策略包括：

根据性能、功耗和温度三者各自的用户体验和各自的权重确定用户整体体验；

根据用户整体体验确定资源管控策略。

24. 根据权利要求 23 所述的方法，其特征在于，所述根据性能、功耗和温度三者各自

的用户体验满足程度和各自的权重确定用户整体体验包括：

根据性能劣化程度、功耗劣化程度、温度劣化程度分别确定所述性能的体验值、所述功耗的体验值和所述温度的体验值，其中，所述体验值用于指示用户体验满足程度，所述劣化程度用于指示目标指标与实际指标的差距，劣化程度越高，对应的体验值越低；

根据所述性能、功耗和温度的三个体验值以及三者在所述应用场景下分别的权重确定用户整体体验。

25. 一种资源调度方法，其特征在于，包括：

确定设备当前的应用场景；

根据所述应用场景，融合性能、功耗和温度三者各自的用户体验以获得用户整体体验；

根据所述用户整体体验确定并执行资源管控策略。

26. 根据权利要求 25 所述的方法，其特征在于，所述根据所述应用场景，融合性能、功耗和温度三者各自的用户体验以获得用户整体体验包括：

根据性能、功耗和温度三者各自的用户体验以及三者在所述应用场景下分别的权重获得用户整体体验。

27. 根据权利要求 25 或 26 所述的方法，其特征在于，所述应用场景包括应用内的使用场景。

28. 根据权利要求 25-27 任意一项所述的方法，其特征在于，所述确定资源管控策略包括：

根据设备的性能瓶颈或主导热区确定候选的资源管控策略，所述候选的资源管控策略能够提升所述性能瓶颈或降低所述主导热区的温度；

从所述候选的资源管控策略中，根据所述用户整体体验确定搜索出所述资源管控策略，其中，所述资源管控策略满足所述用户整体体验。

29.根据权利要求 26-28 任意一项所述的方法，其特征在于，确定所述性能、功耗和温度各自的权重的因素包括以下任意一个或多个：所述设备的应用场景、所述性能、功耗和温度各自的劣化程度、以及所述性能、功耗和温度劣化判断的作用时域。

30.一种计算机设备，其特征在于，所述计算机设备包括处理器和存储器，其中：所述存储器用于存储程序；所述处理器用于读取所述程序并实现如权利要求 1-29 中任意一项所述的方法。

31.根据权利要求 30 所述的计算机设备，其特征在于，所述计算机设备为终端设备。

32.一种计算机存储介质，其特征在于，包括程序，所述程序在被一个或多个处理单元执行时实现如权利要求 1-29 中任意一项所述的方法。

33.一种计算机程序产品，其特征在于，包括程序，所述程序在被一个或多个处理单元执行时实现如权利要求 1-29 中任意一项所述的方法。

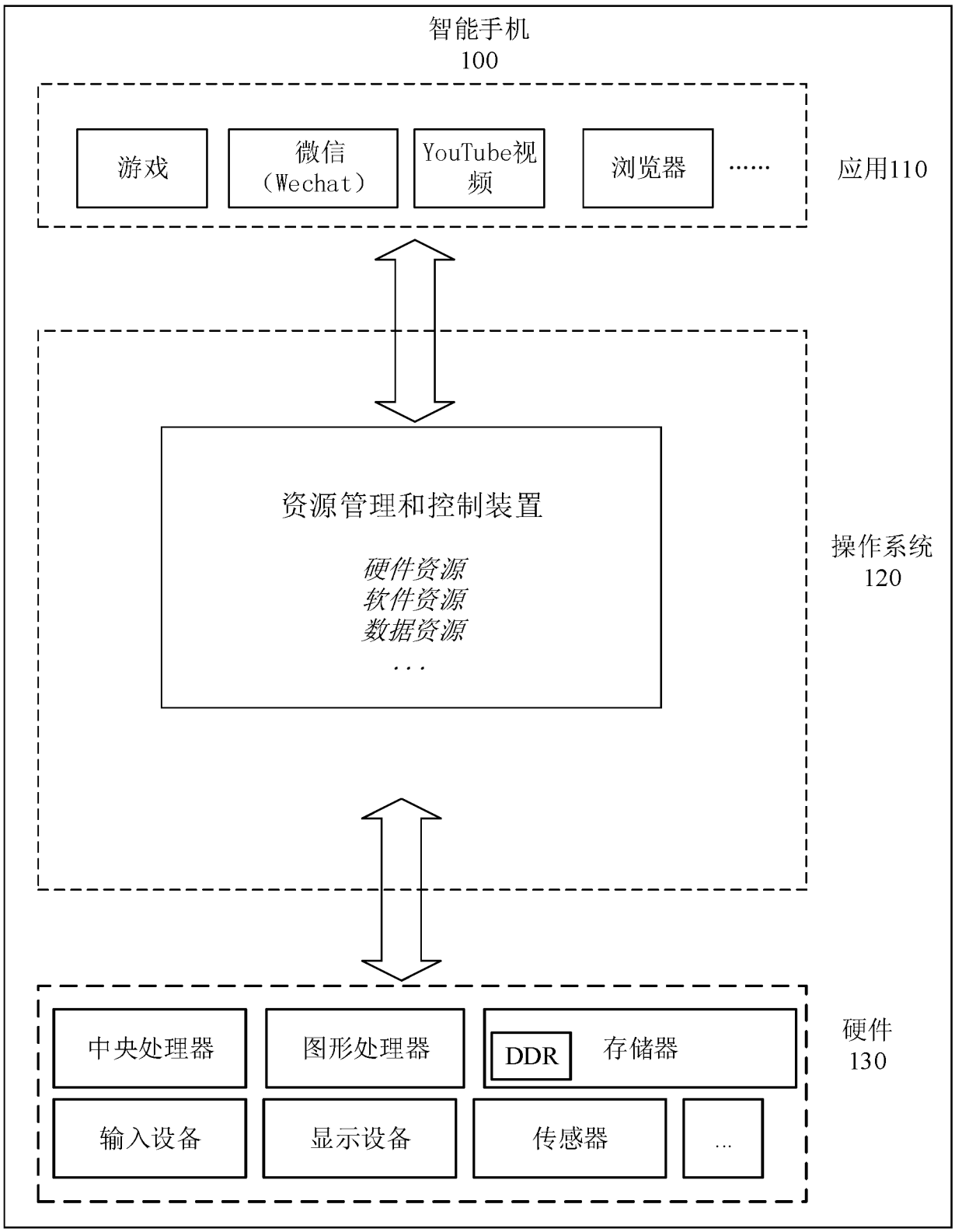

智能手机
100
游戏
微信
（Wechat）
YouTube视频
浏览器
……
应用110
资源管理和控制装置
硬件资源
软件资源
数据资源
...
操作系统
120
中央处理器
图形处理器
DDR
存储器
输入设备
显示设备
传感器
...
硬件
130

