思路、逻辑

开会目的是项目介绍，要尽可能给第一次接触该项目的人讲清楚问题，但是讲的对象对调度算法有比较好的理解。

我们要介绍的内容有哪些？

**首先，要介绍项目的相关内容，项目的背景，要用在什么地方，最终实现一个什么东西出来。**

**其次，调度算法具体要做什么，整体的调度流程是什么样的**

**最后，以一个实际案例为例子，介绍一个实际的调度问题，在该问题下，把整个流程走一下。**

**补充：**

**当前我们建模方式是如何进行的，对于这个问题，我们之前用的解决方式和目前的一些尝试，当前的方法还存在的问题。**

我会从以下这几个方面对我们的项目做一个介绍。**首先**我会介绍一下项目背景，对当前项目的应用场景、用在什么地方、以及最终的实现目标做一个介绍。

我们这个项目的具体内容是做一个调度算法，应用到实际的生产中，**之后**我会对我们调度算法的目标进行介绍、包括调度算法的输入与输出，包括调度问题中的一些问题与难点。

**在理论上进行了一定的介绍后**，我会以一个实际的案例为例，更加详细说明我们调度算法的目标，以及我们调度算法在项目中要执行的具体的流程是什么。

**最后**，我会对我们当前的一些尝试方案进行说明，当前方案存在的一些问题与难点。

下面是具体的介绍：

Page1：首先是项目背景的介绍，项目的场景是实验室自动化场景。实验自动化场景指的是在生物领域或医疗领域中、例如医药研发、医学检验等检测领域，需要实验人员对待检测样本进行一系列的检测操作，随着自动化技术的发展，医疗检测设备正逐渐向着自动化的趋势发展，当前实验室中对于批量操作的任务，自动化设备正逐渐取代人工，大大提高了检测的效率和准确率。例如核酸检测任务的流水线中，如右图所示，该场景中通常有大量的待检测样本需要进行一系列的加工处理，在该场景下，调度系统是该场景的一个核心模块，一个好的调度算法能够对仪器的使用进行合理规划，大大提高设备的利用率以及缩短检测时间，因此在该场景下，如何设计调度算法，在产生合理的调度方案的情况下，同时尽可能早的完成检测任务是目前场景下的一个主要的问题。

这是我们项目的一个背景介绍，项目要用在什么地方，场景主要用于类似核酸检测、基因测序等场景下，整体来说调度算法的输入是给出所有待检测样品的工艺序列，以及可用设备和相关的设备信息，工艺序列中的每个工艺在特定设备上执行。调度算法的输出是给出调度的甘特图，甘特图上的信息包括每个设备执行某个工艺的开始时间、结束时间。算法给出甘特图后，用于指导和控制自动化设备进行相关操作。

之后我会对调度算法具体做什么、整体调度流程做一个介绍，包括调度问题中的一些难点。

我觉得还是以一个具体的例子来说比较好，重点是以具体的例子说明调度算法的目标问题是什么，如果单纯介绍理论还是不好说清楚。

所以第二部分的内容简单介绍即可。

Page2：调度算法

组会分享ppt 稿：

·标题：调度 模型与算法

大家好今天我来介绍一下我目前在研究的调度问题，今天我主要对调度模型、调度算法进行一个介绍。

·contents：下面我会从以下几个方面介绍以下调度问题，first介绍一下调度模型和调度问题，之后介绍一下调度算法，其中包括调度算法的种类、一个启发式的列表调度算法、以及一个实际应用中的实例，介绍一个调度问题种的实际的case。

·Page1：首先介绍一下什么时候需要进行调度，什么是调度呢。当一个系统中存在着对资源的占用，这些资源就必须通过一些方法进行调度，例如计算机执行计算任务时，存在多个处理器资源，计算任务如何分配计算资源、缓冲资源的分配、网络带宽的分配等都属于存在竞争的资源，当存在竞争时就需要进行调度。

·Page2：描述一个调度问题要从以下三个方面出发

分别是1.任务的描述 2.资源的描述 3. 调度目标的描述。

下面我分别详细介绍一下这三个方面

·Page3：首先是任务的描述。

任务的模型通常由任务图来表示，如右图所示。当前存在若干个任务，在任务图中编号为1-10，其中不同的任务之间存在着一定的依赖关系，在图中以有向边的形式展示，另外任务可能为周期任务，每个任务的属性可能有以下这几类，例如：执行任务的开销、传输开销、任务到达时间、任务截止时间、任务是否存在优先级等等

