



运营技术广角

云服务时代下的开源发展

孙福洲, 钱瑾, 杨静, 钱岭
(中移软件技术有限公司, 江苏 苏州 215153)

摘要: 随着云计算技术的迅速发展和应用, 现代社会已经步入了云服务的时代, 而软件即服务模式更是改变了传统的商业模式。在此背景下, 大部分开源许可证的强制开源规定已经被规避, 现有的开源许可证体系已不能有效地维持开源软件的版权和利益平衡, 同时, 版权和利益的主体之间的发展关系也变得更加紧张, 整个开源发展也迎来了新的挑战。以开源软件的版权和利益平衡为切入点, 从平衡手段—开源许可证体系、版权和利益主体两个方面分别探究了新时代下的开源发展策略, 期望在云服务时代下, 开源许可证体系能够更好地维护版权和利益的平衡, 版权和利益的主体能够更加良好地竞争, 整个开发生态更加健康, 开源发展能够迎来新的跨越。

关键词: 云服务; SaaS; 开源许可证; 版权和利益; 开发生态; 开源发展

中图分类号: TP368

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2020298

Open source development in the cloud services era

SUN Fuzhou, QIAN Jin, YANG Jing, QIAN Ling
China Mobile Software Technology Co., Ltd., Suzhou 215153, China

Absrtact: Modern society has stepped into the cloud services era with the rapid development and application of cloud computing technology, especially software as a service mode, which has changed the traditional business model. The source code's passive publication related regulations of most open source licenses have been avoided, the current open source license system can not effectively maintain the balance of copyright and interest, meanwhile, the development relationship between copyright and interest subject-open source manufacturers and cloud manufacturers became even more strained, the whole open source development has also met with great challenges. The copyright and interest balance as the starting point was proposed, and the new time's open source development strategy was explored from two aspects: the balance means-open source license system, the copyright and interest subject. It is expected that in the cloud services era, the open source license system can efficiently maintain the balance of copyright and interest, the subject of copyright and interest can be more competitive, the whole development ecology can be more healthy, and the open source development can usher in a new leap forward.

Key words: cloud service, SaaS, open source license, copyright and interest, development ecology, open source development

1 引言

在云服务时代中,全球各大云厂商例如 Amazon、Azure 等都使用了大量的开源软件,开源许可证伴随着开源软件而产生,开源许可证提供了对开源软件使用的许可授权,其恪守开源精神,保证了任何人使用开源软件的自由,平衡了开源软件的版权保护以及利益共享,促进了开源技术的传播和发展、促进了生活和经济的迅猛发展^[1-2]。强制开源是指依照许可证的规定使用开源软件后需要将源码公开的情形,给出了主流许可证中较受欢迎的几种许可证,见表 1。可以看出,在云服务的软件即服务 (software as a service, SaaS) 模式下,开源许可证中大部分的强制开源规定已经被规避,云厂商能够以不反馈贡献给开源社区的方式免费使用大部分开源软件^[3],尤其是开源促进会 (open source initiative, OSI) 认证的典型强传染性许可证,此许可证的强传染特性甚至是早期 Linux 技术能够快速发展的最大功臣之一,但在此模式下其对强制开源的规定基本都已经被规避,所以在此背景下大部分的开源许可证平衡性已有所倾斜。

于是有了 Neo4j 的企业版仅在商业许可下提供,Confluent 的部分组件从 Apache 2.0 切换到 Confluent Community License,Redis Labs 的一些模块从 AGPL 变更 Commons Clause 和 Apache 2.0 组合、MongoDB 的 AGPL 切换到 SSPL 等。其中,MongoDB 是近些年来发展势头迅猛,不可替代性较强的一种 NoSQL 数据库,据数据库引擎 (db-engines.com) 显示,截至 2020 年 7 月,MongoDB 在全球全部数据库中排名第五,文档型数据库中排名第一,并远远领先第二名;同时,MongoDB 涉及的 AGPL、SSPL 许可证都是针对网络服务模式的许可证,下文将以 MongoDB 为例出发研究 SaaS 模式下的开源许可证体系,探究开源厂商和云厂商的发展关系和策略。

