

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 21/22 (2006.01)

H04L 29/06 (2006.01)



# [12] 发明专利说明书

专利号 ZL 200610063004.X

[45] 授权公告日 2009年1月7日

[11] 授权公告号 CN 100449562C

[22] 申请日 2006.9.30

[21] 申请号 200610063004.X

[73] 专利权人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

[72] 发明人 吕涛 胡云

[56] 参考文献

WO01/074138A3 2001.10.11

JP2002-6972A 2002.1.11

CN1307289A 2001.8.8

US5579222A 1996.11.26

JP2004-54659A 2004.2.19

审查员 冯婷霆

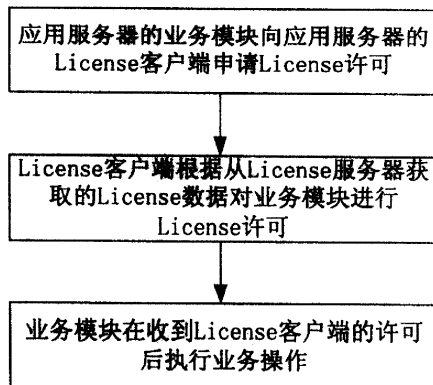
权利要求书 3 页 说明书 8 页 附图 4 页

[54] 发明名称

一种 License 控制方法及装置

[57] 摘要

本发明公开一种 License 控制方法，该方法包括步骤：A：应用服务器的业务模块向应用服务器的 License 客户端申请 License 许可；B：License 客户端根据从 License 服务器获得的 License 全局数据对业务模块进行 License 许可，所述 License 全局数据包括：License 服务器所有 License 数量、正在使用的 License 数量和剩余可用的 License 数量；C：业务模块获得 License 客户端的许可后执行业务操作。本发明利用各应用服务器的 License 客户端提供 License 许可，从而提高了 License 许可的实时性，降低了系统的负荷，增强了 License 许可的可靠性。



1、一种 License 控制方法，其特征在于，包括步骤：

A: 应用服务器的业务模块向应用服务器的 License 客户端申请 License 许可；

B: License 客户端根据从 License 服务器获得的 License 全局数据对业务模块进行 License 许可，所述 License 全局数据包括：License 服务器所有 License 数量、正在使用的 License 数量和剩余可用的 License 数量；

C: 业务模块获得 License 客户端的许可后执行业务操作。

2、根据权利要求 1 所述的方法，其特征在于，在执行步骤 A 之前还包括 License 服务器的数据初始化步骤：

License 服务器读取 License 总数，获取所有 License 客户端使用的 License 数量；

License 客户端暂停向业务模块提供许可，并向 License 服务器返回使用的 License 数量；

License 服务器汇总各 License 客户端使用的 License 数量，更新 License 全局数据，并向所有的 License 客户端发送更新后的 License 数据；

License 客户端根据所述更新后的 License 数据，更新本地可用的 License 数据，并恢复向业务模块提供许可。

3、根据权利要求 1 所述的方法，其特征在于，在执行步骤 A 之前还包括应用服务器的数据初始化步骤：

License 客户端清零本地使用的 License 数量并向 License 服务器发送消息；

License 服务器将所述 License 客户端使用的 License 数量清零，更新 License 全局数据，并向所有的 License 客户端发送更新后的 License 数据；

License 客户端更新本地 License 数据，并根据更新后的数据向业务模块提供许可。

4、根据权利要求 1 至 3 任一项所述的方法，其特征在于，所述步骤 B 中 License 客户端从 License 服务器获取 License 数据的步骤具体为：

License 客户端定时向 License 服务器发送 License 增量数据，License 服务器根据所述 License 增量数据更新 License 全局数据，并向所有的 License 客户端发送更新后的 License 数据，License 客户端更新本地的 License 数据；

或者，

License 客户端定时向 License 服务器发送其累计使用的 License 数量，License 服务器根据所述累计使用的 License 数量更新 License 全局数据，并向所有的 License 客户端发送更新后的 License 数据，License 客户端更新本地的 License 数据。

