

名字空间的分布式实现直接影响着名字解析的实现过程。在这里假定名字服务器没有多个副本，并且不使用客户方的缓存技术。当一个客户要解析一个名字时，它先把名字提交给本地的名字解析程序，假定要求被解析的名字为：

root:<nl, vu, cs, ftp, pub, global, index.txt >

以后在名字前加#代表解析该名字所获得的地址。这个路径名如果用 URL 表示的话，其形式为：ftp://ftp.cs.vu.nl/pub/global/index.txt，有两种方法对此名字进行解析：迭代式名字解析 (iterative name resolution) 和递归式名字解析 (recursive name resolution)。迭代式名字解析过程如图 1 所示。

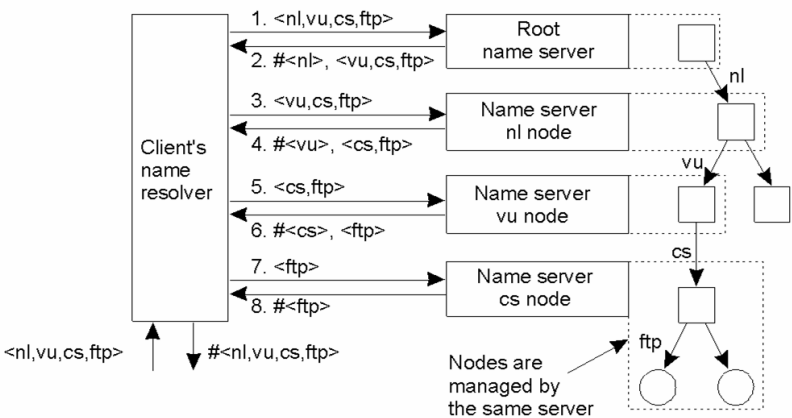


图 1 迭代式名字解析过程

在迭代式名字解析方案中，名字解析程序首先将整个路径名提交给根名字服务器，根名字服务器尽其所能解析它所能解析的部分并将结果返回给名字解析程序，在此例子中，根名字服务器只能对符号 nl 进行解析，并将 nl 所对应的名字服务器的地址返回给名字解析程序。

第二步，本地名字解析程序将路径名的剩余部分 nl:<vu, cs, ftp, pub, global, index.txt>提交给 nl 所对应的名字服务器，此名字服务器只能对符号 vu 进行解析，并将 vu 所对应的名字服务器的地址返回给名字解析程序。

第三步，本地名字解析程序将路径名的剩余部分 vu:<cs, ftp, pub, global, index.txt>提交给 vs 所对应的名字服务器，此名字服务器只能对符号 cs 进行解析，并将 cs 所对应的名字服务器的地址返回给名字解析程序。

第四步，本地名字解析程序将路径名的剩余部分 cs:<ftp, pub, global, index.txt>提交给 vu 所对应的名字服务器，此名字服务器只能对符号 ftp 进行解析，并将 ftp 所对应的名字服务器的地址返回给名字解析程序，再提交给客户程序。

最后，客户程序将路径名的剩余部分 ftp:<pub, global, index.txt>提交给 ftp 服务器，要求传输对应的文件。

与迭代式名字解析方案所不同的是，在迭代式名字解析方案中，名字服务器将中间结果（或部分结果）返回给客户方名字解析程序，而在递归式名字解析中，名字服务器将本身不能解析的部分提交给其他的名字服务器进行解析，而返回给客户方名字解析程序的是全部解

析的最终结果。在上述例子中，根节点名字服务器将不能解析的部分

nl:<vu, cs, ftp, pub, global, index.txt>

提交给 n1 对应的名字服务器继续解析，而根节点名字服务器返回给客户方名字解析程序的是全部解析的最终结果。

递归式名字解析过程如图 2 所示。同迭代式名字解析一样，最后客户程序将路径名的剩余部分 ftp:<pub, global, index.txt>提交给 ftp 服务器，要求传输对应的文件。

递归式名字解析的主要缺点是要求每个名字服务器具有较高的性能。递归式名字解析要求名字服务器完整地解析它所得到的整个路径名，尽管整个解析过程是可能同其他的名字服务器协作完成的。特别是对于全局层的名字服务器来说，情况更为严重。

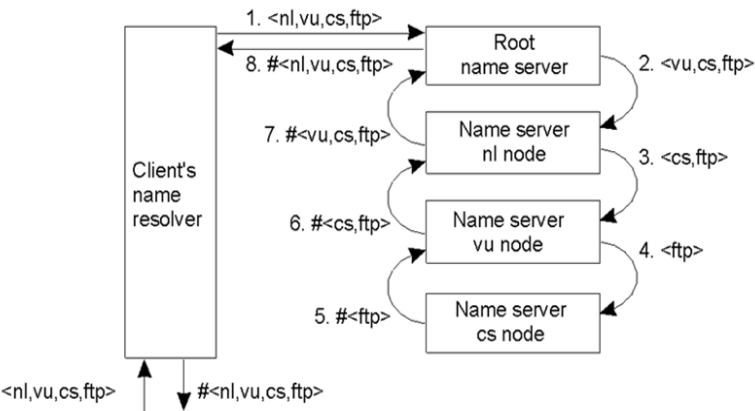


图 2 递归式名字解析过程

递归式名字解析有两个主要优点。第一个优点是如果采用缓存，那么递归式名字解析的缓存效果同迭代式名字解析的缓存效果相比更为有效；第二个优点是可以减少通信代价。

在缓存方面，使用递归式名字解析时，上层节点的名字服务器可以缓存到下层名字服务器所管理的部分内容，即使以后同样的请求是由不同客户发出的，也能立即得到解析的结果，所以缓存对提高性能是非常有效的。表 1 表示了一个客户要求对名字 root:<nl, vu, cs, ftp>进行解析，解析完成后相关名字服务器的缓存内容。

表 1 对名字 root:<nl, vu, cs, ftp>解析完成后相关名字服务器的缓存内容

对应节点的名字服务器	需要解析的部分路径	需要查询的地址	要求其他名字服务器完成的部分	收到并缓存的信息	以后可直接返回给请求者的信息
cs	<ftp>	#<ftp>	--	--	#<ftp>
vu	<cs, ftp>	#<cs>	<ftp>	#<ftp>	#<cs> #<cs, ftp>
ni	<vu, cs, ftp>	#<vu>	<cs, ftp>	#<cs> #<cs, ftp>	#<vu> #<vu, cs> #<vu, cs, ftp>
root	<ni, vu, cs, ftp>	#<nl>	<vu, cs, ftp>	#<vu> #<vu, cs> #<vu, cs, ftp>	#<nl> #<nl, vu> #<nl, vu, cs> #<nl, vu, cs, ftp>

如果使用迭代式名字解析，缓存技术只需要在客户方的名字解析程序中实现，一个客户方的缓存内容自然不会提高其他客户的名字解析效率。

从图 1 和图 2 可以得知，使用递归式名字解析所需的通信代价可能比迭代式名字解析要低，因为根名字服务器同客户的距离更有可能比两个父子节点的名字服务器的距离要远得多。