2 开源许可证体系研究

2.1 GPL、AGPL、SSPL 许可证对比

本文中所述的开源许可证是指通过 OSI 认证的许可证,现通过 OSI 认证的许可证约有 100 种,例如近期新加入的木兰协议,这也是中国国内的首例开源许可证。主流使用的开源许可证的特性对比见表 1,其中仅 SSPL、AGPLv3、GPLv3 三

表 1 主流使用的开源许可证的特性对比

开源许可证	衍生作品	被动开源		专利	商标
		线下	网络互动		
SSPL	服务器端使用和修改	是	是	是	否
AGPLv3	包括程序全部或部分,完整或修改或翻译	是	是(修改)	是	否
GPLv3	包括程序全部或部分,完整或修改或翻译	是	是(组合)	是	否
GPLv2	包括程序全部或部分,完整或修改或翻译	是	否	否	否
LGPLv2.1	基于函数库的作品	是	否	否	否
Apachev2	基于作品(或从作品衍生而来)源程序形式或目标形式任何作品	否	否	是	否
BSD 2-clause	无	否	否	否	否
BSD 3-clause	无	否	否	否	否
MIT	无	否	否	否	否



者有针对网络互动方面的规定,即关于 SaaS 模式的规定。其中 GPL、AGPL 都是属于 OSI 认证的开源许可证,并且也都属于主流使用的许可证,而 SSPL 并未得到 OSI 认证,例如 MongoDB 修改为 SSPL 许可证后向 OSI 提出认证申请但并未通过,本文认为其过于针对 SaaS 模式以及存在明显的强制组合规定,违背了开源精神^[1]的第 6 条、第 9 条。进一步地,基于云服务 SaaS 模式下的研究需求,此处主要对 GPL、AGPL、SSPL 关于网络服务模式相关的规定进行对比。

- **GPL**: GPL 的灵魂之一在于其衍生(derived)作品的规定,但整个 GPL 许可证并没有单独关于网络服务模式的规定,GPLv3 中第 13 条规定了与 AGPLv3 一起使用的情形——当合并(link or combine)为一个著作(a single combined work)时,AGPLv3 的第 13 条将适用于这样的组合著作(combination),同时应注意到 GPLv2 与 AGPLv3 并不兼容^[4]。
- **AGPL**: AGPL 俗称网络增强版 GPL,AGPLv3 关于网络部分的相关规定主要在第 13 条,其规定了当修改程序后通过远程网络互动(remote network interaction)时,则须公开对应源码,对应源码指生产、安装和(可执行著作)运行目标代码以及修改著作所需的所有源码,包括用于控制这些活动的脚本以及规定了与 GPLv3 一起使用的情形,当组合为一个著作时,本条款继续适用于受保护著作部分,参与组合的著作仍受 GPLv3 的约束。
- **SSPL**: SSPL 是直接针对程序即服务(program as a service)模式的许可证,其规定了使用本程序或修改版作为服务提供给第三方,则须公开服务源代码,服务源代码包括管理软件、界面、存储软件等所有

服务器端运行服务的程序。

从上述对比以及表 1 中可以看出,GPLv3 许可证本身对 SaaS 模式并没有单独的相关规定,而 AGPLv3、SSPL 是有着比较针对性规定的。其中 AGPLv3 许可证既是通过 OSI 认证的主流许可证,又有着关于网络服务方式的相关规定,其在云服务时代的背景下理应能独当一面,但却是一直不温不火,主要原因是自身的强制开源规定不够明确^[5];而 SSPL 的相关规定虽然足够明确,但本文认为其针对 SaaS 的规定有悖于开源精神,或者至少明显倾向于商业竞争,达不到开源许可证在版权保护和利益共享上的平衡效果,这也是 SSPL 许可证的开源软件无法商用的原因。

