

分布式操作系统

初始化

扩散算法

24

50+9/14

64-65

分布式操作系统初始化的扩散算法

The Diffusion Algorithm of Initialization in the Distributed Operating System

胡金柱 徐松

TP316

TP393.1

(华中师范大学计算机科学系 武汉 430079)

Abstract In this paper, we discussed the problem in initialization processes of the distributed operating system. Raised and analysed the diffusion algorithm of initialization in distributed operating systems.

Keywords Distributed operating system, Initialization, Diffusion algorithm

1. 前言

分布式操作系统的初始化过程本身也是分布式的,各节点可能在不同的时间内异步初启,系统完成初启达到稳定后还可能新的节点请求加入,也可能在运行过程中有节点因为故障等因素而退出。分布式操作系统的微内核将是实现这个分布式过程的核心。它的一个重要功能就是在系统初始化时建立各个节点的连接共同构成分布式环境。

如果分布式系统中的各个节点都处于同一总线式的局域网中,那么问题可能会变得简单,因为只要不考虑网络拥塞,一个简单的广播就会宣告本节点的存在和能力。如果不是一个总线型拓扑的网络结构,又该如何止各个节点相互感知并构成逻辑上统一的分布式操作系统呢?下面将讨论系统初启时的扩散算法和仲裁节点的建立算法。

2. 扩散算法

采用如下扩散算法,可以保证每个微内核首先得到本节点和其它节点间的通讯代价。

2.1 算法的描述

设:①N为节点数,M为链路数;②L为一个大小为M的链路表,其中L[p].m为链路的度量值,L[p].s为链路的源站点,L[p].d为链路的的目的站点;③D为一个[N,N]的矩阵,其中D[i,j]为从节点i到节点j的距离,则具体算法如下:

- (1)初始化所有的D[i,j],若i=j,则D[i,j]=0,否则,D[i,j]为无穷大;
- (2)对于所有链路p,和所有目的节点k,令i=L[p].s,j=L[p].d,计算出D=L[p].m+D[j,k];
- (3)若d小于D[i,k],则更新D[i,k]=d;
- (4)若至少有一个D[i,k]被更新,则转到(2),否则算法结束。

该算法在小于或等于N步之后收敛,其整体时间复杂度不超过O(M * N²)。在分布式系统中,每个节点只关心代表距离中的纵列D[i,*],纵列D[i,*]实际上就是距离向量,在更新后将传送给相邻的节点。

2.2 算法的分布式执行过程

当系统启动时,假设有5个节点工作站参与初启活动,如图1所示。

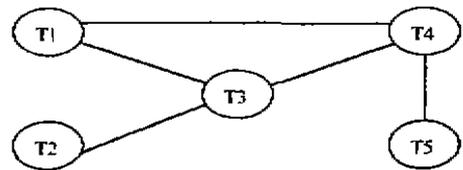


图1 具有5个节点的工作站模型

假设图中每个节点都只了解并记忆本地特性(Local Preferences),而不关心整体网络的布局。它们并不知道另外有多少节点,甚至不知道与之直接相连的另一端是谁。此时,每个节点的D[i,*]数组中有效项个数为1,对于节点T1的通讯开销表(以下简称开销表)为:

起始节点	终止节点	开销	链路说明
T1	T1	0	本地

节点T1以距离向量的形式对该表进行抽象处理,生成一项:T1=0,它将向它的所有的相邻的节点发送该距离向量。以T3为例,在收到T1发来的消息之前,T3的开销表为:

起始节点	终止节点	开销	链路说明
T3	T3	0	本地

在 T3 节点收到 T1 发来的距离向量 $T1=0$ 后,它就更新全部距离.方法是在原有数值上加入本地链路的开销值,假定所有的本地链路的开销都为 1,并将消息内容转化为 $T1=1$.接着,再将距离向量与本地开销表比较,发现 T1 是未知项,于是在该表中增加新的项,如下表:

起始节点	终止节点	开销	链路说明
T3	T3	0	本地
T3	T1	1	2

这时 T3 把这个新增后的距离向量通知本地链路 2、3 和 4: $T3=0, T1=1$.在此期间, T4 也收到了来自 T1 的消息,并将自己的开销表更新为:

起始节点	终止节点	开销	链路说明
T4	T4	0	本地
T4	T1	1	1

T4 同样也要在本地链路 1、4 和 5 上发送更新后的距离向量 $T4=0, T1=1$,不妨设 T3 先收到 T4 的消息,然后 T4 才收到 T3 的消息,当 T3 收到 T4 发来的 $T4=0, T1=1$ 后,往开销表中增加一项 $T4=1, T1$ 一项者保持不变,如下表:

起始节点	终止节点	开销	链路说明
T3	T3	0	本地
T3	T1	1	2
T3	T4	1	4

T4 收到 T3 发来的 $T3=0, T1=1, T4=1$ 的消息后,更新开销表为:

起始节点	终止节点	开销	链路说明
T4	T4	0	本地
T4	T1	1	1
T4	T3	1	4

同理,每次开销表发生变化后,节点都要在本地链路上向相邻节点发送距离向量.直到所有的节点中开销表不再变化,结果如下:

起始节点	终止节点	开销	链路说明
T1	T1	0	本地
T1	T2	2	2
T1	T3	1	4
T1	T4	1	1
T1	T5	2	1

起始节点	终止节点	开销	链路说明
T2	T2	0	本地
T2	T1	2	3
T2	T3	1	3
T2	T4	2	3
T2	T5	3	3

起始节点	终止节点	开销	链路说明
T3	T3	0	本地
T3	T1	1	2
T3	T2	1	3
T3	T4	1	4
T3	T5	2	4

起始节点	终止节点	开销	链路说明
T4	T4	0	本地
T4	T1	1	1
T4	T2	2	4
T4	T3	1	4
T4	T5	1	5

起始节点	终止节点	开销	链路说明
T5	T5	0	本地
T5	T1	2	5
T5	T2	3	5
T5	T3	2	5
T5	T4	1	5

至此算法收敛.最后一次 T1、T2 向相邻节点发送距离向量,但是不会再导致相邻节点的开销表的变化.就这样,通过分布式计算,节点已经获得了从自己出发到达其它节点的开销信息.每个节点自身记忆的开销表实际就是上面矩阵 D 中的 $D[i, *]$ 列.

结论 此算法可以保证在一个初启的系统内部形成一个相关的节点工作站群,群落中的每一个节点都能识别其它节点的存在并能感知距离(通讯的代价).由于该算法类似微粒的扩散过程,因此我们称之为“扩散算法”.

参考文献

- 1 Tanenbaum A S. Distributed Operating Systems. Prentice Hall, 1995. 197~212
- 2 彭德纯,等.分布式并行处理技术导论.武汉大学出版社, 1995. 132~143
- 3 何炎祥,夏循斌.一种新型分布式操作系统自动生成系统模型.计算机研究与发展, 1994, 31(5)