5、根据权利要求 4 所述的方法，其特征在于，所述增量数据指单个 License 客户端在一个更新周期内新占用以及新释放的 License 数量。

6、根据权利要求 1 至 3 任一项所述的方法，其特征在于，所述 License 许可包括：

判断是否可以申请新 License；或者

判断是否可以释放 License；或者

判断是否可以执行业务操作。

7、根据权利要求 1 至 3 任一项所述的方法，其特征在于，所有 License 客户端签发的 License 总数超过 License 服务器总的 License 数量。

8、一种 License 控制装置，其特征在于，所述装置包括：

业务模块，向 License 客户端申请 License 许可，并在获得许可后执行业务操作；

License 客户端，根据从 License 服务器获取的 License 全局数据对业务模块进行 License 许可，所述 License 全局数据包括：License 服务器所有 License 数量、正在使用的 License 数量和剩余可用的 License 数量。

9、根据权利要求 8 所述的装置，其特征在于，License 客户端定时向 License

服务器发送 License 增量数据, 并根据 License 服务器响应更新本地的 License 数据; 或者, License 客户端定时向 License 服务器发送其累计使用的 License 数量, 并根据 License 服务器响应更新本地的 License 数据。

10、根据权利要求 9 所述的装置, 其特征在于, 所述增量数据指单个 License 客户端在一个更新周期内新占用以及新释放的 License 数量。

11、根据权利要求 8 至 10 任一项所述的装置, 其特征在于, 所有 License 客户端签发的 License 总数超过 License 服务器总的 License 数量。

## 一种 License 控制方法及装置

### 技术领域

本发明涉及通信领域，尤其涉及 License 控制方法及装置。

### 背景技术

随着个人电脑的普及、大批程序员和软件公司的出现，软件技术得到长足发展，与此同时，软件保护受到了很大的挑战。

以前用户只要拥有了软件就能够运行并使用它，因此软件公司主要通过控制软件复制和软件分发来控制软件的使用。这时出现了许多防止软件被非法复制技术手段，其基本原理通常是将软件和特定的硬件载体捆绑起来，例如软盘、光盘等，但是这种方式过于依赖于硬件从而给软件的推广带来局限性。

随着软件使用量的急剧增长以及软件规模越来越庞大，软件的传播开始脱离硬件载体，此时软件开始能够进行本地化的安装、运行许可，软件公司通常采用先进的加密技术来保护软件的使用权，不同的软件副本需要不同的密码来安装、运行，此时的 License 相当于一把钥匙，用来开启对应的软件。

随着网络技术的发展，软件系统架构变得越来越复杂，分布式应用的出现使得本地化许可失去了效用，许多软件转而通过服务器来进行安装、运行许可，其组网示例如图 1 所示。部署专门的 License 服务器，管理用户所有的 License 许可，业务模块分布在不同的应用服务器上，每个业务模块直接访问 License 服务器并申请 License 资源，License 服务器动态的判断剩余的 License 并返回 License 许可结果，业务模块根据 License 许可结果决定业务流程。

其方法流程示例如图 2 所示：

S1:业务模块收到业务请求，向 License 服务器申请 License 资源；

**S2:License 服务器检查当前剩余的 License 资源，判断是否发放 License 资源；**

**S3:如果还有可以发放的 License 资源， License 服务器进行 License 许可，将 License 资源扣除 1，并向业务模块返回申请成功的响应消息；**

**S4:业务模块收到 License 许可后，执行业务操作。**

本方案将用户所有的 License 集中交给 License 服务器管理，因此可以实现 License 的动态发放，以及提高 License 使用率。

但是本方案完全依赖 License 服务器管理和控制 License，当应用服务器数量较多，业务模块运行繁忙的时候，因每个业务模块的每次 License 请求都需要向 License 服务器申请，License 服务器负荷重，同时也会增加网络的流量。

对于获取 License 的实时性要求严格的业务流程，例如语音会议业务，由于业务模块在得到 License 服务器的返回结果才能继续进行业务流程，因此业务模块的执行效率很大程度上受制于 License 服务器的运行效率以及应用服务器和 License 服务器之间的网络状况，在 License 服务器和网络负荷重的时候，获取 License 的实时性会受到严重影响。