2.2 云时代 SaaS 模式下的开源许可证体系

对于开源软件,开源许可证在平衡版权和利益之间起到了重大的作用,为开源发展做出了重大的贡献。但作为明文条款规定形式存在的许可证,其同法律一样,存在滞后性,即随着社会的发展,制定许可证时的认知以及当时对未来的预测总是有限的,而社会经济、商业模式又是快速发展的,当时制定的许可证条款对新情况、新商业模式顾及不够全面,例如在 SaaS 模式下,软件版权受到了极大的冲击。面对许可证的滞后性问题,有学者提出了采用版权出租权扩张解释结合 SaaS 服务协议以及开源许可证的方式来解决^[6-7],而此处希望以完善许可证体系的方式来保障 SaaS 模式下开源精神的传承、开源厂商和云厂商的健康发展,并不断地改善优化许可证体系,使其跟上社会的脚步,适应新时代、云服务时代的发展。进一步地,完善和优化应至少包括 AGPL 使用方案明确化、新型网络版宽松型开源许可证两个方面,如图 1 所示。

AGPL 开源许可证在 OSI 认证、使用广泛性、应对 SaaS 模式等方面是有着天然优势的,但要想提高其可使用性,需要克服其强制开源规定方面的不明确性。其不明确性主要体现在 3 点:如何

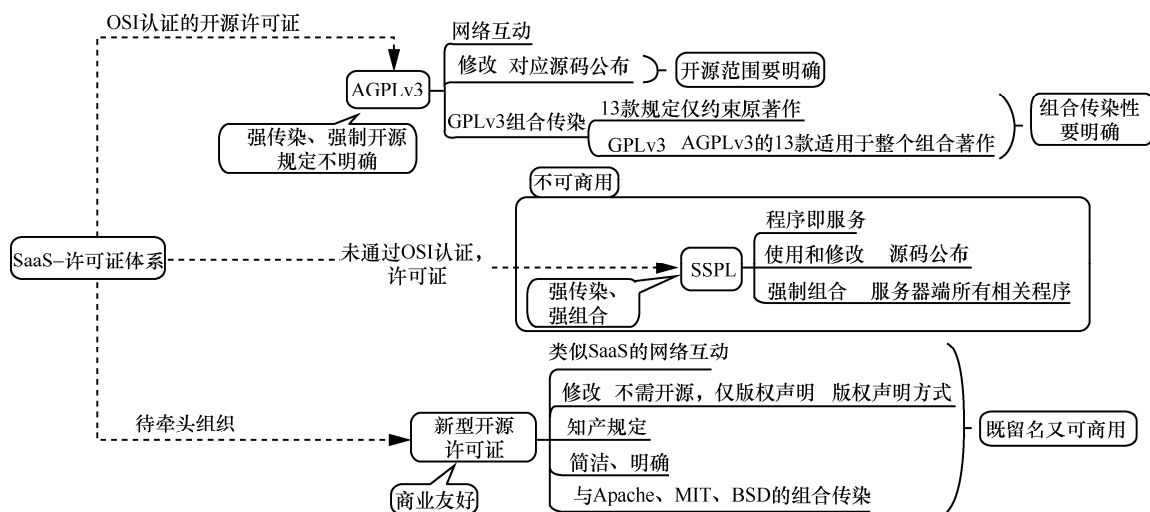


图1 完善和优化的许可证体系示意图

界定远程网络互动、如何界定对应源码、如何界定涉及 GPLv3 联用时的组合传染性。此处以 MongoDB 为例具体分析上述 3 点,以将其明朗化。MongoDB 的 4.0 版及以前版本是 AGPL 版本,4.2 版已更新为 SSPL 版本,对于 AGPL 版本的修改使用,通过模拟其使用场景来研究相应的使用方案,提高 AGPL 许可证的可使用性。

场景 1 SaaS 模式下,用户通过控制台提出数据库实例需求,控制台通过后台服务等完成

MongoDB 的部署,服务器端实现客户需求的流程可能涉及前端界面、监视软件、存储软件、备份软件、控制台、传输软件等,MongoDB 的部署、主要工作原理、关键组件如图 2 所示,当客户通过云网络提出定制实例需求时,部署工作流程为:控制台与 MongoDB 后端服务组件交互相关信息,后端与资源管理组件交互动态创建虚拟机(VM)、并与 MongoDB 部署工具交互信息,在 VM 上启动 VM 管理组件,VM 管理组件向部署工具进行

