拜占庭问题及算法

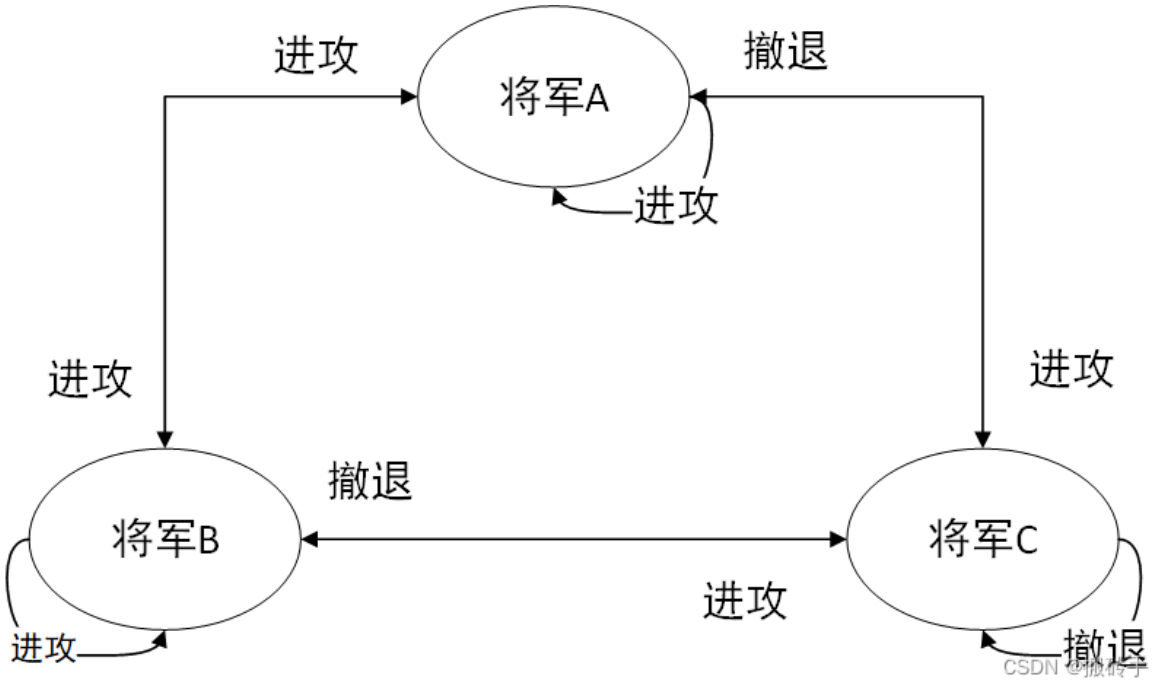
拜占庭问题（Byzantine Problem）又叫拜占庭将军（Byzantine Generals Problem）问题，讨论的是允许存在少数节点作恶（消息可能被伪造）场景下的如何达成共识问题。

拜占庭问题最早是用来解释异步系统中共识问题的一个虚构模型。拜占庭是古代东罗马帝国的首都，由于地域宽广，守卫边境的多个将军（系统中的多个节点）需要通过信使来传递消息，达成某些一致决定。但由于将军中可能存在叛徒（系统中节点出错），这些叛徒将向不同的将军发送不同的消息，试图干扰共识的达成。这种情况十分类似于分布式系统中多个节点达成共识的问题。

