上 海 交 通 大 学

工程硕士生开题报告登记表

|  |  |
| --- | --- |
| **学 号：** | **1130379067** |
| **姓 名：** | **李会一** |
| **导 师 Ⅰ：** | **饶若楠** |
| **单 位：** | **上海交通大学** |
| **导 师 Ⅱ：** |  |
| **单 位：** |  |
| **工程领域：** | **软件工程** |
| **专业方向：** | **软件工程** |
| **班 级：** | **B1303793** |
| **学 院：** | **软件学院** |
| **开题日期：** | **2015年 1月 14日** |

**上海交通大学研究生院制**

**填表日期： 2015年 1月 14日**

|  |  |
| --- | --- |
| **学位论文（设计）题目** | **基于微服务架构的智慧社区应用服务与资源管理系统** |
| **课题来源及类型** | **实验室横向项目** |
| **课题的意义及国内外现状分析：**  **课题的意义：**  智慧社区是指充分利用物联网、云计算、移动互联网等新一代信息技术的集成应用，为居民、居委会、业委会、物业和其他服务机构等社区相关人员提供的一种智慧化的生活环境和新的社区管理形态。随着智慧社区应用在社会进步与发展过程中作用的不断提升，其信息化运营与管理成本也在社会经济中产生了重要的影响。面对智慧社区中海量应用所造成的巨大的系统集成、运维成本，建立统一、高效的应用、资源管理方案是降低运营成本，提高服务质量的迫切需要。  本文以实验室承担的智慧邻里社区服务软件系统研发实际项目为背景，该系统建立了统一智慧社区应用平台，将广泛的社区应用，如居民生活服务、居委会工作服务、物业服务、党建服务、便民便利服务等进行整合，形成完整的社区服务闭环。同时为社区治理中的生活群众、政府管理群众、服务群众提供有价值的服务。例如，社区内的居民可以通过系统内邻里社交、物业、生活（外卖、洗衣、超市、家政等）、二手市场、社区公告等服务中获得生活便利；社区中政府管理群众可通过系统内党建、人事管理、社区公告、活动、邮件等服务中增加与居民沟通渠道，提升社区管理服务质量；社区内服务群众可通过系统内便民便利服务平台发布自身服务，增加自身营收。  然而， 在构建智慧社区治理系统过程中，遇到了以下的难题：1）应用服务间耦合严重，随着应用范围以及复杂度的增加，扩容以及更新流程复杂度大大增加；2）系统计算、存储、网络资源利用率低，没有一个很好的资源管理平台进行统一管理；3）缺乏统一的邻里社交模型，系统内用户交流存在壁垒。因此，如何建立统一、高校的应用、资源管理方案，成为一大问题。  **国内外现状分析：** 资源管理平台 在目前的复杂的企业解决方案中，所有的服务器按照业务逻辑被拆封成若干个集群，这些集群各自运行着相对独立的业务，并由专门的运维人员管理。  然而，由于集群中应用特点的不同，它们的资源利用率可能不一致。例如，同一时间段内有的集群业务繁忙，资源利用率高，甚至可能出现资源排队等待的情况，而另外一些集群则处于业务空闲，资源利用率极低的状态。此外，多个集群也可能导致数据冗余度增加，例如一个集群中的业务若需要另一集群中的数据，可能需要跨集群拷贝数据，进而占用更多的磁盘空间。因此，为了解决多集群所带来的问题，如Borg[4]（Google）， Yarn[1]（Apache，Hadoop项目下一个分支，开源），Mesos[2]（Twitter，开源），Corona[5]（Facebook，开源）这一类统一资源管理系统相继出现。概括起来，这类系统主要解决以下三类问题：  1）提高集群资源利用率  在大数据时代，为了存储和处理海量数据，需要规模较大的服务器集群或者数据中心，一般说来，这些集群上运行着数量众多类型纷杂的应用程序和服务，比如离线作业，流式作业，迭代式作业，crawler server，web server等，传统的做法是，每种类型的作业或者服务对应一个单独的集群，以避免相互干扰。