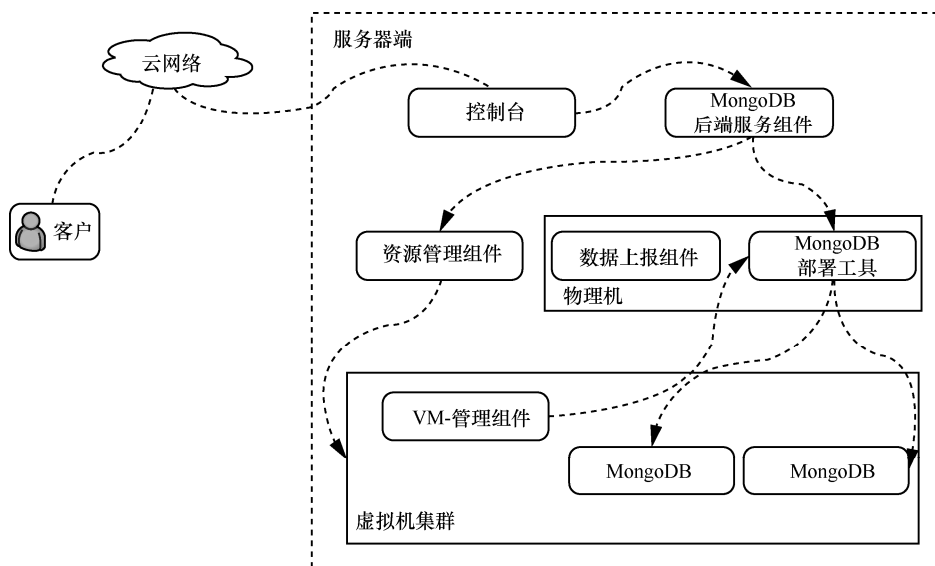


图2 MongoDB 部署的原理示意图



注册,完成 MongoDB 的部署;其中,组件都可以选择领域内常用的开源组件,例如资源管理组件可以为 OpenStack、部署工具可以为 Mongo DB OpsManager (MMS)、管理组件可以为 Agent。对于此场景,本文详细分析了 3 点难题。

(1) 对于远程网络互动的界定,本文认为此处的 SaaS 模式完全属于 AGPL 许可证中规定的远程网络互动,这点在此场景中是比较容易确定的。

(2) 对应源码,许可证记载的规定为:应公开修改版本(modified version)的对应源码,著作(work)目标代码形式的对应源码指生成、安装和(可执行著作)运行目标代码以及修改著作所需的所有源码,包括用于控制这些活动的脚本,与著作源文件相关的接口定义文件,共享库的源码以及基于著作需求专门设计的动态链接子程序(比如用于子程序和著作其他部分的亲密数据通信或控制流程)。为了清晰的界定上述的对应源码,应该抓住以下 3 点核心。

1) 基于许可证通用的分析方式,先定义作品,此处将部署 MongoDB 完成客户需求的一整套可能涉及的流程作为一个总作品(或联合作品、联合著作),即包括前端界面、监视软件、存储软件、备份软件、控制台、传输软件、数据库等,修改后的 MongoDB 版本作为著作。

2) 作为 GPL 加强版的许可证,其可以借鉴 GPL 中的独立著作(independent/separate work)的含义。

3) 作为恪守开源精神的许可证应以“传播使用自由、促进开源技术发展”为指导,忌将对应源码界线过于扩张至商业竞争性质。

基于上述 3 点核心,本文认为 APGL 的被动开源实质范围应是:“您可以修改此著作,但是应确保其他人能够获取著作的所有源码,并且能够在现有技术的基础上正常运行和使用或修改您修改后的著作”。再次回到对应源码的界定上:首先,修改程序版本即 MongoDB 修改后的版本,对应