拜占庭问题即讨论在此情况下，如何让忠诚的将军们能达成行动的一致。

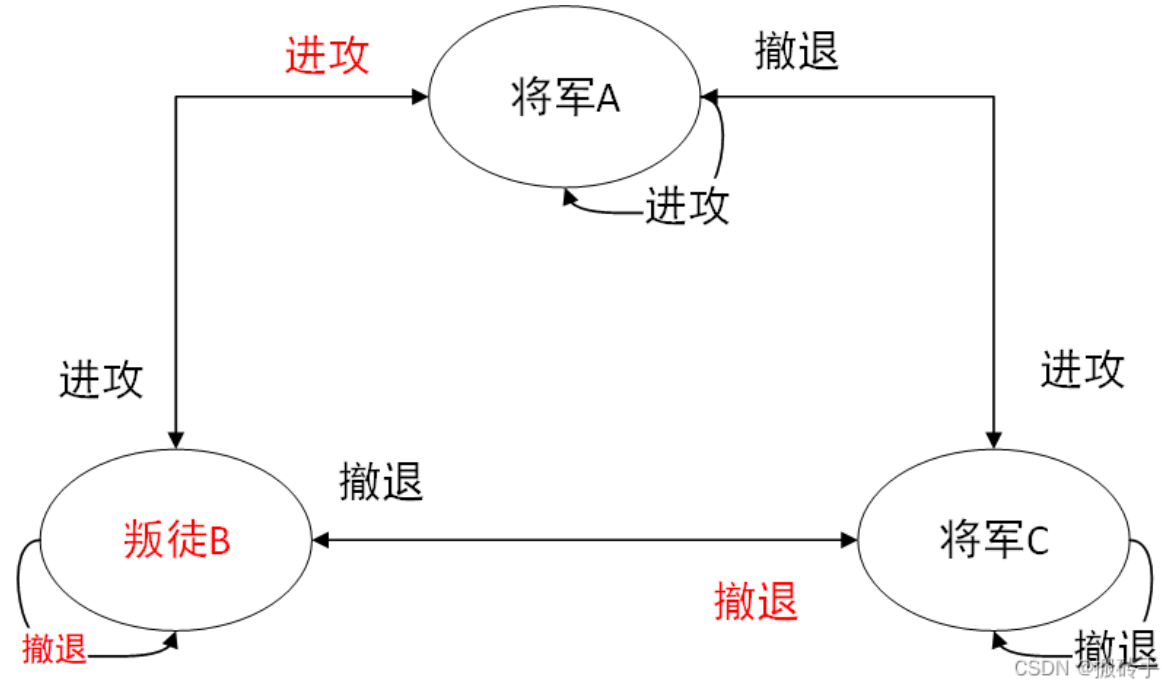
典例-三将军模型

拜占庭有n个将军，他们都可以独立的作出决策，选择进攻和撤退，彼此之间通过信使传递消息。对于一场战争，所有的将军必须作出共同的作战决策并依此执行，只有半数以上的将军同时进攻才能取得战斗的胜利，作战计划的制定遵循“少数服从多数原则”。为了简化问题，我们采用3个将军进行说明。
假设现有三个将军A、B、C，他们都是忠诚于拜占庭的将军，对于一场战斗，他们三人将讨论出一个共识的作战计划，选择进攻或者撤退，并忠诚的执行作战指令。他们首先分别根据自己的判断选择进攻或者撤退，然后将自己的决策发送给其他两个将军，其他两个将军也将他们的作战计划传递给该名将军，然后这名将军根据少数服从多数的原则，执行大多数人选择的作战方案。
假设A、B两名将军选择进攻，C将军选择撤退。那么，
A将军通过信使告知B、C选择进攻；
B将军通过信使告知A、C选择进攻；
C将军通过信使告知A、B选择撤退；



对于每个将军而言，收到的消息中进攻与撤退的比例均为2：1，因此每个将军根据少数服从多数的原则，均执行进攻指令，实现了共同作战的目标。
但是，拜占庭地域辽阔，将军中可能存在叛徒。叛徒并不一定会如实的发送相同的指令，就会导致将军之间的一致性遭到破坏，无法达成共识作战目标。
下面假设叛徒为B，叛徒B将告知A和C不同的作战信息。
A将军通过信使告知B、C选择进攻；

C将军通过信使告知A、B选择撤退；
叛徒B通过信使告知A选择进攻，但告知将军C选择撤退；



在这种情况下，A收到的指令中进攻与撤退的比例为2:1，A将军将选择进攻.C将军收到的指令中，进攻与撤退的比例为1:2，C将军选择了撤退。而叛徒B理所当然会选择撤退。于是我们发现，只有将军A发动了进攻，无法取得战争的胜利，将军之间的共识也被打破，产生了结果的不一致性。