如果 License 服务器业务量过大、系统故障等原因造成业务堵塞甚至崩溃，所有的业务模块都由于无法获得 License 从而中止运行。另外，如果应用服务器和 License 服务器之间网络异常，也将给业务模块带来极大的影响。因此对于业务模块和 License 服务器之间紧耦合的架构以及其方法流程会给 License 业务带来很大的运行风险，降低了系统的可靠性。

## 发明内容

本发明提供一种 License 控制方法、装置及系统，以解决现有技术中 License 集中管理导致的系统负荷重、实时性和可靠性不好的问题。

本发明提供以下技术方案：

一种 License 控制方法，包括步骤：

A: 应用服务器的业务模块向应用服务器的 License 客户端申请 License 许可;

B: License 客户端根据从 License 服务器获得的 License 全局数据对业务模块进行 License 许可, 所述 License 全局数据包括: License 服务器所有 License 数量、正在使用的 License 数量和剩余可用的 License 数量;

C: 业务模块获得 License 客户端的许可后执行业务操作。

其中:

在执行步骤 A 之前还包括 License 服务器的数据初始化步骤:

License 服务器读取 License 总数, 获取所有 License 客户端使用的 License 数量;

License 客户端暂停向业务模块提供许可, 并向 License 服务器返回使用的 License 数量;

License 服务器汇总各 License 客户端使用的 License 数量, 更新 License 全局数据, 并向所有的 License 客户端发送更新后的 License 数据;

License 客户端根据所述更新后的 License 数据, 更新本地可用的 License 数据, 并恢复向业务模块提供许可。

在执行步骤 A 之前还包括应用服务器的数据初始化步骤:

License 客户端清零本地使用的 License 数量并向 License 服务器发送消息;

License 服务器将所述 License 客户端使用的 License 数量清零, 更新 License 全局数据, 并向所有的 License 客户端发送更新后的 License 数据;

License 客户端更新本地 License 数据, 并根据更新后的数据向业务模块提供许可。

所述步骤 B 中 License 客户端从 License 服务器获取 License 数据的步骤具体为:

License 客户端定时向 License 服务器发送 License 增量数据, License 服务器根据所述 License 增量数据更新 License 全局数据, 并向所有的 License 客户端发送更新后的 License 数据, License 客户端更新本地的 License 数据;

或者,

License 客户端定时向 License 服务器发送其累计使用的 License 数量,

License 服务器根据所述累计使用的 License 数量更新 License 全局数据，并向所有的 License 客户端发送更新后的 License 数据，License 客户端更新本地的 License 数据。

所述增量数据指单个 License 客户端在一个更新周期内新占用以及新释放的 License 数量。

所述 License 许可包括：

判断是否可以申请新 License；或者

判断是否可以释放 License；或者

判断是否可以执行业务操作。

所有 License 客户端签发的 License 总数超过 License 服务器总的 License 数量。

本发明还提供一种 License 控制装置，所述装置包括：

业务模块，向 License 客户端申请 License 许可，并在获得许可后执行业务操作；

License 客户端，根据从 License 服务器获取的 License 全局数据对业务模块进行 License 许可，所述 License 全局数据包括：License 服务器所有 License 数量、正在使用的 License 数量和剩余可用的 License 数量。

在本发明的实施例中，将 License 的许可模块直接布署在应用服务器上，每个应用服务器均有单独的 License 客户端负责 License 许可，这样原本由 License 服务器独自承担的许可工作分解给众多的应用服务器，License 服务器的并发处理压力将得到很大缓解。同时，当应用服务器上的 License 客户端运行异常时，业务模块也可以直接向 License 服务器进行许可，这样的机制提高了系统的运行效率，且有效的解决了 License 服务器的性能瓶颈。

在本发明的实施例中，应用服务器不再直接向 License 服务器进行远程许可，而是向本地的 License 客户端进行许可，应用服务器和 License 服务器之间