源码处的规定虽然采用了著作(work)术语,但基于第 13 条的规定应公开修改版本的对应源码,此处对应源码改版 MongoDB 的对应源码;其次,生成、安装以及修改所需的源码即是指生成、安装、修改版 MongoDB 目标代码的源码,脚本也即上述活动中相应的控制脚本,这些实质上属于 MongoDB 著作本身的范畴;最后运行所需源码的界定应结合对应源码后半段的规定来综合分析,具体地,在运行改版 MongoDB 时涉及的特定接口、特定设计的通信和控制子程序其特定接口以及特定设计的与总作品其他部分第一层动态链接的通信、控制子程序应当同时公开源码,例如基于新改功能而使用的特定 API、相应的 MongoDB 部署工具;而虽然运行改版 MongoDB 时可能涉及总作品,但其他例如监视软件、控制台等是相对独立的,也不是亲密通信的,不属于对应源码范围。因此,SaaS 模式下,在场景 1 的情形或类似情况中,修改 AGPL 程序后,应至少公开 MongoDB 本身范畴的所有源码、特定的 API、相应的部署工具。

场景 2 此场景与场景 1 的区别为控制台涉及 GPLv3 许可证,其他与场景 1 一致。对于此场景,首先,基于客户提出的数据库实施需求,认为整个流程总作品中至少 MongoDB 与控制台以及二者之间必要的通信等程序是完成需求所必需的,其组合属于 AGPLv3、GPLv3 中第 13 条规定的组合著作(combined work),而其他例如备份软件、监视软件等辅助性质、通用性质或者优化性质的程序等不属于组合著作;其次,关于 MongoDB 修改后的强制开源部分同场景 1;最后,因此处控制台涉及 GPLv3 许可证,且控制台和 MongoDB 属于一个组合著作,所以存在网络条款规定的传染性。此时,根据 GPLv3 的第 13 条规定,当其与 AGPLv3 组合时,第 13 条应适用于整个组合著作,即当 GPLv3 组合 AGPLv3 著作时,也就变成了网络增强版的 GPLv3 约束的组合著

作, 此处控制台相关修改部分同样要按照场景 1 中分析的情况去强制开源。

场景 3 此场景与场景 2 的区别为客户需求为完成 MongoDB 的备份, 其他与场景 2 一致。基于客户的需求认为整个流程总作品中至少 MongoDB、备份软件与控制台以及三者之间必要的通信等程序的组合属于 AGPLv3、GPLv3 中第 13 条规定的组合著作 (combined work), 即与场景 2 相比场景 3 又多了备份软件中 GPLv3 部分的强制开源, 备份软件相关修改部分同样要按照场景 1 中分析的情况去强制开源。

综上, 通过全场景的模拟, 对 AGPL 修改中的被动开源范围一对应源码和组合传染进行了充分的研究, 对应源码应依据“您可以修改此著作, 但是应确保其他人能够获取著作的所有源码, 并且能够在现有技术的基础上正常运行和使用或修改您修改后的著作”这样的实质界定, 组合传染性应依据 SaaS 模式下客户的具体需求来界定。通过上述分析, AGPL 的使用方案已经变得更加明确化, 这样能够有效地提高其在云服务时代下的可用性。

2.3 SaaS 模式下的新型开源许可证研究

近年, 主流开源许可证中强传染性许可证 GPL 的使用占比在逐渐下降, 而宽松型开源许可证例如 Apache、BSD、MIT 的使用占比则在不断增长, 可以看出开源用户越来越偏好宽松型的许可证。虽然通过研究让 AGPL 的使用方案明确化并提高其可使用性, 但是依然避免不了 AGPL 即使明确化也仍属于强传染性开源许可证的事实, 这对于现代的开源用户来说, 可能依然不能满足他们的需求, 达不到预想中的 AGPL 许可证独当一面的效果。因此, 在云服务时代的背景下, 开源的发展同样需要一些相应的宽松型开源许可证, 更好地平衡版权和利益、开源开发者和开源用户之间的关系。SaaS 模式下的新型网络版宽松型开源许可证应至少包含如下相关条款。

(1) 存在关于 SaaS 模式的相关规定和限制, 但是不能存在对 SaaS 模式歧视的规定, 且商业友好。

(2) 许可证内容清晰明了、简洁不拗口, 例如语言可以是类似木兰协议的中英文双语版, 减少法律性晦涩的术语使用。

(3) 知识产权方面规定清晰, 例如专利、商标都有明确的使用规定, 但应注意关于专利的例如许可或者反制规定涉及的范围要合理, 不能重蹈当初 Facebook 采用带有专利附属规定 BSD 许可证的覆辙。