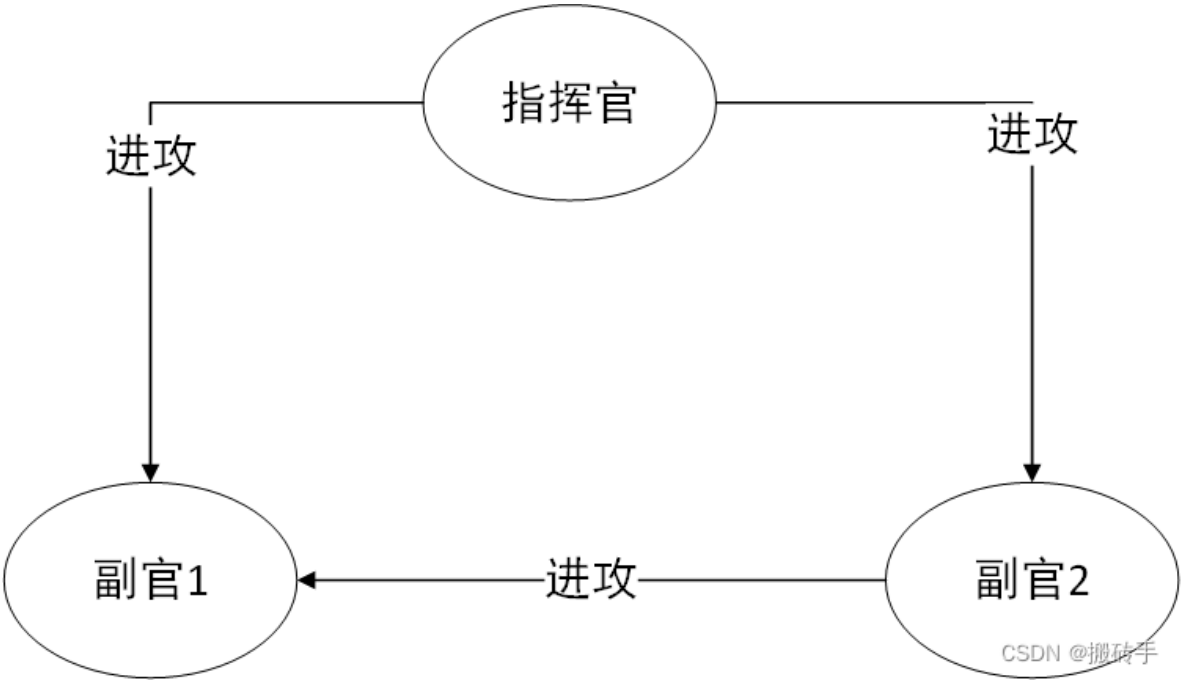
指挥官-副官模型

将拜占庭问题简化一下描述即为：有一个指挥官发送一个命令给他的n-1个副官们，而且在这个问题中要有两个基本条件：

条件一：所有忠实的副官遵从相同的指令；

条件二：如果指挥官是忠实的，那么所有副官都将执行他的命令。

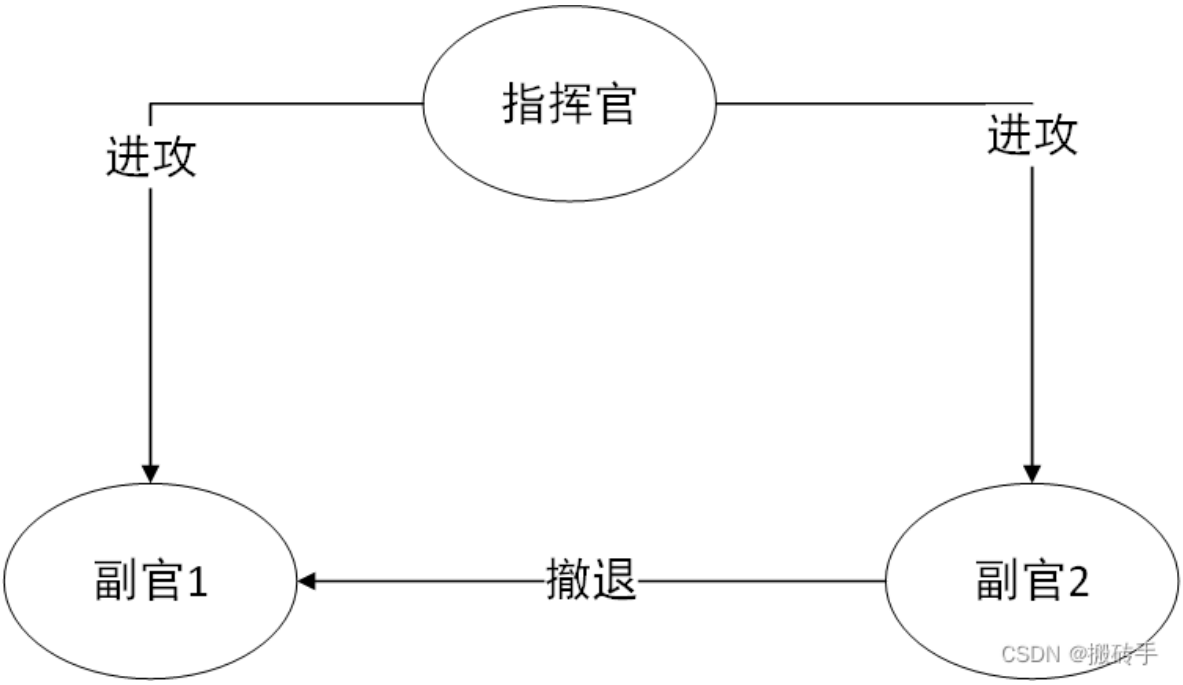