通过以上任务图、以及任务的属性的描述来对要完成的任务进行建模。

·Page4：第二点是对资源进行建模

1可用资源集合通常包含一系列设备集合或者处理器集合，设备\处理器之间由网络进行连接通信。

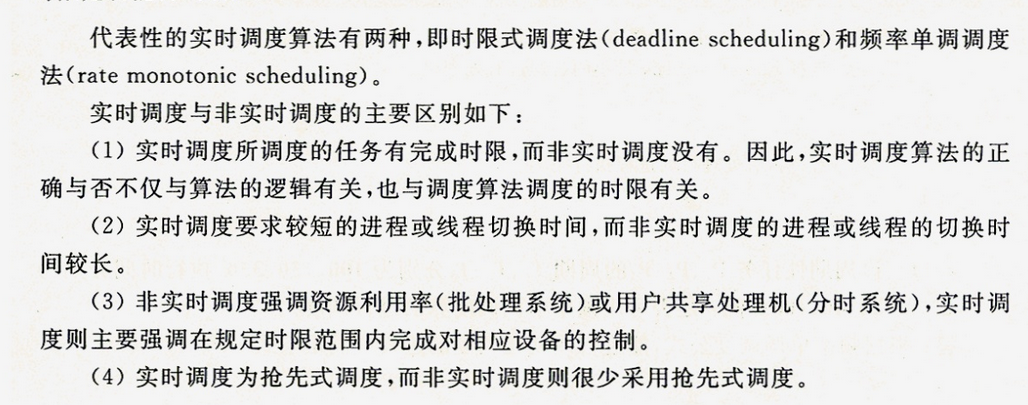
2 设备、处理器建模：包括设备处理能力、处理速度、处理器能量消耗等。

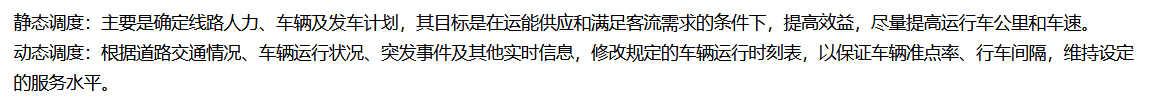
3 网络模型：包括网络的拓扑结构、网络的带宽、消息和能源的消耗等、例如处理器网络模型、数据中心网络模型、移动云网络模型等等。

·Page 5：描述一个调度问题第三点是描述调度问题的优化目标：例如最小化完成时间、在截止日期前交付、最大化吞吐量、尽量减少数据\消息的传输、最小化能量消耗等。

·Page 6:上面介绍的三个要素是描述一个调度问题的三方面。那么当前调度问题的分类依照不同的判别条件可以分为以下几类：

1 依据调度发生的时间可以分为：实时调度和非实时调度



2 静态调度和动态调度：静态调度：在条件给定的情况下进行调度，动态调度强调依据实际情况进行调度

1. 离线调度和在线调度

1、离线（ off-line ）和在线（ on-line ）调度

根据获得调度信息的时机，调度算法可以分为离线调度和在线调度两类。对 于离线调度算法，运行过程中使用的调度信息在系统运行之前就确定了，如时间驱动的调度。离线调度算法具有确定性，但缺乏灵活性 ，适用于特征能够预先确定，且不容易发生变化的应用。在线调度算法的调度信息则在系统运行过程中动态获得，如优先级驱动的调度（如 EDF,RMS 等）。在线调度算法在形成最佳调度决策上具有较大的灵活性。

2，抢占（ preemptive ）和非抢占（ non-preemptive ）调度

根据 任务在运行过程中能否被打断的处理情况。调度算法分为抢占式调度和非抢占式调度两类。在抢占式调度方法中，正在运行的任务可能被其他任务打断。在非抢占式调度算法中，一旦任务开始运行，该任务只有在运行完成而主动放弃 CPU 资源，或是因为等待其他资源被阻塞的情况下才会停止运行。实时内核大都采用了抢占式调度算法，使关键任务能够打断非关键任务执行，确保关键任务的截止时间能够得到满足。相对来说，抢占式调度算法要更复杂些，且需要更多的资源，并可能在使用不当的情况下造成低优先级任务出现长时间得不到执行的情况。非抢占式调度算法常用于那些任务需要按照预先确定的顺序执行，且只有当任务主动放弃 CPU 资源后，其他任务才能得到执行的情况。

3， 静态（ static ）和动态（ dynamic ）调度

根据任务优先级的确定时机，调度算法分为静态调度和动态调度两类。在静态调度算法中，所有任务的优先级在设计时已经确定下来，且在运行过程中不会发生变化（如 RMS ）。在动态调度算法中，任务的优先级则在运行过程中确定，并可能不断发生变化（如 EDF ）。静态调度算法适用于能够完全把握系统中所有任务及其时间约束（如截至时间、运行时间、优先顺序和运行过程中的到达时间）特性的情况。静态调度比较简单，但缺乏灵活性，不利于系统扩展；动态调度有足够的灵活性来处理变化的系统情况，但需要消耗更多的系统资源。