其中, 最难、最迫切的是第 (1) 点, 此处提出一些构思希望能够为行业专业人士提供一定的借鉴意义。触发方式可以增加远程网络服务或者程序即服务或者 SaaS 模式; 衍生作品的规定可以借鉴 Apache 的规定; 不设置强制开源的规定, 采用设置许可证、开源软件明确声明或主张的方式保护版权, 例如规定在通过触发方式进行服务时, 应当在相关云主页明确列出此许可证或开源软件, 或者在向客户提供服务前应当主动传送许可证副本或开源软件使用声明; 可借鉴 GPL 和 AGPL 组合传染的方式, 更新 Apache、BSD、MIT 的版本, 在其中增加与上述新型开源许可证组合使用适用于整个组合的主动传染规定。

网络版宽松型的开源许可证结合 AGPL 能够形成适用于云服务时代的一个基本的许可证体系。完善开源许可证体系的方式能够直接有效地平衡版权和利益、开源开发者和开源用户的关系, 促进开源的发展。

3 云服务时代下的开源厂商和云厂商发展策略

云服务时代下, 开源软件版权和利益主体的典型代表是开源厂商和云厂商, 除了完善许可证体系促进开源发展之外, 还需要开源厂商、云厂商实施健康的发展策略积极推动开源的发展。

对于开源软件, 开源厂商一般会包括两方面



的运营,一是自身 SaaS 商业业务,二是反馈开源社区的开源开发,即企业版和社区版软件开发运营,其中有些厂商会将核心模块闭在企业版中,有些厂商会将核心模块也开源到社区版中。云厂商一般会包括 3 方面的运营:一是自身 SaaS 商业业务,这也是主要业务,可以是自研,或是开源软件的闭源应用等;二是主动向开源社区推出自己的开源项目,目的是集众开发者之力优化项目、推动事实技术标准、提高形象等;三是积极参加社区开源项目并反馈贡献。在开源许可证的平衡下,开源厂商和云厂商保持着较好的竞争关系。但是在如今的云服务时代,由于新型商业模式 SaaS 的发展,云厂商极大地提高了对开源软件的商业使用度,在此情形下开源厂商尤其是将核心模块也开源出来的开源厂商自身的商业业务受到了极大的冲击。例如 MongoDB,其社区版中也是包含核心功能的,在 SaaS 模式下,由于各云厂商例如 AWS、Azure 等纷纷推出了与 MongoDB 社区版相关的云服务产品 DynamoDB、DocumentDB、CosmosDB,使 MongoDB 自身 SaaS 服务 MongoDBAtlas 受到了极大的挑战,于是导致了后续的 SSPL 许可证修改。对于这种情况开源厂商

以及云厂商应该如何发展?云时代下开源厂商、云厂商的发展关系和策略示意图如图 3 所示。本文认为对于开源厂商、云厂商而言,无论是从提高自身竞争力还是从促进开源发展的角度,其都应该积极运营或参加开源社区、开发者社区等促进开源技术发展的组织,维护好开源精神的传承,维护好开发生态。具体的发展关系以及发展策略如下。

(1) 对于 MongoDB 修改许可证的情况,有人认为 MongoDB 存在先开源再“割韭菜”之嫌疑,也有人认为云厂商过于“拿来主义”,违背开源本意。对于上述观点,值得鼓励的是大家都能够自发地维护开源精神、开发生态,懂得维护行业的核心财富,但是对于这种情况不能简单地将其归咎于是否违背了开源精神、破坏了开发生态,实质上在 SaaS 模式下的这种情况,更准确地理解是将此种情况归类于纯粹的商业竞争,而不应当也不需要以开源精神、开发生态去绑架开源厂商或者云厂商,其实质上是因为版权和利益的平衡发生倾斜后的竞争者之间的常规商业博弈,这也是随着时代的发展开源厂商和云厂商之间竞争关系的一个适应性变化,而应对此情形,开源厂商、