图 1

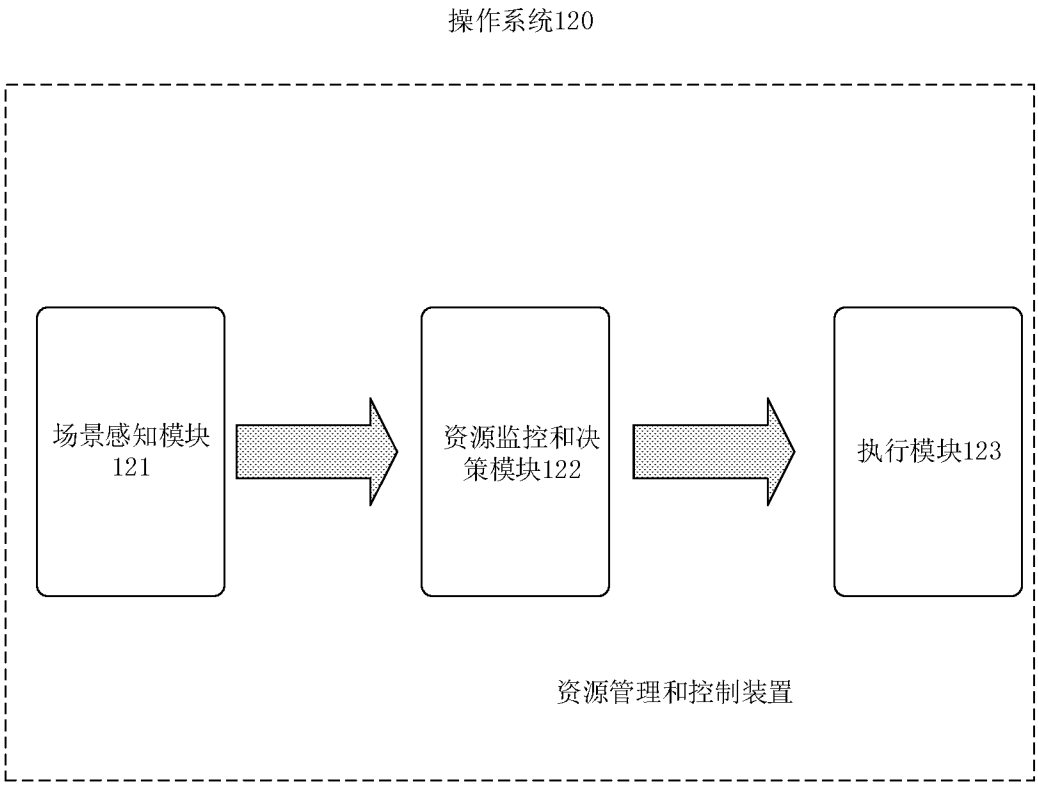

操作系统120
场景感知模块121
资源监控和决策模块122
执行模块123
资源管理和控制装置

图 2a

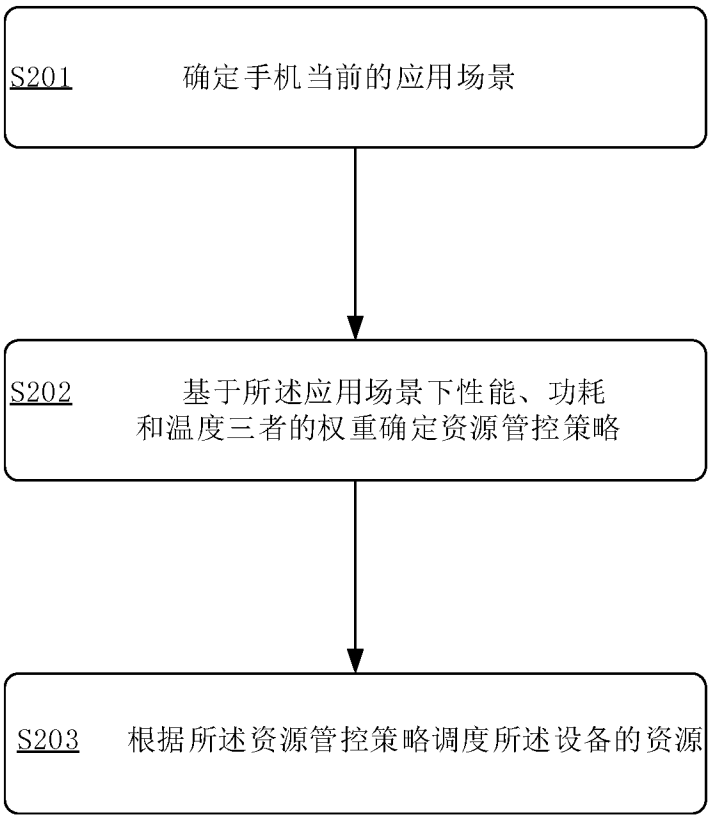

S201
确定手机当前的应用场景
S202
基于所述应用场景下性能、功耗和温度三者的权重确定资源管控策略
S203
根据所述资源管控策略调度所述设备的资源

图 2b

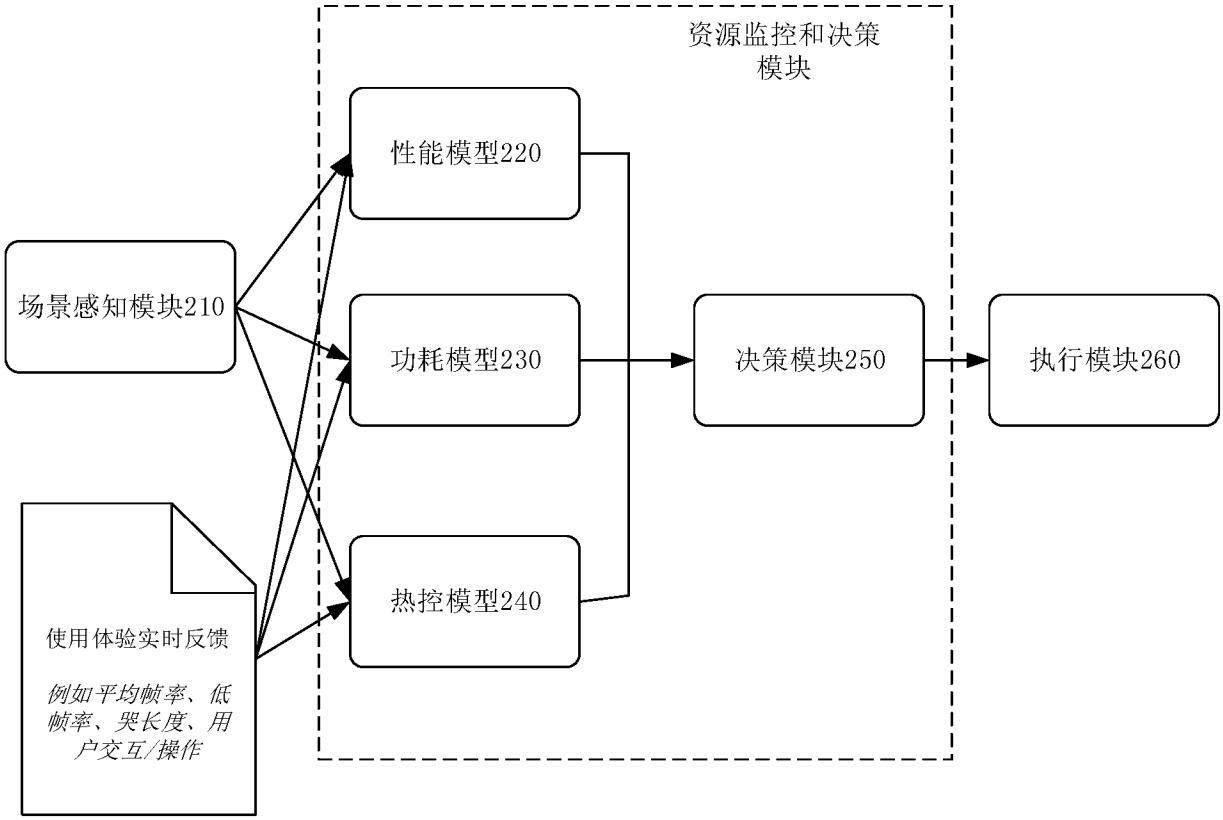

资源监控和决策模块
性能模型220
场景感知模块210
功耗模型230
决策模块250
执行模块260
热控模型240
使用体验实时反馈
例如平均帧率、低帧率、哭长度、用户交互/操作

