

## 分布式死锁检测算法

原显智 17888828056

分布式的死锁检测算法可以分成中心化(centralized)和分布式(distributed)两类。由于后一类过于复杂,本文讨论仅限于中心化的死锁检测算法。

中心化算法的基本思想是,在各个节点构建局部等待图(wait for graph),并在某个节点根据多个局部等待图构建全局等待图。这样,全局视角下的死锁得以被检测。

中心化算法可进一步划分成两类。一,将所有局部等待图汇总到同一个中心节点,并构建全局等待图。第二,按不同层次(hierarchy)构建等待图。举例来说,在城市层次构建的等待图可以汇总为所在省的等待图,这样,在该层次发生的死锁在当前层次即可被检测到,无需等到在更高的国家甚至全球层次的等待图中被发现。如果有理由假设绝大多数死锁在较低层次上发生(如同一个省),而非更高层次(如同一个国家),该方法可相对第一种方法更快检测死锁。

中心化算法可能存在两个问题。第一,由于网络延迟,构建出来的全局等待图和实际上的等待关系并不相同,导致不存在死锁的情况下误认为存在死锁,引发不必要的回滚。此外,即便死锁真的发生,在死锁处理执行之前,涉及到死锁中的某个事务可能因为其他原因回滚,打破死锁,此时没有死锁处理的必要。但死锁处理算法仍旧运行,引发不必要的回滚。

当然,如果说多个节点难以分享等待图,例如各个节点运行着不同的数据库软件,可设定每个事务等待时间的界限,超过则回滚等待中的事务。

参考文献:

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). Database System Concepts. P844-847

Ramakrishnan, R., & Gehrke, J. (2003). Database management systems. P625-627