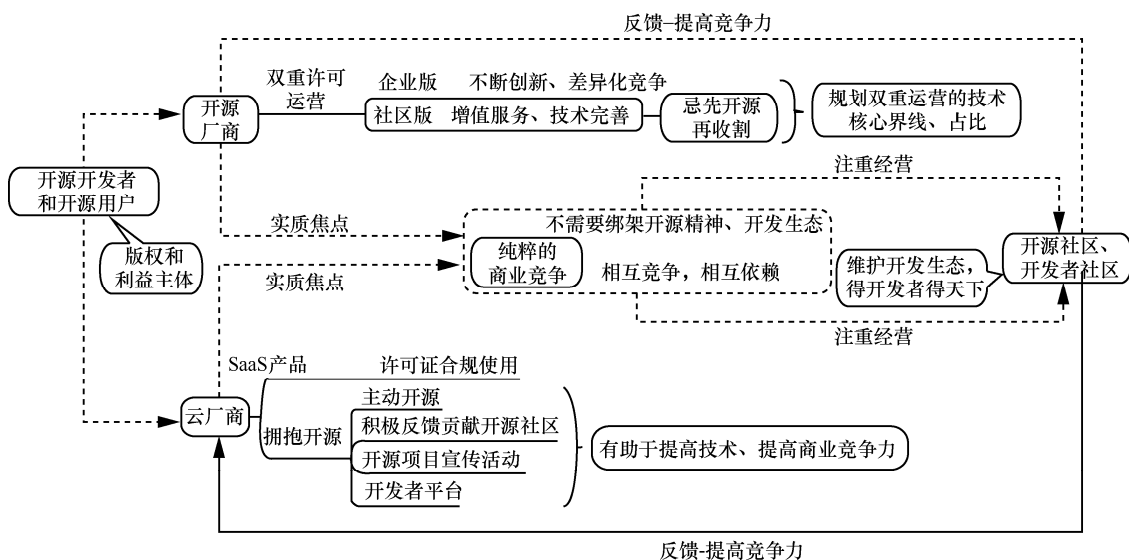


图3 云时代下开源厂商、云厂商的发展关系和策略示意图

云厂商最需要做的是采用围绕核心财富—开源精神、开发生态这样的发展策略。

(2) 对于开源厂商, 其应当采取的策略是: 首先, 应对社区版和企业版两方面的技术和功能有着明确的规划^[3], 例如核心功能是否开源、开源比例, 并且在社区版的运营中, 要谨慎选择以什么样的许可证推出项目, 应避免“宽出紧入”这种先开源再“闭源”的情况。其次, 虽然 MongoDB 认为其社区版仅是作为增值服务策略, 不是用于收集众用户的贡献, 但实际上并不能忽略众开发者对其的贡献, 这也是其从开源中的获利之一, 所以, 开源厂商应积极给予众贡献者予认可, 并且注重从众贡献者的反馈和贡献中发掘值得优化的地方, 保持与开源社区等的积极互动交流。最后, 对于开源厂商而言, 最好的运营方式是一方面规划和调整核心功能或更新在社区版、企业版上的占比; 另一方面是不断创新和集成新功能, 利用这种不断发展的差异化保持其竞争力(例如其 SSPL 许可证的 4.2 版更新了分布式事务、通配符索引、字段级加密等一系列重磅新功能), 实质上开源厂商可以利用这种自身技术不断发展的差异化来提高其竞争力、可持续发展。

(3) 对于云厂商, 其应当采取的策略是: 首先, 应认识其 SaaS 模式本身并没有问题; 其次, 云厂商需要注重许可证的合规使用; 最后, 应注意开源。其中尤其需要注意的是, 在 SaaS 模式对版权、许可证体系的形成巨大冲击的背景下, 云厂商更要注意开源, 开源可以是主动向社区推自己的开源项目(例如基于技术完善、技术宣传、许可证宣传和推动等目的); 积极反馈贡献社区(例如对于自身 SaaS 产品涉及使用的开源软件积极提交 bug 反馈、代码贡献等); 对开源项目的积极宣传和推动; 主动参与维护开发生态^[8], 为开发者提供好的交流平台。引用行业内流行的话“得开发者得天下”^[9], 云厂商在积极开源方面的运营其实并不违背自身商业业务运营盈利的意