这样，集群被分割成数量众多的小集群，一部分集群运行Hadoop[25]，一部分运行Storm[10]，一部分运行Spark[11]，一部分运行web server，然而，由于不同类型的作业/服务需要的资源量不同，因此，这些小集群的利用率通常很不均衡，有的集群满负荷、资源紧张，而另外一些则长时间闲置、资源利用率极低，为了提高资源整体利用率，一种解决方案是将这些小集群合并成一个大集群，让它们共享这个大集群的资源，并由一个资源统一调度系统进行资源管理和分配，这就诞生了Borg，YARN，Mesos，Torca，Corona。从集群共享角度看，这类系统实际上将所有硬件资源抽象成一个台大型计算机，供所有用户使用。  2）服务自动化部署  一旦将所有计算资源抽象成一个“大型计算机”后，就会产生一个问题：各种不同的服务如何进行部署？同样，Borg/YARN/Mesos/Torca/Corona一类系统需要具备服务自动化部署的功能，需要系统能够支持从简单的web server到诸如Spark，Storm等复杂的分布式计算框架的自动部署，管理。因此，从服务部署的角度看，这类系统实际上是服务统一管理系统，这类系统提供服务资源申请，服务自动化部署，服务容错等动能。  3）动态调整切分资源，增强系统扩展性  由于各类服务特征的不同，例如离线服务大都在夜间启动运行，对资源需求高，而传统应用服务则在日间活跃，导致系统对资源的规划很难一次性准确或稳定不变。因此，需要此类系统能够支持对已部署服务资源的动态调整与切分。增强系统的扩展性。 服务治理 服务治理是分布式系统及面向服务系统架构的核心组成。客户端怎样知道存在于多台主机上的服务的IP以及端口即为服务治理在最初所面临的问题。然而，随着系统中服务的不断增多，在自动或手动扩容、主机失效、新服务部署的情况下，服务的地址变化将变得极其频繁。此时，为了避免服务终端，动态服务注册与发现重要性将大大增加。  在定位服务地址的问题上，有两个关键性问题：  1）服务注册：服务将自身地址注册到一个中央注册中心当中。通常它将注册自己的IP及端口，验证凭据、使用的协议、版本号或一些使用环境细节信息。  a）自注册模式：  自注册模式意味着服务实例对自身的注册以及注销负责。同时，在必要的情况下需要发送心跳信息至服务中心防止其注册信息过期。  自注册服务拥有许多的优势与缺陷，其中一个优势在于它的相对简单性，且不依赖于其他的系统组件。然而，其中一个主要的缺陷在于它导致服务实例与服务中心耦合。必须针对每种变成语言或框架实现特定的代码。  b）第三方注册模式：  使用第三方注册模式将服务实例与注册中心解耦，服务实例不再负责自身的注册与注销。反之，一个称为服务注册器的系统组件被用以处理服务注册。服务注册器通过拉取运行信息或订阅时间的方式跟踪一组运行实例，当它发现一个新的可用服务时就将该服务注册至服务中心。与此同时，服务注册器也会处理服务注销的问题。 微服务软件架构风格 微服务是一种软件架构风格，它代表着一种开发多个运行在独立进程、环境中的使用轻量级通信协议进行交互的小型服务来代替传统的单一应用的大型应用的软件架构风格。这些微服务是围绕着业务功能进行划分的，并且可独立部署、升级、管理。  为了更好的解释微服务架构，我们将其与传统的单一应用架构进行比较。企业级应用通常包括三个部分：1）客户端接口（通常包括运行在客户浏览器中的HTML页面以及javascript代码），2）数据库（通常为关系型数据库中的多个表集合），3）服务器端应用（处理HTTP请求，执行业务逻辑，获取更新数据）。而服务器端应用通常是一个逻辑上的集合，对于服务器端的任何修改都会导致对整个服务器端应用的重新编译以及部署。单一应用架构是可行的，但随着应用范围以及复杂度的增加，其缺点也显得越来越明显：每次更改都要求整个服务器端重新编译以及部署，在扩容时也只能够对整个应用进行扩容，而不能够只针对资源需求更低的一个模块来进行扩容。  上述的问题就引出了一个新的软件架构：微服务架构。微服务架构将传统的单一应用以一组独立的服务来构成，与此同时，这些服务是可以单独部署以及水平扩展的。每一个这样的服务都拥有着严格的模块边界，甚至允许不同的服务使用不同的开发语言来进行开发。  微服务架构会使用依赖包，但其主要通过将软件分割成服务来进行模块化。我们定义依赖包为链接入程序并通过in-memory程序调用使用的程序模块，而服务为使用RPC或者网络请求进行交互的进程。