图 3

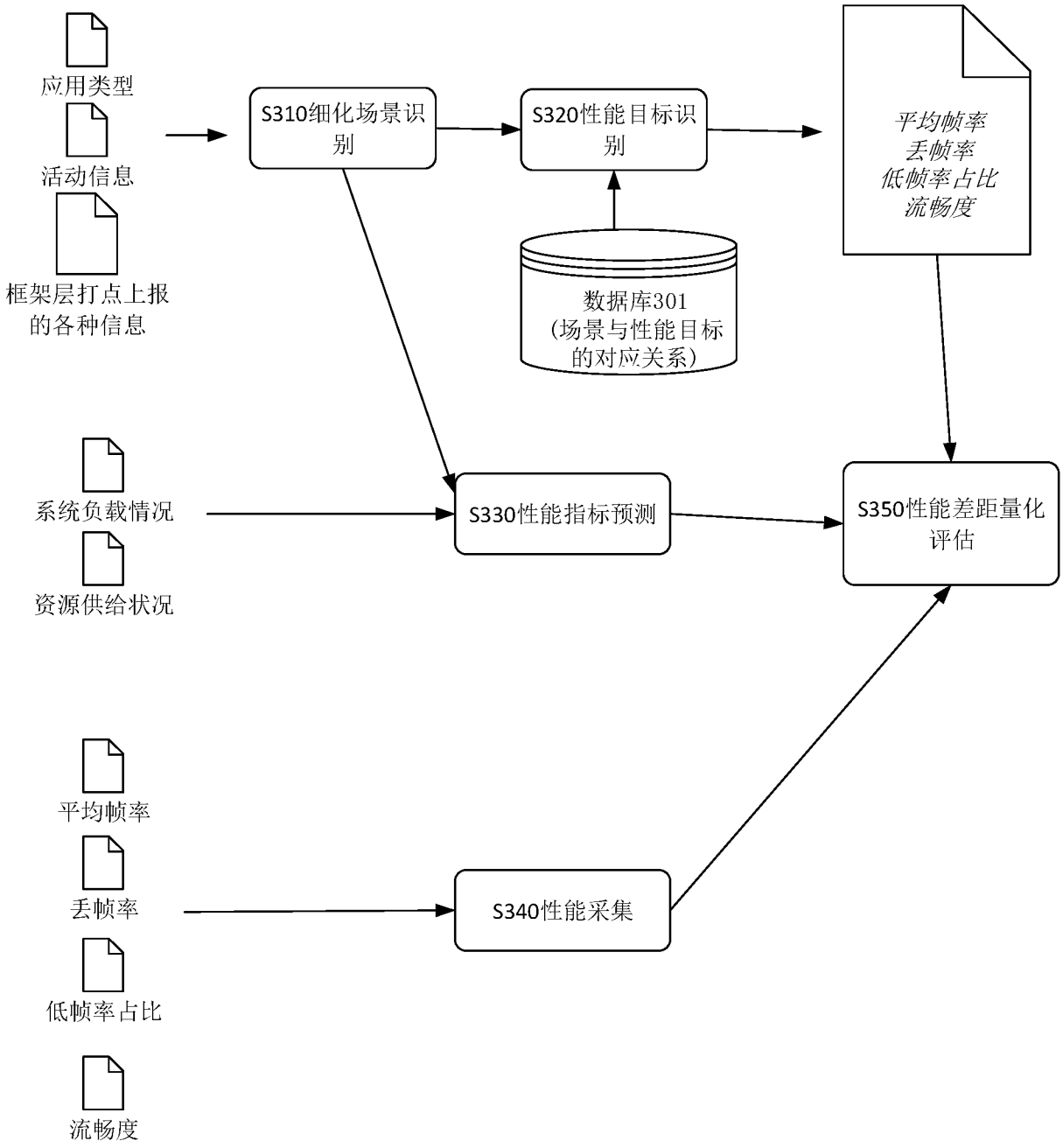

应用类型
活动信息
框架层打点上报的各种信息
S310细化场景识别
S320性能目标识别
数据库301
(场景与性能目标的对应关系)
平均帧率
丢帧率
低帧率占比
流畅度
系统负载情况
资源供给状况
S330性能指标预测
S350性能差距量化评估
平均帧率
丢帧率
低帧率占比
流畅度
S340性能采集

图 4

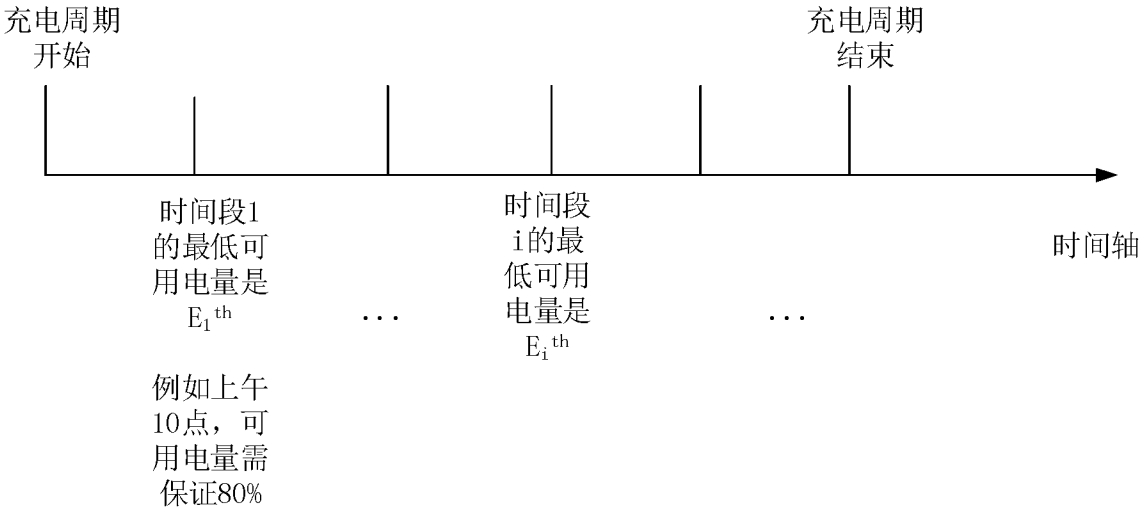

充电周期
开始
充电周期
结束
时间段1
的最低可
用电量是
E_1^{th}
例如上午
10点，可
用电量需
保证80%
...
时间段
i的最
低可用
电量是
E_i^{th}
...
时间轴

