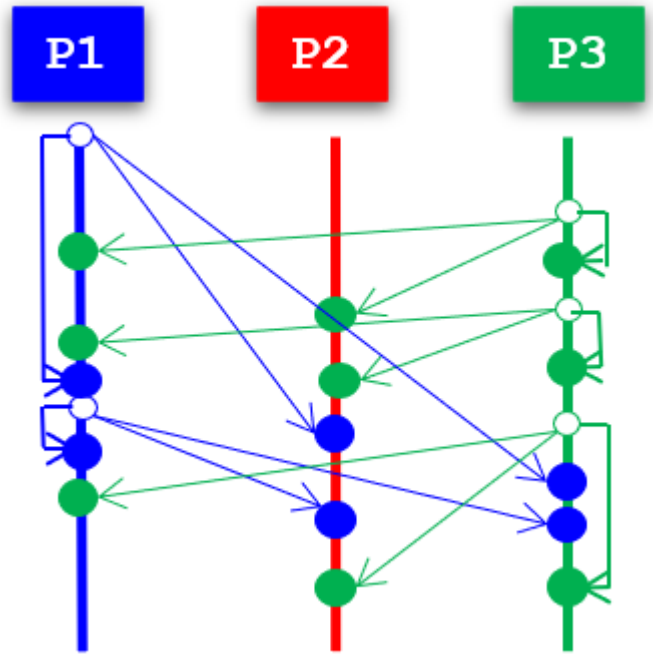
分布式理论作业

1. **Given the following operations of three processes (P1, P2, P3). They belong to（）**



(a) Total Ordering
(b) Sequential Ordering
(c) Causal Ordering
(d) Data-centric ordering

- (a) **全序（Total Ordering）**：全序是指对于系统中的所有消息，所有的进程或节点都按照相同的顺序接收它们。也就是说，如果进程A先于进程B接收了消息m1，那么所有的进程都必须先于进程B接收消息m1。全序是最严格的消息排序方式，它可以保证系统的一致性，但是也需要较高的开销。
- (b) **顺序（Sequential Ordering）**：顺序是指对于系统中的每个发送者，所有的进程或节点都按照发送者发送的顺序接收它们的消息。也就是说，如果发送者S先发送了消息m1，再发送了消息m2，那么所有的进程都必须先接收消息m1，再接收消息m2。顺序是一种较弱的消息排序方式，它可以保证发送者的因果关系，但是不保证不同发送者之间的因果关系。

- © **因果（Causal Ordering）**：因果是指对于系统中的所有消息，所有的进程或节点都按照它们的因果关系接收它们。也就是说，如果消息m1导致了消息m2的发送，那么所有的进程都必须先接收消息m1，再接收消息m2。因果是一种更弱的消息排序方式，它可以保证系统的因果一致性，但是不保证没有因果关系的消息的顺序。
- (d) **数据中心（Data-centric Ordering）**：数据中心是指对于系统中的每个数据对象，所有的进程或节点都按照相同的顺序接收对该数据对象的操作。也就是说，如果进程A先于进程B对数据对象D进行了操作o1，那么所有的进程都必须先于进程B对数据对象D进行操作o1。数据中心是一种特殊的消息排序方式，它可以保证数据对象的一致性，但是不保证不同数据对象之间的一致性。

由图上关系容易看出 A&B 是正确的；然后因果关系的话图上并没有表示一个消息的发出引起另外一个信息的发出；然后 D 的话因为本身就是全序的了，所以对数据对象的操作也肯定是顺序的；所以答案为：A B D

2. Explain in your own words what the main reason is for actually considering weakconsistency models.

弱一致性模型在分布式系统中被广泛考虑使用，主要原因有以下几点：

1. 性能优化：在分布式系统中，数据的读写操作往往需要在多个节点之间进行同步，如果采用强一致性模型，每次读写操作都需要等待所有节点达到一致，这会导致大量的网络延迟，降低系统的性能。而弱一致性模型则允许在一定程度上的数据不一致，可以减少网络同步的次数，实现高并发和高吞吐量，从而提高系统的性能。
2. 提高系统可用性：弱一致性模型可以提高系统的可用性。在强一致性模型中，如果某个节点发生故障，可能会导致整个系统无法进行读写操作。而在弱一致性模型中，即使某些节点发生故障或者网络延迟，其他节点仍然可以进行读写操作，不会影响到系统的可用性。
3. 支持系统的可扩展性：在大规模的分布式系统中，节点的数量可能会非常大，如果使用强一致性模型，那么每次数据更新都需要所有的节点进行通信，这在实际中是不可行的。而弱一致性模型则可以更好地处理这种情况，使得系统可以更容易地扩展。
4. 降低系统开发和维护复杂性：弱一致性模型不需要所有写操作立即传播到所有副本，这可以降低系统的复杂性，允许不同节点上有不同版本的数据，并且只要求在一个特定时间点内，所有节点上看到相同版本的数据即可。