下面就用实际的例子讨论一下拜占庭问题



指挥官和副官都是忠诚的

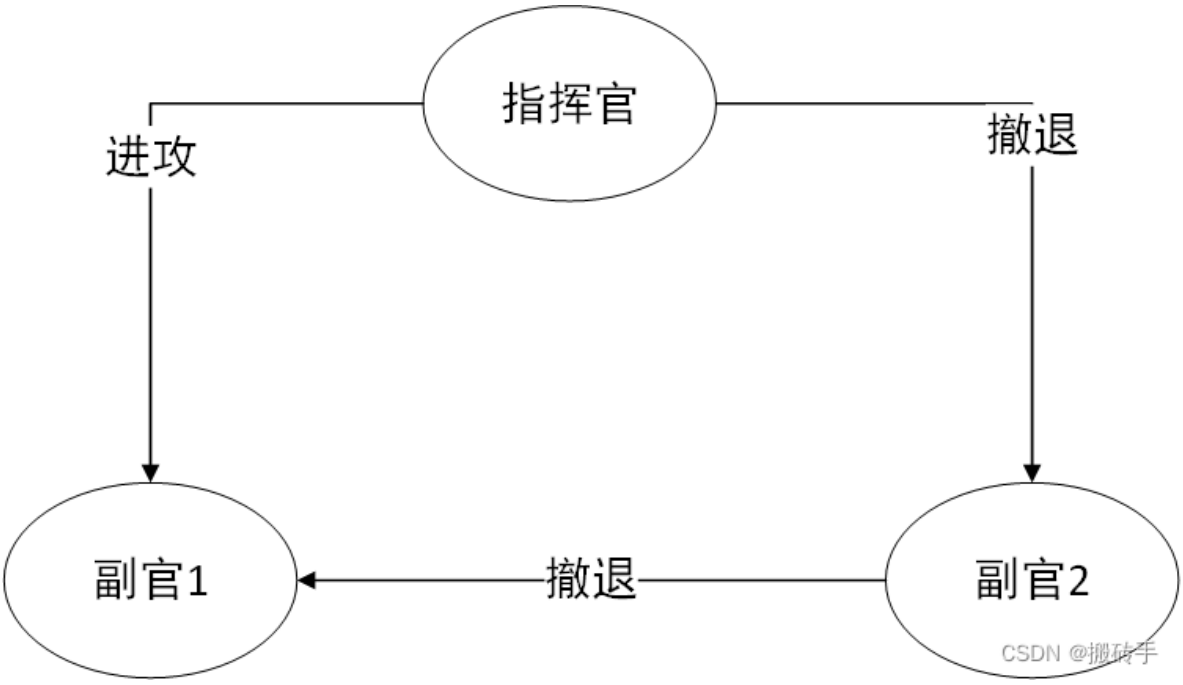
指挥官分别向两个副官发出进攻命令。副官1和副官2都是忠诚的，副官2向副官1说：“指挥官下达了进攻的命令”。副官1收到的副官2的消息和指挥官的消息是相同的，这时就达成了共识，共同发起进攻。