图 5

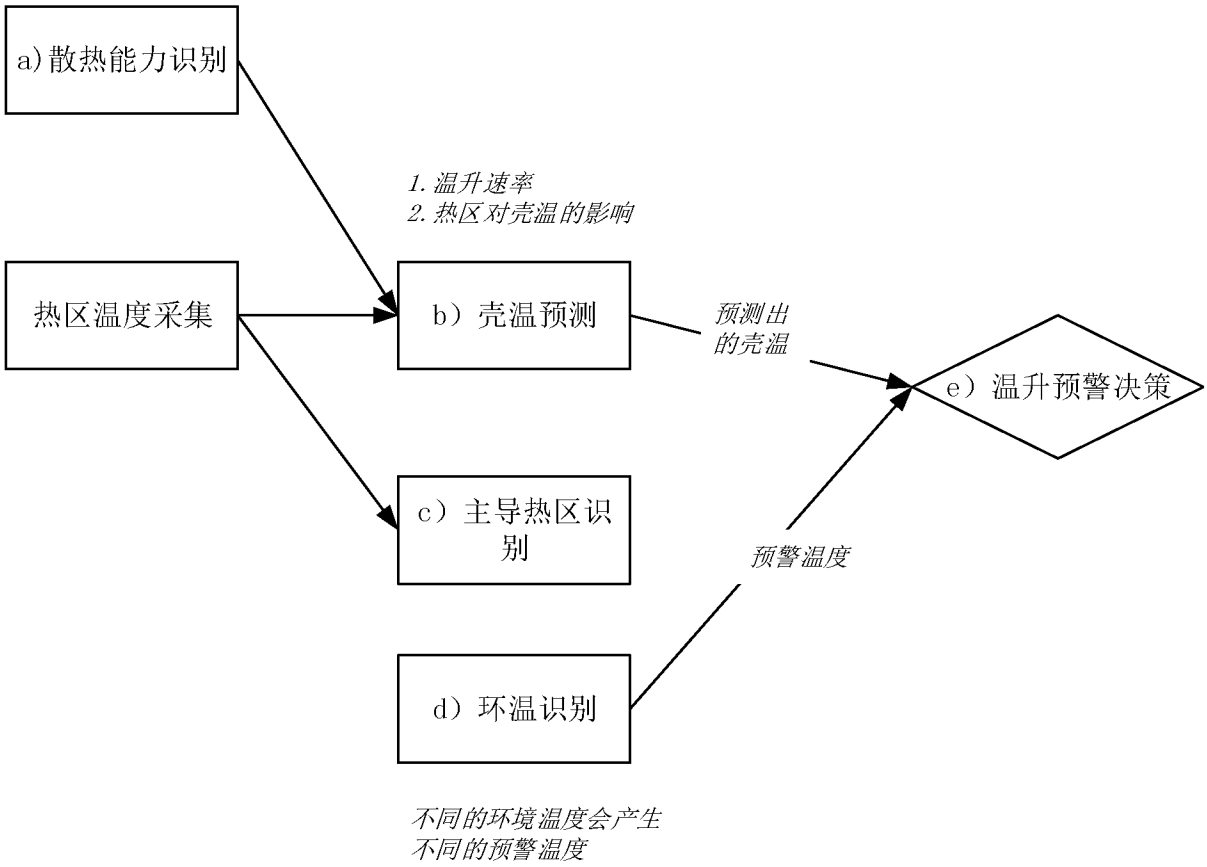

a)散热能力识别
1. 温升速率
2. 热区对壳温的影响
热区温度采集
b）壳温预测
预测出的壳温
e）温升预警决策
c）主导热区识别
预警温度
d）环温识别
不同的环境温度会产生不同的预警温度

图 6

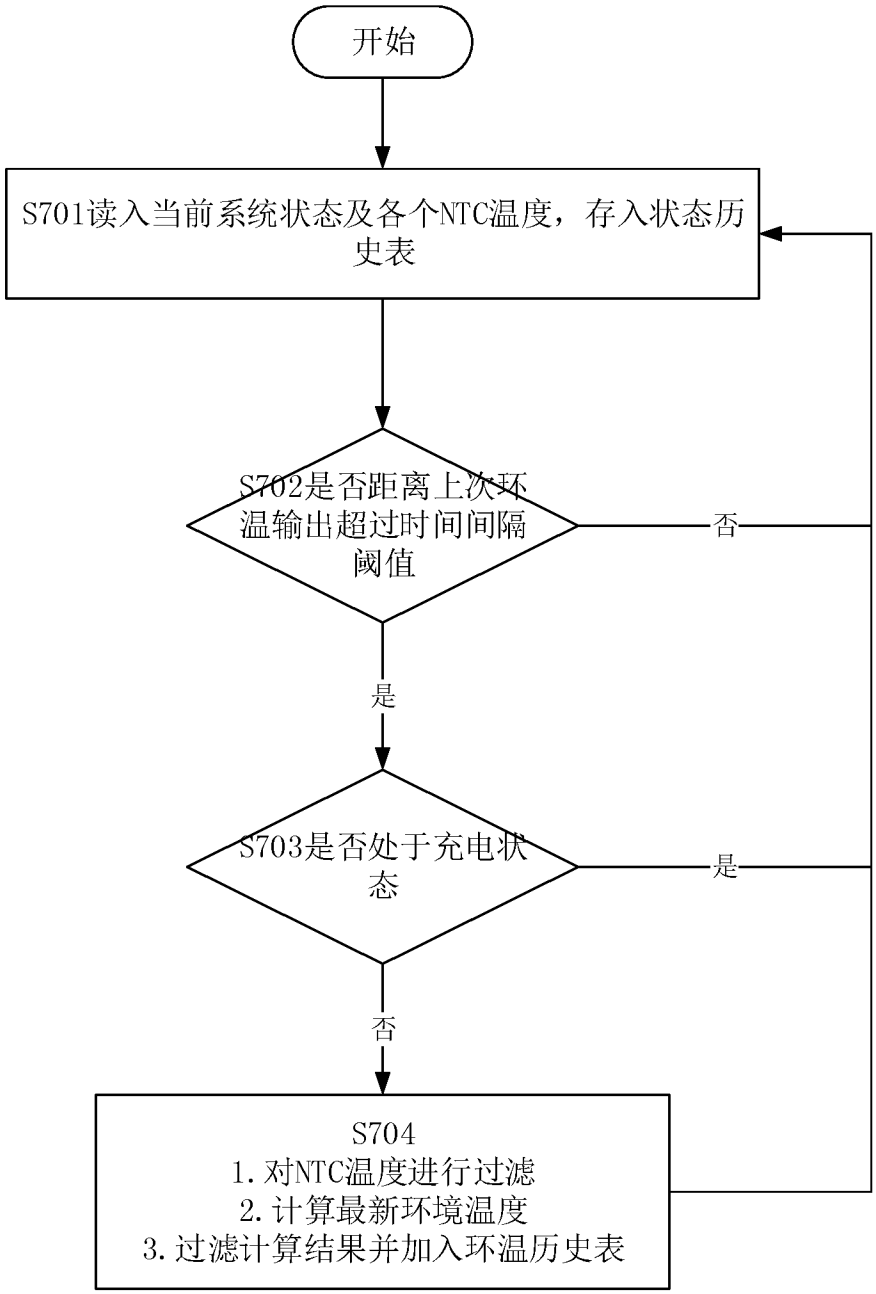

开始
S701读入当前系统状态及各个NTC温度，存入状态历史表
S702是否距离上次环温输出超过时间间隔阈值
否
是
S703是否处于充电状态
是
否
S704
1.对NTC温度进行过滤
2.计算最新环境温度
3.过滤计算结果并加入环温历史表