然而，弱一致性模型也有其缺点，那就是可能会导致数据的不一致，这在一些需要高度一致性的应用中是无法接受的。因此，在选择是否使用弱一致性模型时，需要根据具体的应用场景和需求进行权衡。

弱一致性模型适用于那些对实时性要求较高，但对数据完整性要求较低的场景。例如，在电子商务、社交网络、在线游戏等领域，用户可能更关心能够及时地访问和交互数据，而不太在意数据是否有误差或冲突。但是，在金金融、医医疗、电力等领域，用户可能需要对数据有更高的信任度和准确度，因此不能接受弱一致性模型带来的风险

> ### 3. Why is computer clock synchronization necessary? Describe the design requirements for asystem to synchronize the clocks in a distributed system.

1. 准确性： 不同设备间的时钟保持同步，才能保证事件发生的顺序是准确的。例如，在金融交易中，如果不保证时间的统一，可能会导致错失交易机会或错误执行交易指令。

2. 一致性： 当多个设备需要协作完成同一任务时，时钟同步可以确保这些设备按照相同的顺序执行操作，避免产生竞争条件或数据冲突。例如，在数据库复制中，如果没有时间同步，可能会导致不同副本的数据不一致。

3. 可靠性： 许多协议和算法都依赖于准确的时间信息，例如分布式锁、选举算法等。如果没有时钟同步，这些协议和算法可能会失效，导致系统不可用。

4. 安全性： 一些安全机制也依赖于时间信息，例如数字签名和时间戳等。如果没有时钟同步，这些安全机制可能会被破解。

设计一个用于同步分布式系统中的时钟的系统，需要考虑以下几个要求：

1. 精度：时钟同步的精度应该尽可能高，以满足系统的需求。
2. 可靠性：时钟同步系统应该能够在各种条件下都能正常工作，包括网络延迟、节点故障等情况。
3. 可扩展性：随着系统规模的增长，时钟同步系统应该能够轻松地扩展到更多的节点。
4. 安全性：时钟同步系统应该能够防止恶意节点篡改时间，以保证系统的安全性。

5. 效率：系统应该能够高效地同步时钟，尽量减少同步过程的 overhead. 这可能需要使用轻量级的协议和算法。
6. 容错性：系统应对可能影响时钟同步的故障和网络延迟具备容错能力。它应具备检测和恢复时钟同步故障的机制，这些故障可能由节点故障、网络分区或通信延迟引起。
7. 准确性：同步的时钟应具有高度的准确性，以最小化节点之间的时间差异。系统需要考虑时钟漂移，它指的是时钟随时间略微偏离实际时间的倾向。可以使用时钟调整算法或外部时间参考（如 GPS）等技术来实现准确性。

实现这些要求的一种常见方法是使用网络时间协议（NTP），它是一个用于同步网络中计算机时钟的协议。

> **4. Given a certain merge sort algorithm, we assume that the sequential part accounts for 60%running time and the remaining parallel part 40% time. Moreover a 6-core computer canspeed up the parallel part by 4×, 10-core computer by 5×, and yet 20-core computer by 8×, could you approximate the sweet point of the merge sort algorithm?（根据给定的归并排序算法，我们假设顺序部分占总运行时间的60%，而剩余的并行部分占40%的时间。此外，一个6核的计算机可以将并行部分加速4倍，10核的计算机可以加速5倍，而20核的计算机可以加速8倍。您能否大致估计出归并排序算法的最佳性能点？）**

这个问题是关于合并排序算法的甜点问题。甜点问题是指在并行计算中，寻找最佳的处理器数量，使得并行部分的加速比最大。一般来说，可以使用**阿姆达尔定律**来估算甜点问题。阿姆达尔定律是指：

$$S_p = \min(N, \frac{1}{(1-P) + \frac{P}{N}} + \alpha N)$$

其中，Sp是并行部分的加速比，P是并行部分占总运行时间的比例，N是处理器的数量。根据这个公式，我们可以计算出不同处理器数量下的加速比，然后找出最大值对应的处理器数量，即为甜点。