使用服务作为系统模块有两大优势：首先服务可独立部署，这样对一个单独的模块进行更改不会影响到系统中的其他模块，另一个优势则是服务间通过显示的远程调用来进行通讯，从而减小程序间的接口定义难度。  然而，微服务架构也是有缺点的。首先，远程调用比进程内函数调用开销更大，其次，将程序职责在服务间进行迁移的难度也会更加的复杂。  去中心化治理：中心化治理的会导致整个程序开发使用单一的技术和平台。然而，这样的解决方案是受到很大限制的，不是每个问题都能通过单一的技术来解决。针对具体的问题选择最适合的技术显然是一个更加的做法。微服务架构则将这样的方法带入了软件开发之中。  去中心化数据管理：在对概念模型以及业务逻辑进行去中心化的同时，微服务架构同样对数据进行了去中心化的管理。每个微服务都独自管理自己的数据，如图2-11所示，这意味着微服务可根据具体的业务需求选择合适的数据库（关系型或非关系型），但同时也意味着在服务间数据一致性问题会更加难以处理。  测试部署自动化：随着云平台的日渐成熟，测试部署自动化在近年中发展迅速。而云平台大大减少了构建、部署、运行微服务的复杂度。  预防软件错误：微服务架构需要在设计时将服务异常纳入考虑范围，由于任何服务调用都可能因为服务提供者失效或网络失联而失败，因此客户端需要很好地处理这一类情况。  迭代式设计：当尝试将一个系统服务化时，我们将面临一个至关重要的问题：怎样将系统划分成不同的服务？应该使用怎样的原则来进行划分？由于一个服务的关键要素是独立部署升级，因此在系统划分时应该考虑在其需要重构时不会影响到其他的协作者。 | |

|  |
| --- |
| **课题研究目标、研究内容和拟解决的关键问题：**  **研究目标：**  本文以实验室承担的智慧邻里社区服务软件系统研发实际项目为背景，在对微服务系统架构、即时通讯系统、统一资源管理平台以及服务治理等相关技术进行深入分析研究的基础上，设计并实现了一个基于微服务架构的应用服务与资源管理系统。该系统在提供基于实名制的邻里社交系统的基础上，利用基于Dubbox的服务治理系统以及基于Apache Yarn的资源管理系统两大关键系统的结合解决了应用服务耦合严重以及平台资源利用率低的问题。并通过测试及试运行应用，以验证该系统的可行性及有效性。  **研究内容：**  围绕上述研究目标，本文的研究内容主要有：  1）实名制邻里社交系统：  a) 分析并建立基于实名制的邻里社交模型，对社区中的管理、生活、服务群众的社交模式进行抽象；  b) 建立即时通讯系统，支持点对点、群组等社交方式；  c) 实现用户状态管理服务，支持对用户或外接设备在线状态的管理；  d) 实现应用消息推送服务，支持聊天消息、应用广告等信息推送；  2) 实现基于Apache Yarn的支持“长服务”的统一资源管理系统：  a) 改进Yarn自身基于jobId的服务管理机制，通过服务发现注册机制扩展服务管理的功能及灵活性。解决由于节点迁移后导致的整体服务不可用的问题，并提供更多的协议支持；  b) 改进Yarn对于服务日志的管理机制，通过重写日志管理器及客户端API，支持在服务运行过程中对日志的管理，并支持日志管理策略的配置；  c) 实现Yarn中Container资源动态变更功能，使Yarn能够在不重启Container的情况下对Container资源进行变更，从而增加服务的可用性；  3) 基于微服务架构的社区管理系统：  a) 对智慧社区应用进行抽象划分，微服务化；  b) 基于服务治理框架Dubbox，实现服务发现、注册、监控调用统计、容量评估等功能，对服务进行管理。  c) 与统一资源管理系统集成，以服务监控信息为依据，按照一定策略对服务进行自动水平扩展。  4) 系统验证及测试：通过系统测试及试运行应用，验证该系统的可行性及有效性。 |