图 7A

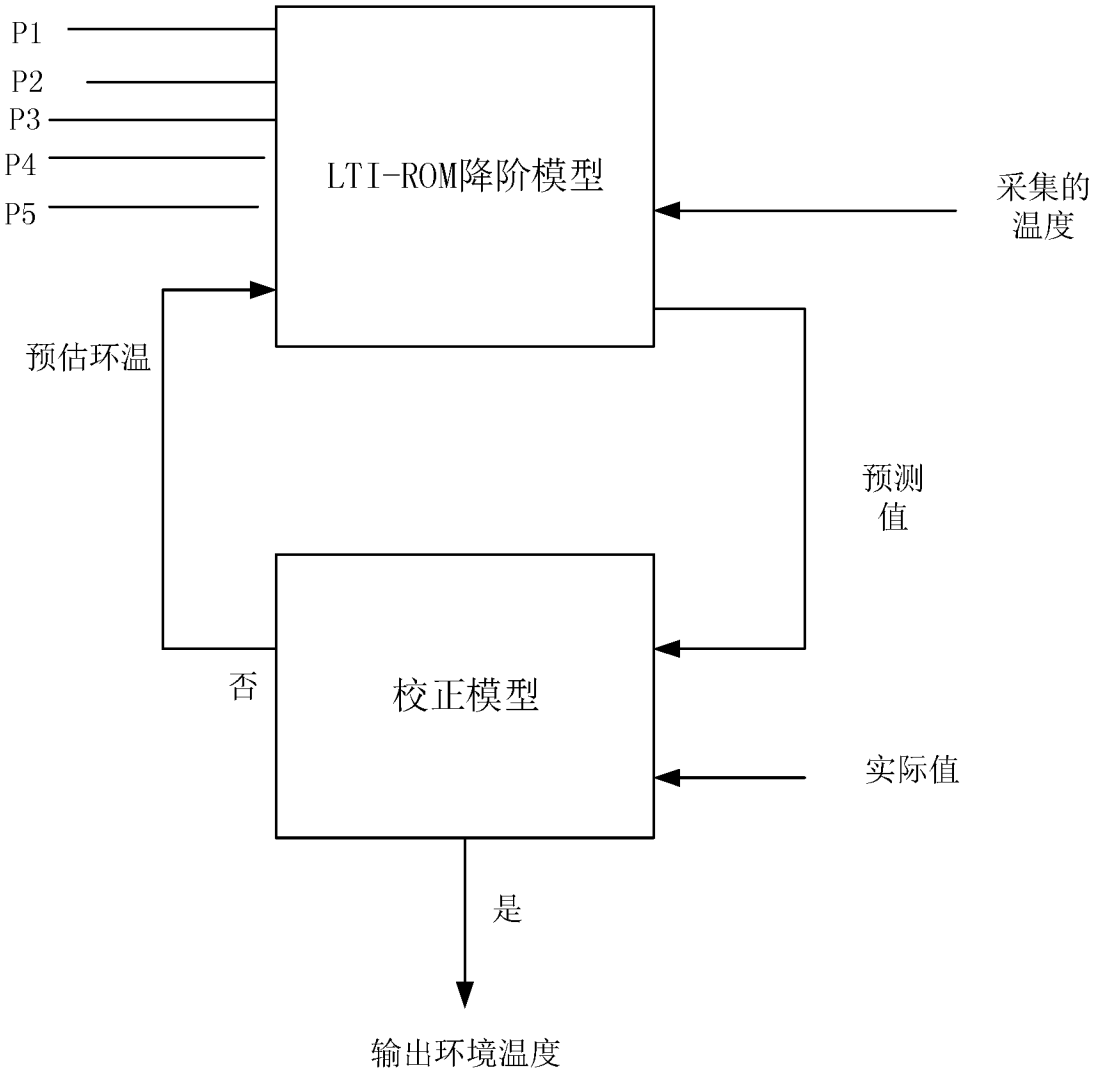


图 7B

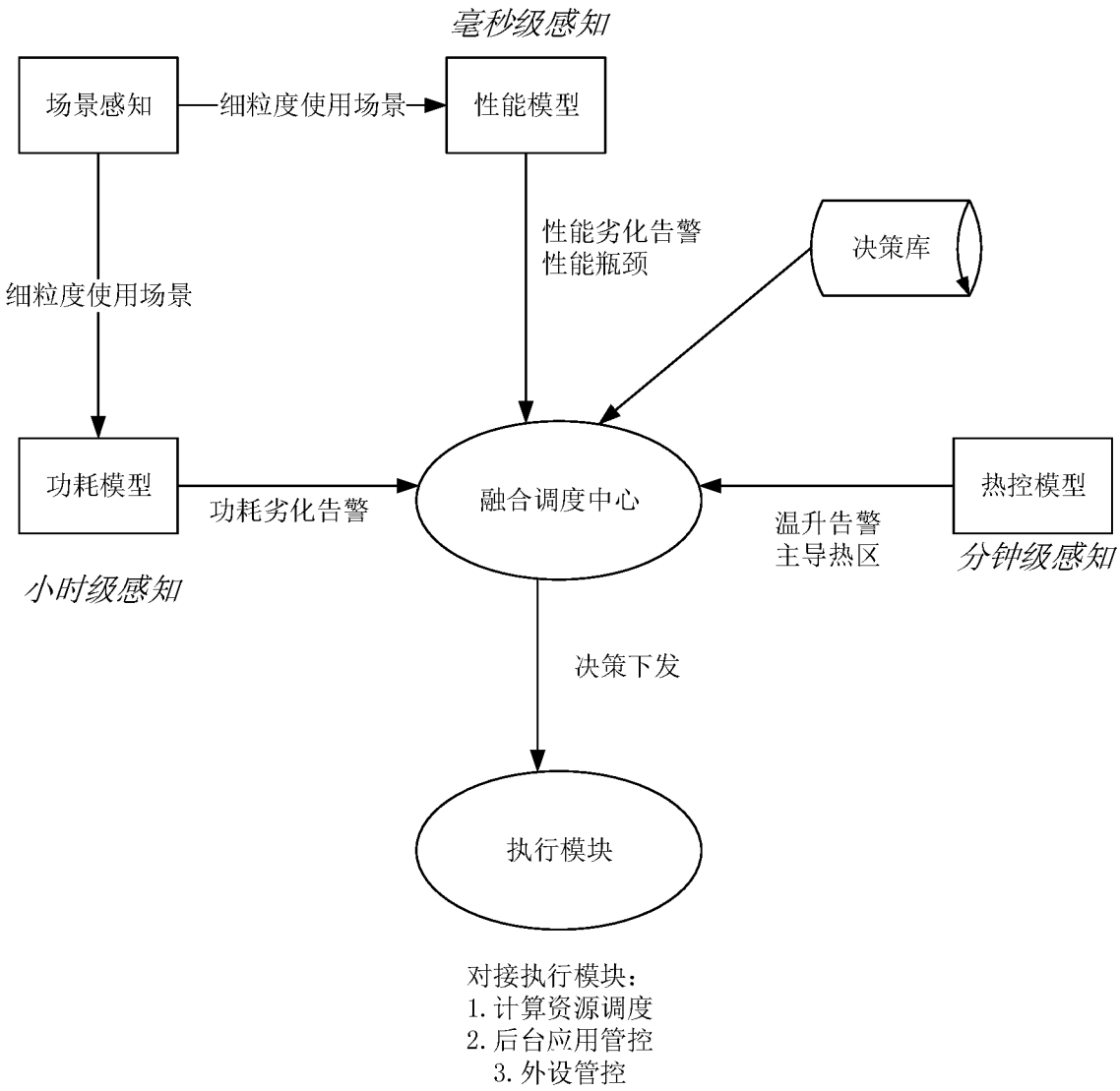

毫秒级感知
场景感知
细粒度使用场景
性能模型
性能劣化告警
性能瓶颈
决策库
细粒度使用场景
功耗模型
功耗劣化告警
融合调度中心
温升告警
主导热区
热控模型
小时级感知
分钟级感知