副官2是叛徒



指挥官会分别向两个副官发出进攻命令。副官2是叛徒，他欺骗副官1说：“指挥官下达了撤退命令”。副官1收到了两个不同的命令，判定指挥官与副官2中至少有一个叛徒。他无法判断谁是叛徒，因此也无法正确行动。

指挥官是叛徒



指挥官向两个副官下达了不同的命令。副官1收到的是不同的命令，那么指挥官和副官2中有叛徒，但由于不能确定谁是叛徒，无法采取正确的行动。

结论
当三人中有一个是叛徒时，无法达成一致的行动，即拜占庭问题无解，且无法判断叛徒是谁。
所以必须要满足问题中存在的节点必须是4个及以上。

问题的解决

PBFT（Practical Byzantine Fault Tolerance）实用拜占庭容错

PBFT算法的核心理论是N>=3F+1，其中N是总节点数，F是故障节点（故障包括叛徒或不响应）。假如有F个故障节点，至少有3F+1个节点才能保证整个系统正确运行。

算法要求至少要4个参与者ABCD，1个为总司令，3个师长

没有叛徒：

1. 4个将军分别为ABCD，选举A为总司令，总统（终端用户）向A说进攻
2. A对BCD说进攻
3. BCD都各自对其他节点说进攻
4. 每个节点都收到其他人说进攻的指令，则进攻

假如出现1个叛徒D：

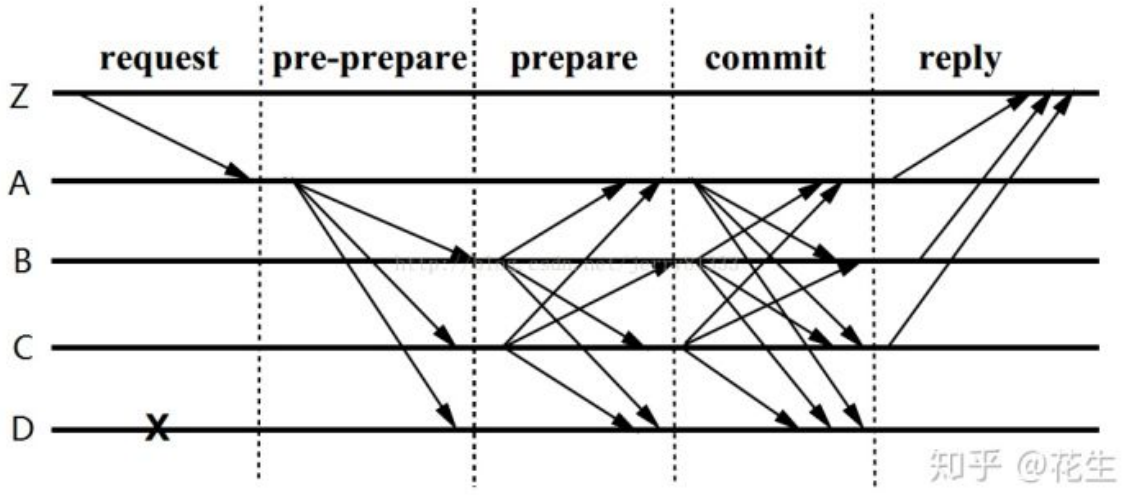
1. A对BCD说进攻
2. D对BC说撤退
3. 对B而言，A和C说进攻（2票）大于D说撤退（1票），进攻

假如出现2个叛徒CD：

1. A对BCD说进攻
2. CD对B说撤退，撤退
3. A又进攻，一致性失败

所以当有4个节点时，只能容许1个节点出问题，该问题可能是叛徒（伪造消息）或者没有消息（无响应）

实际上PBFT会更复杂一些，分为预准备（pre-prepare）、准备(prepare)和确认(commit)