| **课题的研究方法、设计及试验方案，可行性分析（含经费预算及落实情况）：**  **研究方法：**  本课题的研究方法主要包括：  1. 大量阅读文献，了解各种不同的分布式计算框架的模型及业务需求，总结出最适合的调度策略  2. 研究Hadoop Yarn源码，并在其基础上对其进行改进，使其支持长服务  3. 实现服务治理系统，对服务进行监控、管理、容量评估、发现与注册。  4. 研究邻里社交模型以及MQTT通讯协议，实现邻里社交系统  **设计及实验方案：**  本研究将分为“邻里社交子系统”，“基于微服务的应用服务管理子系统”，“支持长服务的资源管理子系统”三个部分实现：   1. 邻里社交子系统：   分析并建立基于实名制的邻里社交模型，对社区中的管理、生活、服务群众的社交模式进行抽象；建立即时通讯系统，支持点对点、群组等社交方式；实现用户状态管理服务，支持对用户或外接设备在线状态的管理；实现应用消息推送服务，支持聊天消息、应用广告等信息推送。   1. 基于微服务的应用服务管理子系统   对智慧社区应用进行抽象划分，微服务化；基于服务治理框架Dubbox，实现服务发现、注册、监控调用统计、容量评估等功能，对服务进行管理；与统一资源管理系统集成，以服务监控信息为依据，按照一定策略对服务进行自动水平扩展。   1. 支持长服务的资源管理子系统：   改进Yarn自身基于jobId的服务管理机制，通过服务发现注册机制扩展服务管理的功能及灵活性。解决由于节点迁移后导致的整体服务不可用的问题，并提供更多的协议支持；改进Yarn对于服务日志的管理机制，通过重写日志管理器及客户端API，支持在服务运行过程中对日志的管理，并支持日志管理策略的配置；实现Yarn中Container资源动态变更功能，使Yarn能够在不重启Container的情况下对Container资源进行变更，从而增加服务的可用性；  **可行性分析：**   1. 项目来源：由于本研究基于智慧社区服务软件系统研发这一实际项目，需求数据均由甲方提供，保证了真实性和有效性。 2. 平台基础：目前已有比较成熟的开源资源服务管理平台Yarn以及服务治理平台Dubbox，可以在其上进行优化改进。 3. 经费可行性：本项目成果的测试环境由实验室内部提供，不需要额外的资源，因此从经费上来说是可行的。 4. 进度可行性：整个项目有完整的安排，从调研到开发到部署，将按照开发计划有条不紊地进行，因此从进度上来说是可行的。   综上所述，本课题是可行的。 | |
| **课题计划进度和预期成果：**  **计划进度：**   * 2014.1 – 2014.3查阅国内外文献，研究并学习技术，开题； * 2014.4 – 2014.5 设计并实现邻里社交子系统 * 2014.6 – 2014.7设计并实现支持长服务的资源管理子系统 * 2014.8 – 2014.9 设计并实现基于微服务的应用服务管理子系统 * 2014.10 – 2014.11完成整体项目集成，进行整体评估测试 * 2014.12 撰写学位论文。   **预期成果：**   * 基于微服务架构的智慧社区应用服务与资源管理系统 * 小论文（学术论文）一篇 * 硕士学位论文一篇 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **课程**  **学习**  **情况** | **已修学位课学分** | **19** | **待修学位课学分** | **0** |
| **已修学位课平均级点** | **2.2** | **已修课程总学分** | **32** |
| **导师Ⅰ意见** | 李会一同学在开题过程中，工作认真踏实，查阅了大量文献，并进行了综述。在此基础上识别出关键问题和技术，提出了合理可行的研究方案和研究计划。故同意递交开题答辩。  **签名： 日期：2015年1月14日** | | | |
| **导师Ⅱ意见** | 李会一同学能针对实际问题，开展研究，识别出关键问题，并提出了合理的技术方案，同意递交开题答辩。  **签名： 日期：2015 年1月14日** | | | |
| **开**  **题**  **报**  **告**  **专**  **家**  **组**  **意**  **见** | **开题报告时间： 2015年1月14日**  **参 加 人 数 ： 教师及专家 人； 研究生 人。**  **审 查 结 果 ： □ 同 意； □ 不 同 意。**  **（ 同意 人； 不同意 人 ）**  **专家组组长签名： 日期： 年 月 日** | | | |
| **备注** |  | | | |