决策下发
执行模块
对接执行模块：
1. 计算资源调度
2. 后台应用管控
3. 外设管控

图 8

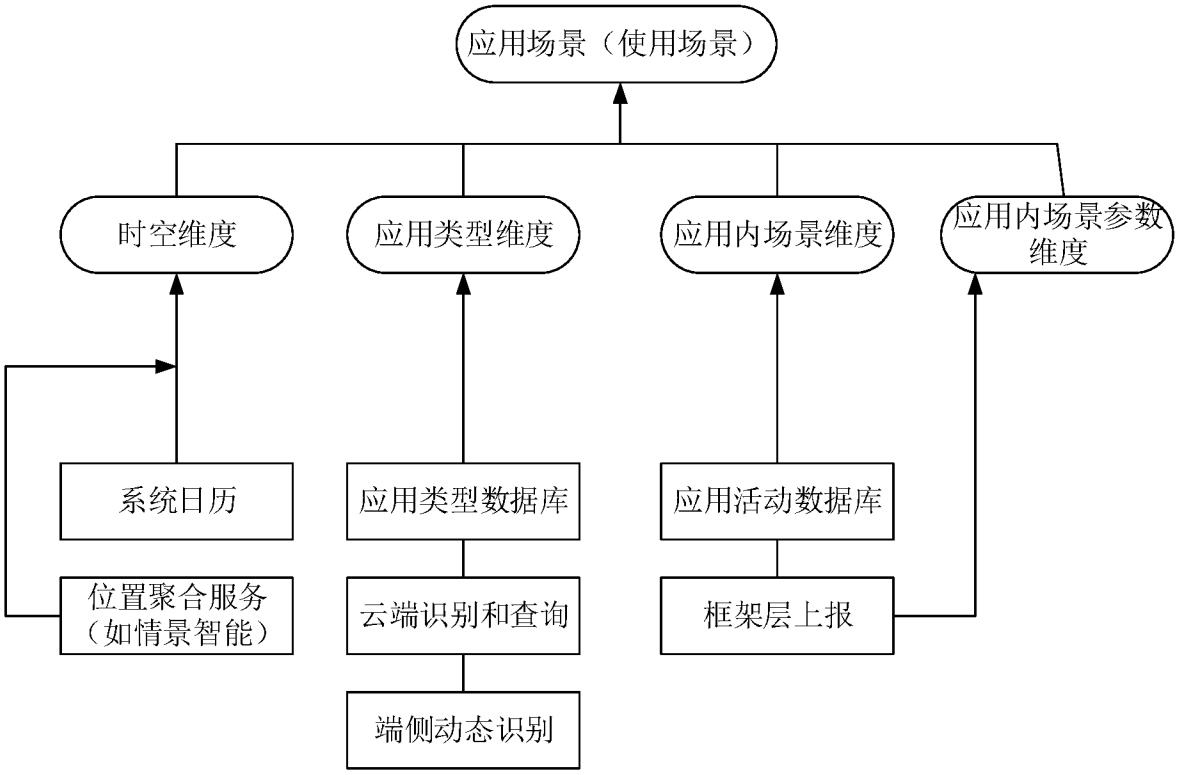

应用场景（使用场景）
时空维度
应用类型维度
应用内场景维度
应用内场景参数维度
系统日历
应用类型数据库
应用活动数据库
位置聚合服务（如情景智能）
云端识别和查询
框架层上报
端侧动态识别