是松耦合关系，本地调用比起远程调用来节省了网络传输和消息包编解码的开销；本地调用的效率和可靠性远大于远程调用；此外，License 客户端只需要对单个应用服务器提供许可服务，运行压力很小，因此可以支持更复杂、更灵活的 License 策略；由于分布式部署并且独立运行，License 服务器的暂时故障不会立刻影响到应用服务器的业务运行，License 服务器和应用服务器之间网络发生异常也不会立即影响到业务运行，系统的可靠性大大增强。

在本发明的实施例中，License 客户端获取的是前一次刷新的 License 资源，在下一次数据获得之前，如果遇到业务并发的峰值，多个 License 客户端签发的资源总数有可能超过总的 License 资源池，使得业务系统遇到大规模并发的业务高峰时允许 License 实际签发数量暂时超出业务提供商 ASP 拥有的 License 总数，因此本方案很好的支持了 ASP 的“License 重叠”运营模式，但是又不会给软件权利人的合法利益造成损失。

### 附图说明

图 1 为现有技术的 License 控制系统的组网示例；

图 2 为现有技术的 License 控制方法；

图 3 为本发明实施例的 License 控制系统的组网示例；

图 4 为本发明实施例的 License 控制方法流程图；

图 5 为本发明实施例的 License 服务器的初始化流程；

图 6 为本发明实施例的应用服务器的初始化流程；

图 7 为本发明实施例的 License 服务器从应用服务器获取数据的过程。

### 具体实施方式

为使本发明的目的、技术方案及优点更加清楚明白，下面参照附图并举实施例，对本发明作进一步详细说明。

请参阅图 3，License 服务器管理整个 License 资源，在系统启动时将 License

全局数据同步给所有的应用服务器，并根据应用服务器的请求定时将最新的 License 数据发送给应用服务器。其中，License 全局数据主要包括 License 服务器所有 License 数量、正在使用的 License 数量、剩余可用的 License 数量。

每个应用服务器包括至少一个 License 客户端和至少一个业务模块。所述业务模块向本端应用服务器的 License 客户端请求 License 许可，在获得 License 客户端的许可后进行业务操作。所述 License 客户端根据从 License 服务器获得的 License 数据对业务模块进行 License 许可，并向业务模块发送许可响应消息。所述 License 客户端还定时将 License 数据发送给 License 服务器，并根据 License 服务器的响应消息更新本地的 License 数据。其中，License 客户端发送给 License 服务器的 License 数据为 License 增量数据或 License 累计数据，所述增量数据指单个 License 客户端在一个刷新周期内新占用以及新释放的 License 数量，所述 License 累计数据是指单个应用服务器从启动开始累计占用以及释放的 License 数量。

根据图 3 描述可知，License 服务器存放的数据主要包括各 License 客户端使用的 License 数量以及最新 License 全局数据。在第一实施例中，即 License 服务器和 License 客户端交互增量数据时，License 客户端存放的是增量数据和本地可用的 License 数据。在第二实施例中，即 License 服务器和 License 客户端交互其累计使用的 License 数量时，License 客户端存放的是其累计使用的 License 数量和本地可用的 License 数据。与此相应，License 客户端中存放的数据主要包括 License 增量数据或累计使用的 License 数量，以及可用的 License 数量其中。其中 License 客户端中的数据还可存放在其他的模块或装置中。

请参阅图 4，是本实施例的一种 License 控制方法的流程图，包括如下步骤：

A: 应用服务器的业务模块向本应用服务器的 License 客户端请求 License 许可；

**B: License 客户端根据从 License 服务器获取的 License 数据对业务模块进行 License 许可;**

**C: 业务模块在获得 License 客户端的许可后执行业务操作。**

因 License 服务器和 License 客户端是分布式的架构, 两者的运行状态以及两者存放的数据并不是时刻同步的, 当 License 服务器启动或重新启动时, 需要对 License 服务器上的数据进行初始化操作。请参阅图 5, 是 License 服务器的数据初始化步骤为:

**License 服务器读取 License 总数, 向所有应用服务器的 License 客户端获取其使用的 License 数量;**

**License 客户端暂停向业务模块提供许可, 并向 License 服务器返回其使用的 License 数量;**

**License 服务器汇总各应用服务器使用的 License 数量, 生成 License 全局数据, 并向所有的 License 客户端提供更新后的 License 数据;**

**License 客户端更新本地可用的 License 数据, 并恢复向业务模块提供许可。**

应用服务器有可能会异常终止, 在应用服务器终止之后 License 服务器会认为该 License 许可仍然占用着 License 资源。因此当应用服务器启动或重新启动时, 需要向 License 服务器发消息, 让 License 服务器初始化该 License 客户端的 License 数据, 即需要对应用服务器的数据进行初始化, 请参阅图 6, 为应用服务器的初始化步骤:

**License 客户端清零本地使用的 License 数量并向 License 服务器发送消息;**

**License 服务器将所述 License 客户端使用的 License 数量清零, 更新 License 全局数据, 并向所有 License 客户端发送更新后的 License 数据;**

**License 客户端更新本地可用的 License 数据, 并根据更新后的本地 License 数据向业务模块提供许可。**

请参阅图 7, 是应用服务器从 License 服务器获取 License 数据的步骤:

**License 客户端定时向 License 服务器上本端的 License 数据;**

License 服务器根据上报的 License 数据更新 License 全局数据，并向所有 License 客户端发送更新后的 License 数据，其中所述更新后的 License 数据至少包括可用的 License 数据；

License 客户端更新本地的 License 数据。

在具体实施例中，所述 License 客户端定时向 License 服务器上报本端的增量数据或累计使用的 License 数量。若上报的为增量数据，则 License 客户端更新本地的 License 数据时将增量数据清零。

可选地，License 客户端从上报 License 数据至更新本地的 License 数据期间，暂停提供 License 许可。

业务模块直接访问本地的 License 客户端进行 License 许可，主要有三种许可操作：判断是否可以执行业务或申请新的 License 或释放 License。

在申请新的 License 时，如果许可操作为申请新的 License，则 License 客户端根据本地的 License 全局数据判断剩余的可用 License 数量是否足够，如果 License 数量不够则拒绝业务模块的申请，如果足够则向业务模块提供许可成功响应消息，并将本地的可用的 License 数据数量减 1。

如果许可操作为释放 License，则将本地可用的 License 数据数量加 1。

显然，本领域的技术人员可以对本发明进行各种改动和变型而不脱离本发明的精神和范围。这样，倘若对本发明的这些修改和变型属于本发明权利要求及其等同技术的范围之内，则本发明也意图包含这些改动和变型在内。