1. request：总统Z向A发消息
2. pre-prepare：服务端A收到Z的请求后进行广播，扩散至ABC
3. prepare：BCD,收到后记录并再次广播，B->ACD，C->ABD，D因为宕机无法广播
4. commit:ABCD节点在Prepare阶段，若收到2F（F为可容忍错误个数）个或以上和自己相同的请求，则进入Commit阶段，广播Commit请求
5. reply：ABCD节点在Commit阶段，若收到2F+1个的相同请求，则对Z进行反馈

加入出现1个伪装节点D，对于B号节点来说，有A、C号节点和自己相同，大于等于2F个即2个，则进入commit阶段，并告知其他节点；同理对于A号节点和C号节点也会进入commit阶段。最终有ABC三个节点进入commit阶段，3个请求满足2F+1个即3个，它们会对Z进行反馈。

- 为什么是3F+1

容错，就是不需要等待的数量，当有4个节点，只需要有3个节点反馈数据就立即决策了，那么3个有可能是一个是有伪装的，这种情况下也能保证系统一致性（节点不能无限制的等待所有其他节点都发数据过来，他只需要等待N-F即2F+1个反馈）。

1. 如果未接到反馈的恰好是叛徒节点，剩下的全都保持一致，没有问题。
2. 如果未反馈的是好节点，叛徒节点只有1票，则2个好节点>1个叛徒，系统正常。
3. 3F+1。系统只等N-F个就做决策。极端情况下，所有的F个叛徒都先发数据，那么剩下的必须必须是F+1个正确的节点才能保证 F+1个叛徒 > F个叛徒，那么所有N的节点为（F+1个收到的节点 + F个未收到节点 + F个叛徒）也就是3F+1个。

4 比特币POW（Proof Of Word）工作量证明

比特币通过工作量证明来解决一致性问题，通过数学的方式，让其中一个节点成为当前情况下的总司令，后续所有的命令都是根据该结果来进行。

举例来到将军问题，先假定某个数序规则Puzzle需要将军的军师平均花一天即24小时的时间才能解开。每次算出来的将军可以获得总司令给的军械奖励。

1. 任何一个将军都可以说6天后进攻或撤退，然后根据‘6天后进攻’或者‘6天后撤退’内容按照Puzzle规则进行计算。
2. 将军A花了23个小时率先发现‘天行健’和‘6天后进攻’符合Puzzle规则，于是他送信给其他将军。
3. 其他将军收到该信息之后，下一步‘攻击东城门’或者‘攻击西城门’，根据‘6天后进攻&天行健&A#攻击东城门’或‘6天后进攻&天行健&A#攻击西城门’作为内容按照Puzzle规则进行计算。
4. 将军B花了25个小时率先发现‘天圆地方’和‘6天后进攻&天行健&A#攻击东城门’符合Puzzle规则，于是他送信给其他将军。
5. 其他将军收到该信息之后，下一步‘使用云梯’或者‘使用攻城锤’，根据‘6天后进攻&天行健&A#攻击东城门&天圆地方&B#使用云梯’或‘6天后进攻&天行健#攻击东城门&天圆地方&B#使用攻城锤’作为内容按照Puzzle规则进行计算。
6. 一直重复以上4、5。

为了获得奖励，他们一般都是谁先到这里就根据谁的算。并且谁的指令最长最有可能被司令认可获得奖励军械，这样，所有的将军都会在第六天进行进攻。

其中的问题：

1. 在步骤2将军Z也花了23个小时发现‘自强不息’和‘6天后撤退’符合Puzzle规则，也送信给其他将军。那么其他将军一部分根据将军A的结果往下算，一部分根据将军Z的结果往下算，在第二天谁最先解开Puzzle那么极有机会率先通知其他将军。结果将军B率先发现第二个Puzzle，并通知了70%的将军的时候，这70%的将军解开第三个Puzzle的概率远远大于剩下的30%的将军解开第三个Puzzle。到第四个Puzzle的时候就可能变成90%：10%，第五个Puzzle的时候变成98%：3%，第六个Puzzle的时候变成99.99%：0.01%。这样差不多所有将军都会进行进攻。（即为比特币的6次确认可以判定交易真实）
2. 也可能将军Z胜出，只不过上面的比例反过来，那么内容变成‘6天后撤退&人之初&Z#向东撤退5公里&赵钱孙&Y#扎营做饭&弟子规&X’等。但是并不影响第6天他们集体进攻还是撤退。