图 9

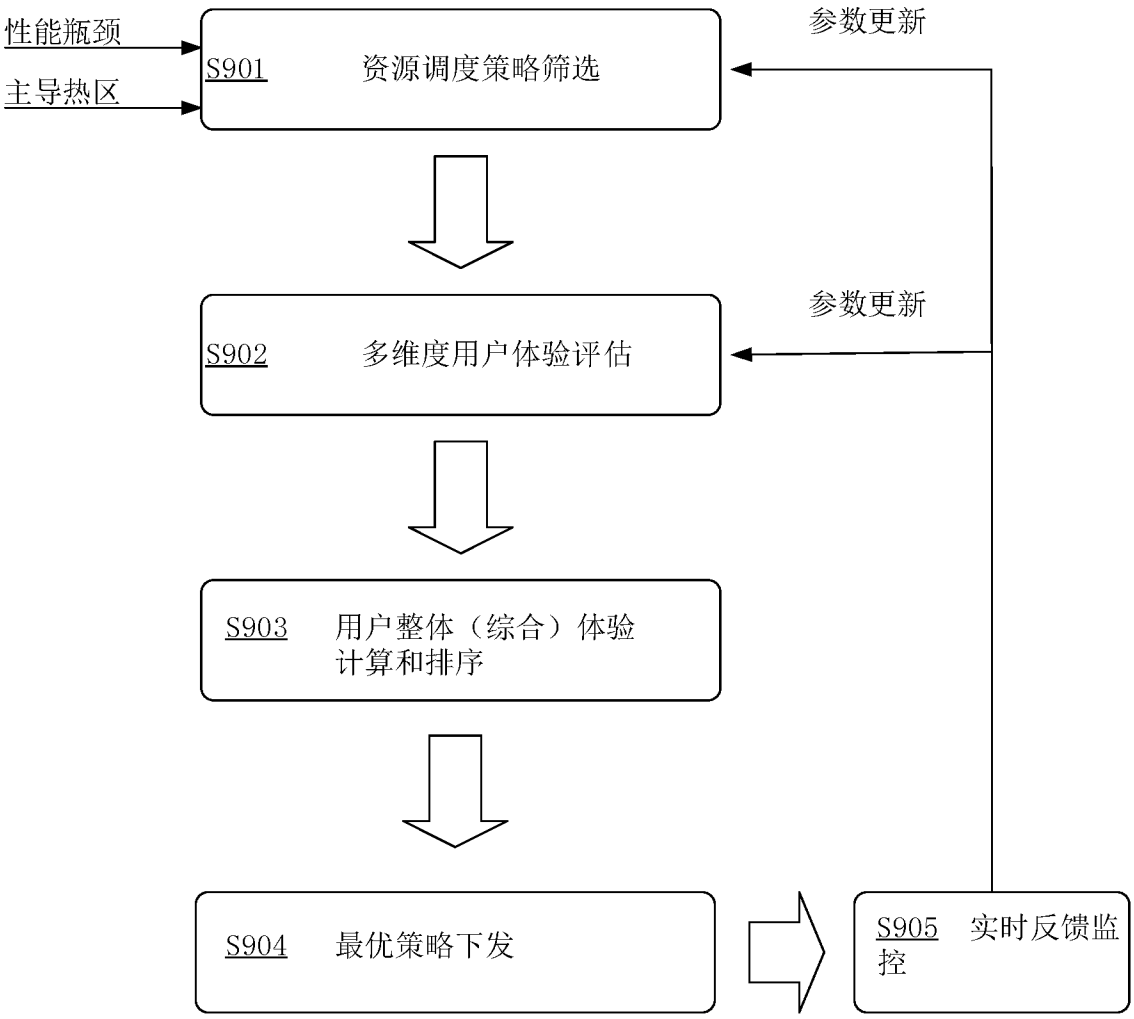

性能瓶颈
主导热区
S901 资源调度策略筛选
参数更新
S902 多维度用户体验评估
参数更新
S903 用户整体（综合）体验计算和排序
S904 最优策略下发
S905 实时反馈监控

图 10

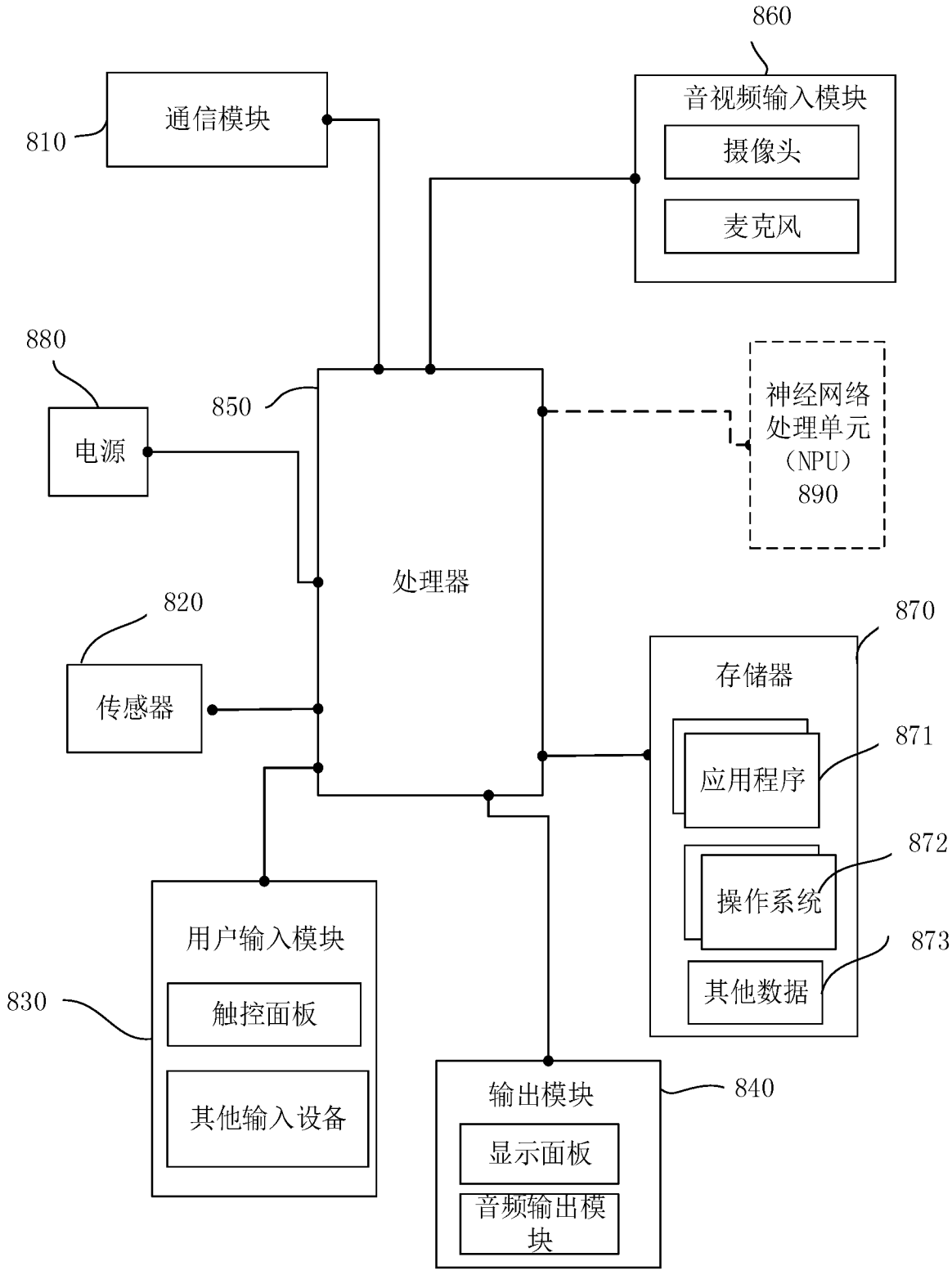

860
音视频输入模块
摄像头
麦克风
通信模块
810
880
电源
850
处理器
神经网络处理单元（NPU）890
820
传感器
870
存储器
871
应用程序
872
操作系统
873
其他数据
用户输入模块
830
触控面板
其他输入设备
输出模块
840
显示面板
音频输出模块