1. 确定性调度问题和随机调度问题，确定性调度问题中任务的执行时间确定，随机调度问题中

根据生产环境的特点,可将调度分为确定性调度和随机性调度。前者是指加工时间和其他参数是已知的、确定的量;而后者的加工时间和有关参数是随机的变量。

根据加工任务或被加工工件的特征,可将调度分为静态调度和动态调度。静态调度是指所有待安排加工的工件均处于待加工状态,因而进行一次调度后,各作业的加工被确定,在以后的加工过程中就不再改变;动态调度是指作业依次进入待加工状态,各种作业不断进入系统接受加工,同时完成加工的作业又不断离开,还要考虑作业环境中不断出现的不可预测的动态扰动。

·Page 7：调度问题目前有以下几类：

1. 在单处理器上的多个任务的调度问题。
2. 在多处理器上的多任务调度问题
3. 任务图在异构处理器上的调度问题
4. 作业车间调度问题
5. 周期性任务调度问题

·Page8：

1. 单处理器上的多任务调度问题，在该问题场景下，存在多个任务，给出多个任务的达到时间、每个任务的负载或截止完成时间，来决定每个任务的执行时间。

优化目标是：所有任务的平均完成时间最短或在截止时间前完成。

·Page9：

多处理器上的多任务调度问题。给出若干个任务的到达时间、每个任务的负载，决定什么时间、在哪个处理器上执行哪个任务。

优化目标是：最小化makespan，即优化从开始到结束的时间段，即完成一项工作所使用的时间

·Page10：

任务图在异构处理器上的调度任务。

给出每个任务在每个处理器上的执行的时间，任务在处理器之间进行传输需要消耗一定的传输时间、来决定每个任务什么时间、在哪个处理器上执行。

优化目标：最小化makespan、，即优化从开始到结束的 时间，即完成一项工作所使用的时间。

·Page11：

JSP问题：即车间调度问题。存在若干任务，给出每个任务在每个处理器上的处理时间，每个任务在每个处理器上都执行一次。

优化目标：最小化makespan

·Page12： 介绍完了当前的调度问题以及相关的分类后，下面对解决调度问题的调度算法进行介绍。

首先介绍一下调度算法的分类，1. 数学规划方法，例如整数规划、线性规划等数学方法

1. 进化算法，例如遗传算法、粒子群算法、蚁群算法等。
2. 简单完备的启发式算法。例如列表调度算法。
3. 基于机器学习的调度算法，例如强化学习。

Page13：当前调度算法从系统为中心考虑，可以从两方面考虑调度算法应该如何实现，

1. 一方面可以从设备角度出发。例如关注设备的状态，当设备空闲时调度过程触发。

对于每个可用的设备，依据某些规则来选择其要执行的任务。例如先来先服务、短作业优先、最早截止时间优先、等待时间最长优先等等。

1. 另一方面，调度算法可以从任务为中心考虑。例如当一个新任务到达时，触发对该任务的调度。对于每个任务，为该任务依照某些规则 分配设备资源。例如依照设备最早结束时间短的选择设备。

Page14：上面介绍了当前的调度算法的分类，其中提到了启发式算法，列表调度方法。

下面以该算法为例介绍一下该heuristic算法的流程和核心思想。

列表调度算法主要分为两个步骤：

1. 任务选择 2. 处理器/设备选择。

在介绍该算法时我们考虑任务图在异构处理器上进行处理的场景。

第一步，任务选择：对于每个任务分配优先级，并按照优先级进行排序，按照所得到的优先级排序后的顺序选择任务，优先级高的任务优先选择。

第二步，处理器选择:对于选中的任务分配处理器资源，在分配时最小化预先定义的花费函数。

重复以上两个步骤，直到所有任务都调度完成。

Page15：下面以一个实际的算例来说明一下列表调度算法。

首先看下图的任务图表示：首先图中共有10个task需要在处理器上进行调度，左图为10个task的任务图，其中有向图表示了task之间的依赖关系，每个任务需要等待上一个任务执行结束后才能执行。右边这张图是每个任务在每个处理器上的执行时间，依据该时间，可以计算出Upward RANK值，并依据该值对所有任务进行排序，得到排序后的序列，n1.n3...

其中节点上的红色值为UPWARD RANK值作为任务的优先级

Step2：处理器选择：最早完成时间，对于每个任务，选择其能最早完成的处理器进行分配。

Page16：如上图所示，任务的执行顺序为1，3，4，2，5.。。。。

对于任务1来说，其对应在每个处理器上执行时间为14，16，9，由于9最小所以任务1在处理器3上执行

对于任务3来说，需要有1的结果，则P1 = 11+9+12 = 32 P2 = 13+12+9 =34 P3 =9+19=28

对于任务4来说：P1 = 9+9+13=31 P2 =9+8+9=26 P3 = 28+17=45