愿, 积极开源既能提升自研技术又能得到开发者的拥护, 这是商业大步创新发展的坚实后盾。

因此, 云服务时代下的开源厂商和云厂商二者之间是相互依赖相互竞争的关系, 云厂商需要用到开源厂商的开源技术, 开源厂商需要借助云厂商的资源使用推广, 二者之间的商业竞争都不需要绑架到开源精神、开发生态维度上, 但得开发者得天下, 二者都可以从开发生态的经营上去提高自己的竞争力。

4 结束语

SaaS 模式已经成为云服务时代下的一种特色, 并且还在不断衍生类似的模式, 其本身既是时代发展的必然产物, 也将成为推动时代再次发展进化的助力。本文研究了云服务时代 SaaS 模式下新型许可证体系以及版权和利益主体的发展策略, 为开源发展提供有力的保障和支撑。以期开源发展纵向能够推动行业前沿技术的突破, 横向能够推动垂直行业的应用, 促进社会技术和经济的迅速发展。

参考文献:

- [1] LAWRENCE R. Open source licensing : software freedom and intellectual property law[M]. New Jersey: Prentice Hall Professional Technical Reference, 2004.
- [2] ERIC D, DOMINIQUE T. Dual licensing strategy with open source competition[EB]. (2017-04-19)[2020-11-13].
- [3] YEGULALP S. SaaS doesn't have to gum up open source's code sharing[J]. Information Week, 2008(1188): 21.
- [4] 李斌, 杜涛, 钱岭. 开源软件法律风险分析与防范[J]. 信息技术与应用, 2019(1): 29-32.
LI B, DU T, QAIN L. Legal risk analysis and prevention of open source software[J]. China Science & Technology Overview, 2019(1): 29-32.
- [5] 付娜, 秦乐, 毕春丽. 从 MongoDB 更换开源许可协议谈开源软件法律风险[J]. 信息通信技术与政策, 2019(1): 7-10.
FU N, QIN L, BI C L. Discussion on legal risk of open source software from MongoDB replacing open source license[J]. Information and Communications Technology and Policy, 2019(1): 7-10.
- [6] 董颖. 解决云计算 SaaS 模式下版权问题的另一途径[J]. 电子知识产权, 2012(12): 54-59.
DONG Y. Another way to solve copyright problem in SaaS mode of cloud computing[J]. Electronic Intellectual Property, 2012(12): 54-59.



- [7] 刘静. SaaS 模式下计算机软件版权保护的困境与出路[D]. 重庆: 西南政法大学, 2016.
LIU J. Plight and answer of computer software copyright protection under SaaS mode[D]. Chongqing: Southwest University of Political Science & Law, 2016.
- [8] 黄海峰. 华为丁耘: “沃土”计划将使开发者生态有序化 与运营商合作成绩斐然[J]. 通信世界, 2015: 27.
HUANG H F. Huawei DING Yun: the “fertile soil” plan will make the ecological order of developers and the cooperation with operators achieve remarkable results[J]. Communication World, 2015: 27.
- [9] 柯海. 互联网巨头的终极战场得开发者得天下[J]. 计算机与网络, 2018: 38-41.
KE H. The ultimate battlefield of the Internet: giant is the developer[J]. Computer and Network, 2018: 38-41.

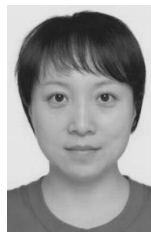
[作者简介]



孙福洲（1990- ），男，中移软件技术有限公司助理知识产权管理专员，主要研究方向为开源协议和专利。



钱瑾（1984- ），女，中移软件技术有限公司知识产权管理专员，主要研究方向为知识产权管理。



杨静（1982- ），女，中移软件技术有限公司高级知识产权管理专员，主要研究方向为知识产权管理。



钱岭（1972- ），男，博士，中移软件技术有限公司首席科学家，主要研究方向为大数据与云计算。