图 11

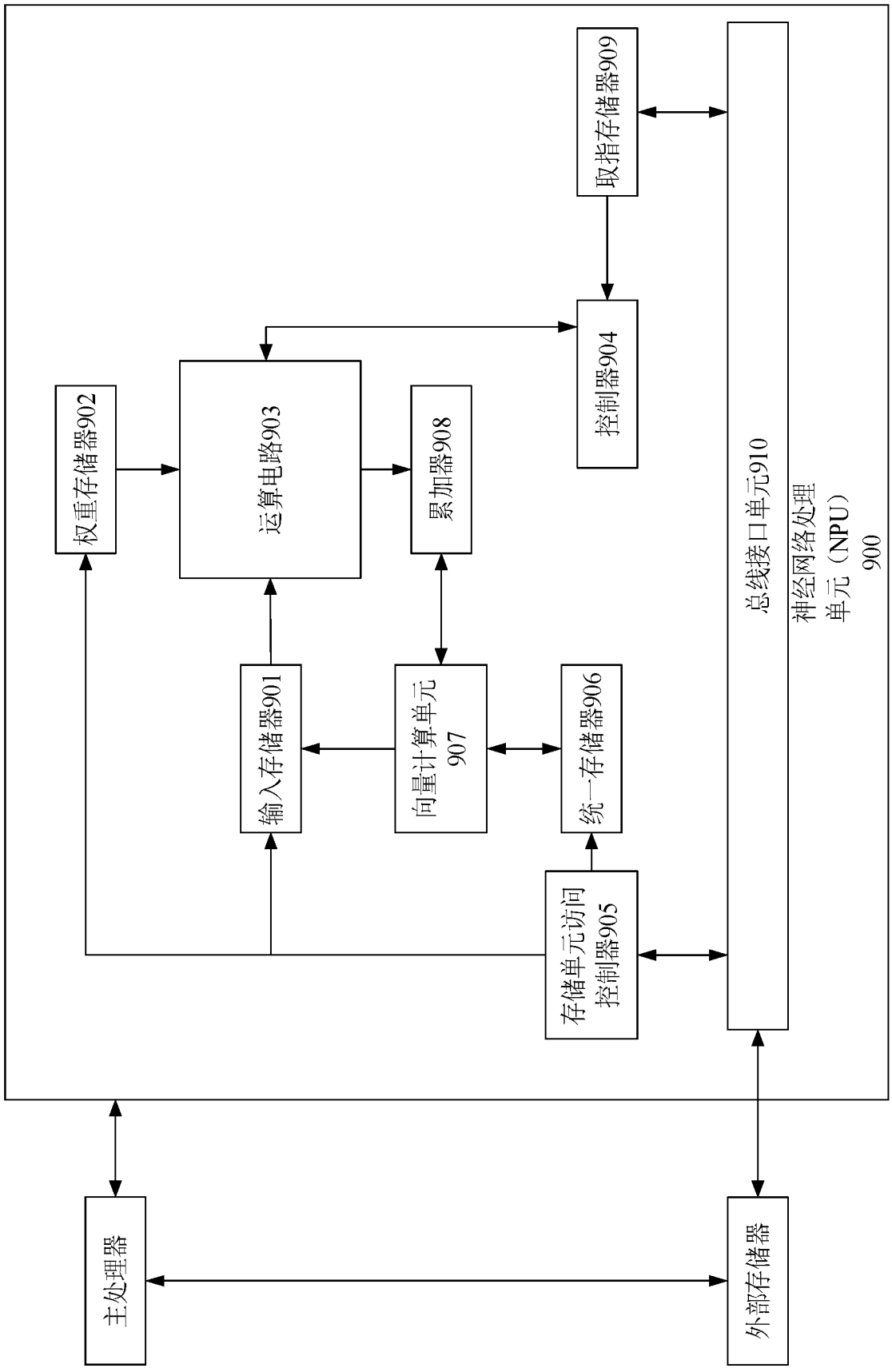

主处理器
外部存储器
神经网络处理单元（NPU）900
总线接口单元910
取指存储器909
控制器904
权重存储器902
运算电路903
累加器908
输入存储器901
向量计算单元907
统一存储器906
存储单元访问控制器905

图 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2019/104292

A. CLASSIFICATION OF SUBJECT MATTER

G06F 1/3212(2019.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F; H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNKI; CNPAT; WPI; EPODOC; IEEE: 场景, 环境, 策略, 调度, 温度, 资源, 性能, 功耗, 权重, 散热, 能耗, 参数, 配置, 分配, scene, environment, policy, schedule, temperature, resource, performance, consum+, weight, parameter, configur+, allocat+

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 109960395 A (HUAWEI TECHNOLOGIES CO., LTD.) 02 July 2019 (2019-07-02) claims 1-33	1-33
X	CN 107577533 A (GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP., LTD.) 12 January 2018 (2018-01-12) description, paragraphs 32-150	1-33
A	CN 106774786 A (MEIZU TECHNOLOGY CO., LTD.) 31 May 2017 (2017-05-31) entire document	1-33
A	US 2015186184 A1 (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 02 July 2015 (2015-07-02) entire document	1-33
A	CN 107515787 A (GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CO., LTD.) 26 December 2017 (2017-12-26) entire document	1-33

☐ Further documents are listed in the continuation of Box C.

☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 October 2019	**03 December 2019**

Name and mailing address of the ISA/CN	Authorized officer
China National Intellectual Property Administration (ISA/CN) **No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088** **China**	
Facsimile No. **(86-10)62019451**	Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2019/104292

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	109960395	A	02 July 2019	None			
CN	107577533	A	12 January 2018	WO	2019042169	A1	07 March 2019
CN	106774786	A	31 May 2017	None			
US	2015186184	A1	02 July 2015	KR	20150075499	A	06 July 2015
CN	107515787	A	26 December 2017	WO	2019042171	A1	07 March 2019

国际检索报告

国际申请号 PCT/CN2019/104292

A. 主题的分类

G06F 1/3212(2019.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F; H04L

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用）)

CNKI;CNPAT;WPI;EPODOC;IEEE: 场景，环境，策略，调度，温度，资源，性能，功耗，权重，散热，能耗，参数，配置，分配，scene, environment, policy, schedule, temperature, resource, performance, consum+, weight, parameter, configur+, allocat+

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
PX	CN 109960395 A (华为技术有限公司) 2019年 7月 2日 (2019-07-02) 权利要求1-33	1-33
X	CN 107577533 A (广东欧珀移动通信有限公司) 2018年 1月 12日 (2018-01-12) 说明书第32-150段	1-33
A	CN 106774786 A (珠海市魅族科技有限公司) 2017年 5月 31日 (2017-05-31) 全文	1-33
A	US 2015186184 A1 (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 2015年 7月 2日 (2015-07-02) 全文	1-33
A	CN 107515787 A (广东欧珀移动通信有限公司) 2017年 12月 26日 (2017-12-26) 全文	1-33

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

"A" 认为不特别相关的表示了现有技术一般状态的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"P" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2019年 10月 11日	2019年 12月 3日

ISA/CN的名称和邮寄地址	受权官员
中国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088	汪德闯
传真号 (86-10)62019451	电话号码 86-(10)-53961791

表 PCT/ISA/210 (第2页) (2015年1月)

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2019/104292

检索报告引用的专利文件	公布日（年/月/日）	同族专利	公布日（年/月/日）
CN 109960395 A	2019年 7月 2日	无	
CN 107577533 A	2018年 1月 12日	WO 2019042169 A1	2019年 3月 7日
CN 106774786 A	2017年 5月 31日	无	
US 2015186184 A1	2015年 7月 2日	KR 20150075499 A	2015年 7月 6日
CN 107515787 A	2017年 12月 26日	WO 2019042171 A1	2019年 3月 7日