在您的例子中，P=0.4，N分别为6，10，20。我们可以得到：

$$S_{p6} = \frac{1}{(1-0.4)+\frac{0.4}{6}} \approx 1.54$$

$$S_{p10} = \frac{1}{(1-0.4)+\frac{0.4}{10}} \approx 1.67$$

$$S_{p20} = \frac{1}{(1-0.4)+\frac{0.4}{20}} \approx 1.82$$

由此可见，Sp20是最大的，因此20核的计算机是甜点。但是，这个结果是基于理想情况的，没有考虑处理器之间的通信开销。如果通信开销不可忽略，那么甜点可能会变小，因为处理器数量越多，通信开销越大。为了考虑通信开销，我们可以使用**古斯塔夫森定律**来估算甜点问题。古斯塔夫森定律是指：

$$S_p = \min(N, \frac{1}{(1-P)+\frac{P}{N}+\alpha N})$$

其中，α是一个常数，表示**通信开销的影响因子**。根据这个公式，我们可以计算出不同处理器数量下的加速比，然后找出最大值对应的处理器数量，即为甜点。

在这个例子中，P=0.4，N分别为6，10，20，α的具体值需要根据实际情况确定，这里我们假设α=0.01。我们可以得到：

$$S_{p6} = \min(6, \frac{1}{(1-0.4)+\frac{0.4}{6}} + 0.01 \times 6) \approx 1.56$$

$$S_{p10} = \min(10, \frac{1}{(1-0.4)+\frac{0.4}{10}} + 0.01 \times 10) \approx 1.68$$

$$S_{p20} = \min(20, \frac{1}{(1-0.4)+\frac{0.4}{20}} + 0.01 \times 20) \approx 1.82$$

由此可见，Sp20仍然是最大的，因此20核的计算机仍然是甜点。但是，如果α的值增大，那么甜点可能会变小，因为通信开销的影响会更大。例如，如果α=0.05，那么我们可以得到：

$$S_{p6} = \min(6, \frac{1}{(1-0.4)+\frac{0.4}{6}} + 0.05 \times 6) \approx 1.64$$

$$S_{p10} = \min(10, \frac{1}{(1-0.4)+\frac{0.4}{10}} + 0.05 \times 10) \approx 1.75$$

$$S_{p20} = \min(20, \frac{1}{(1-0.4)+\frac{0.4}{20}} + 0.05 \times 20) \approx 1.82$$

这时，Sp10是最大的，因此10核的计算机是甜点。

综上所述，甜点问题的估算方法有两种，一种是使用阿姆达尔定律，一种是使用古斯塔夫森定律。阿姆达尔定律适用于理想情况，古斯塔夫森定律适用于考虑通信开销的情况。具体的甜点取决于并行部分占总运行时间的比例，处理器的数量，以及通信开销的影响因

子。

5. A client attempts to synchronize with a time server. It records the round-trip times and timestamps returned by the server in the table below.
Which of these times should it use to set its clock? To what time should it set it? Estimate the accuracy of the setting with respect to the server's clock. If it is known that the time between sending and receiving a message in the system concerned is at least 8 ms, do your answers change?（一个客户端尝试与时间服务器进行同步。它记录了服务器返回的往返时间和时间戳，如下表所示。它应该使用哪个时间来设置自己的时钟？应该将时钟设置为什么时间？估计相对于服务器时钟的设置精度。如果已知在该系统中发送和接收消息之间的时间至少为8毫秒，你的答案是否会改变？）

Round-trip (ms)	*Time (hr:min:sec)*
22	*10:54:23.674*
25	*10:54:25.450*
20	*10:54:28.342*

- 客户端可以选择最短的往返时间（round-trip time）作为最佳的同步时间，因为这样可以减少网络延迟的影响。在这个例子中，最短的往返时间是 20 毫秒，对应的服务器时间是 10:4:28.342。客户端可以将其时钟设置为该时间加上往返时间的一半，即 10:4:28.352。
- 客户端可以使用所有的往返时间和服务器时间的平均值来设置其时钟，因为这样可以减少随机误差的影响。在这个例子中，往返时间的平均值是 22.333 毫秒，服务器时间的平均值是 10:54:25.822。客户端可以将其时钟设置为服务器时间的平均值加上往返时间的一半，即10:54:25.833。
- 客户端可以使用最新的服务器时间来设置其时钟，因为这样可以保证时钟的实时性。在这个例子中，最新的服务器时间是 10:4:28.342。客户端可以将其时钟设置为该时间加上往返时间的一半，即10:4:28.352。

无论客户端使用哪种方法，其时钟的精度都受到往返时间的限制。如果往返时间是t，那么客户端的时钟和服务器的时钟之间的最大误差是 t/2 。在这个例子中，如果客户端选择最短的往返时间，那么其时钟的精度是 +-10 毫秒；如果客户端选择平均值或最新值，那么其时钟的精度是 11.167 毫秒。

如果已知发送和接收消息的时间至少是8毫秒，那么客户端的时钟的精度会有所提高，因为这样可以减少往返时间的不确定性。在这个例子中，如果客户端选择最短的往返时间，那么其时钟的精度是（20 - 8 * 2）/ 2 = +-2 毫秒；如果客户端选择平均值或最新值，那么其时钟的精度是 +-3.167 毫秒。