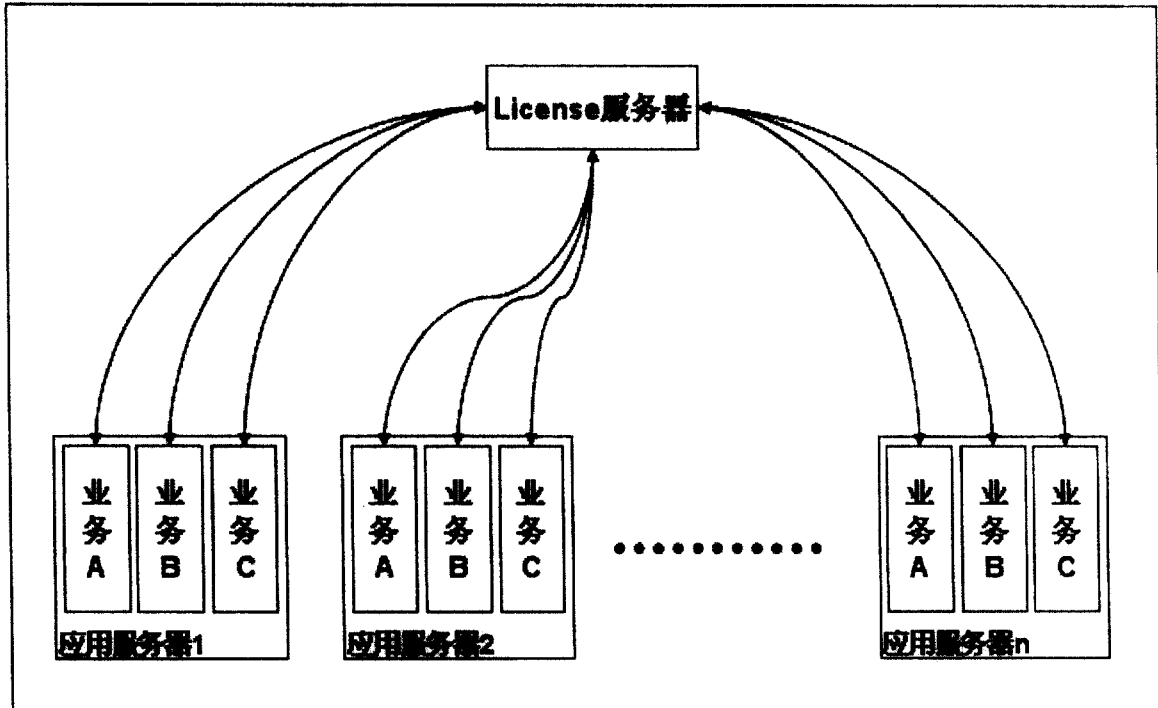


图 1

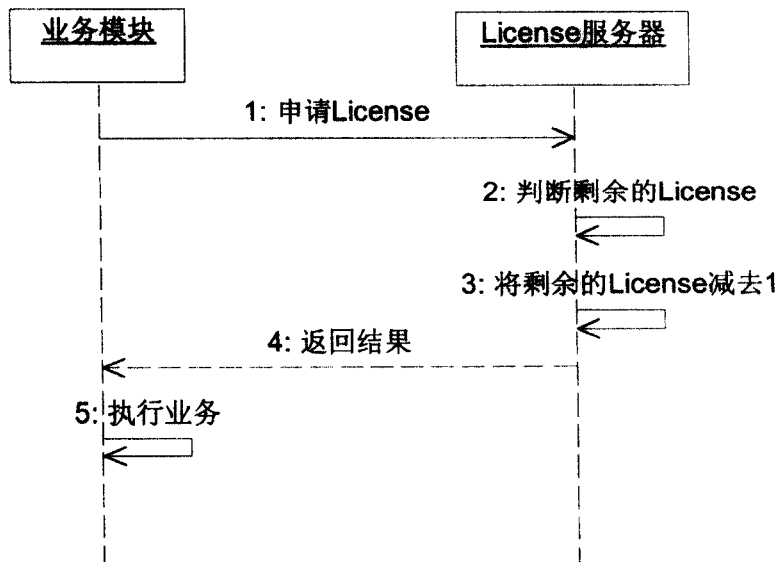


图 2

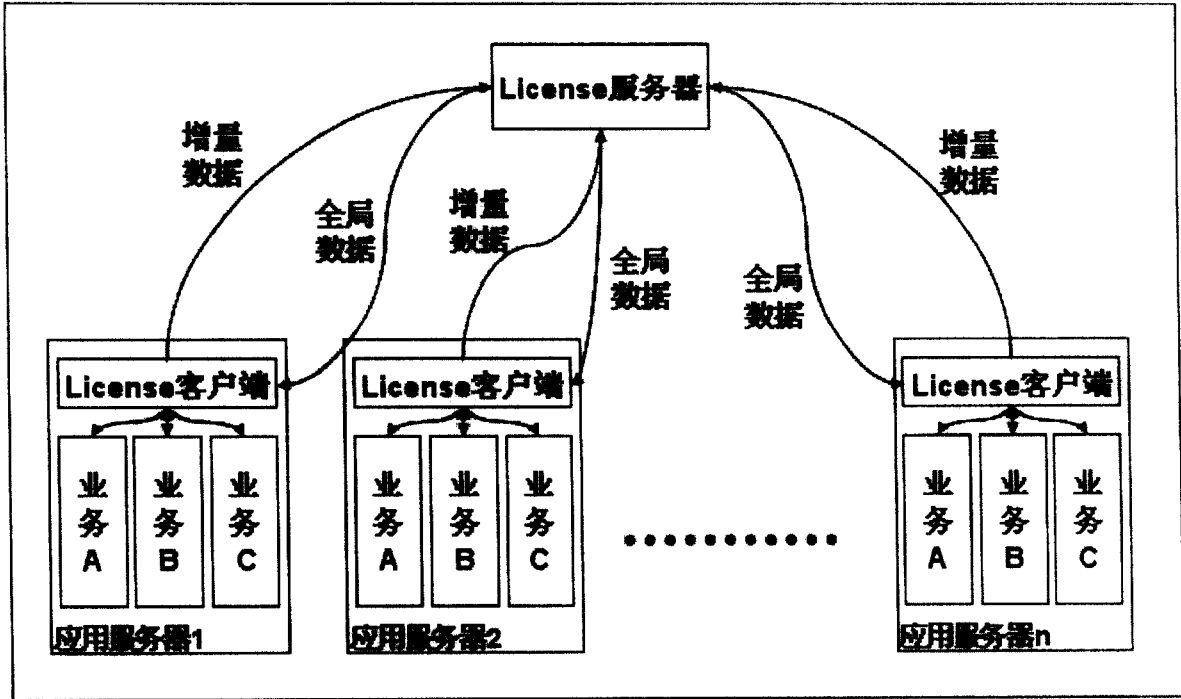


图 3

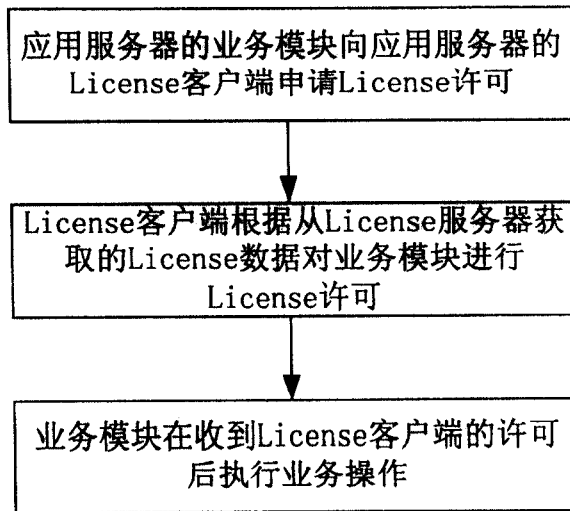


图 4

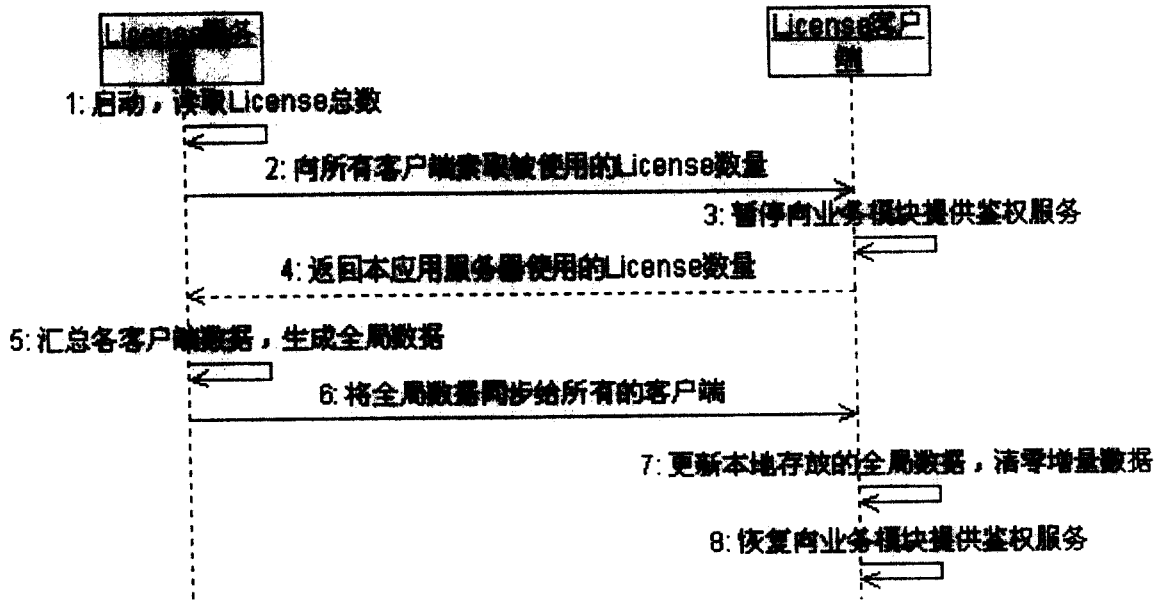


图 5

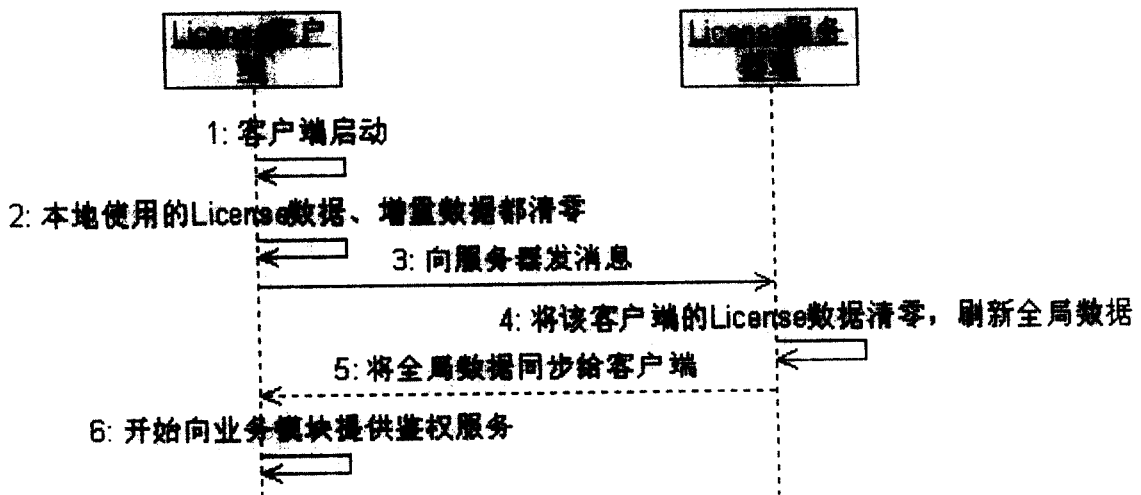


图 6

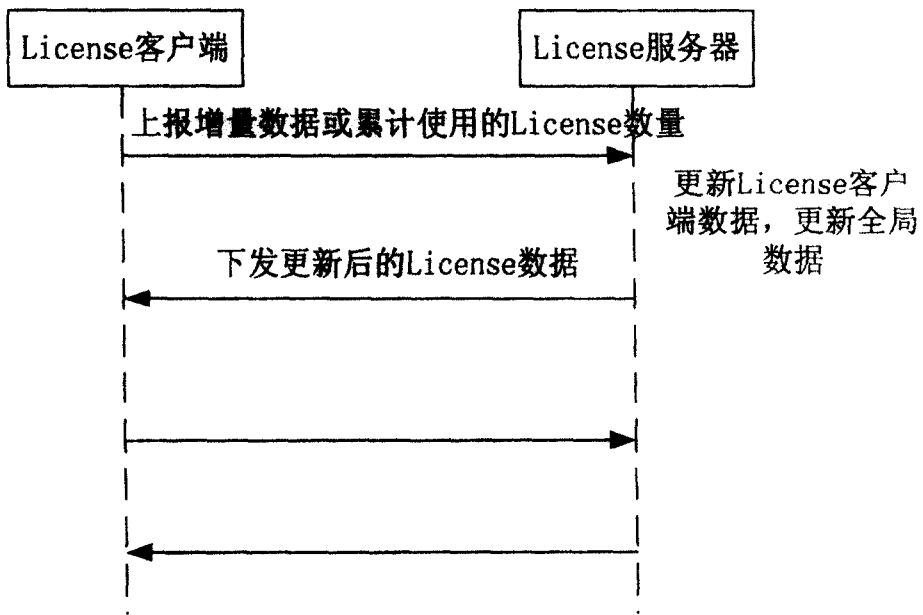


图 7