**正文：**参照以下提纲撰写，要求内容翔实、清晰，层次分明，标题突出。**请勿删除或改动下述提纲标题及括号中的文字。**

**（一）立项依据与研究内容**（4000-8000字）：

1．**项目的立项依据**（研究意义、国内外研究现状及发展动态分析，需结合科学研究发展趋势来论述科学意义；或结合国民经济和社会发展中迫切需要解决的关键科技问题来论述其应用前景。附主要参考文献目录）；

**1.1 依托云计算模式，大量具有大数据、实时处理、连续流式等特征的应用场景涌现，分布式缓存技术及服务成为必要支撑，为构建高时效、可扩展的大数据应用提供了切入点，也带来新的技术挑战。**

随着信息技术网络化、泛在化、智能化的发展，“人-机-物”深度融合成为趋势，数据规模大、来源丰富、类型复杂、变化迅速等诸多特征，使得云计算成为大数据处理的主要基础设施。云计算环境下，高时效、可扩展的数据处理和访问服务不仅仅是保障用户服务质量的必要条件，也是维持经济、社会正常运行的重要因素。当前，**依托云计算模式，涌现出以智慧城市、智能装备、智能制造、极端交易等为代表的具有大数据、实时处理、连续流式等特征的数据处理和访问场景**[1][2]。例如：北京市6.67万台出租车每分钟将产生超过10万条实时GPS位置记录；大型城市中，百万规模的智能电表每分钟产生超过100万条用电记录；波音737飞机引擎在飞行状态下每小时产生20TB数据；“双11”期间支付宝每分钟交易量超过数百万笔。这些场景具有具有数据产生速度快、整体时延要求苛刻、特定场景下服务可用性和数据一致性要求苛刻等特点[3][4]。

工业界和学术界的早期工作主要针对传统集中式的数据库环境开展数据访问的低时延化以及数据处理的流式化，如实时数据库（real-time database）、内存数据库（in-memory database）。由于新时期的数据密集型应用呈现出大数据、实时处理、连续流式相结合的特征，对系统提出了新的更高要求，大量新型分布式系统架构及技术相继出现（例如，面向极端事务处理的GigaSpaces[5]、面向交互式大数据分析的Spark Streaming[6]、面向流式数据处理的Storm[7]等），**其共同特点是基于分布式缓存框架进行数据管理，即将数据分布到多个内存数据节点进行实时处理和访问**。分布式缓存管理的数据通常包括：1）**原始数据及元数据**：在分布式集群环境下，内存容量瓶颈得到缓解，可以针对部分甚至全部原始数据进行处理，其数据项（record）及元数据（meta data）通常以内存数据对象的形式存在。2）**查询缓存**：用于缓存用户的查询结果（query result）。3）**中间数据**：数据处理过程中产生的中间结果数据（intermediate data）。4）**状态数据**：用于系统恢复或扩展时的全局、局部状态数据（state）。

与传统的本地缓存、内存数据库等方案不同，分布式缓存框架兼具了NoSQL和内存数据库的特点，主要采用键值的存储类型，希望通过分布式的内存数据模型在一致性、可用性和分区容忍性之间获得更好的平衡，满足大数据、高并发场景的低时延数据处理和访问需求。分布式缓存框架通常具有如下特性[8]：1）高性能：将高速内存作为数据对象的存储介质，提供Map、List、Queue等分布式数据结构。2）动态扩展性：通过动态增加或减少节点应对变化的数据处理和访问负载，提供可预测的性能与扩展性。3）高可用性：提供数据副本，自动实施故障切换，不会因节点失效导致数据丢失或服务中断，并且在节点动态扩展时自动均衡数据分区，保障服务持续可用。4）易用性：提供单一的数据与管理视图，API接口简单，且与拓扑结构无关；动态扩展或失效恢复时无需人工配置；自动选取备份节点。5）分布式计算：支持Map/reduce等分布式计算编程模型。

从云计算的三层服务模式（IaaS, PaaS, SaaS）角度分析，分布式缓存服务作为一种新型的分布式内存数据管理中间件，主要运行于PaaS层。作为PaaS层的重要云服务，分布式缓存服务在高时效、可扩展的数据管理方面发挥着日益突出的作用[9][10]。Gartner的2015-2016年最新调查报告显示[11][12][13]，一方面，近年来分布式缓存框架系统持续涌现，例如：Oracle Coherence、IBM WebSphere eXtreme Scale、GigaSpaces eXtreme Application Platform、VMware GemFire、Red Hat Infinipan、Hazelcast、GridGain、Apache Ignite、Memcached等；另一方面，基于云的分布式缓存模式已经初具规模，国内外主流的云计算平台均提供了分布式缓存服务，相关系统提供商也纷纷推出面向云计算环境的服务镜像，为用户提供个性化、可定制的内存计算服务。

与此同时，**由于内存计算服务化、租户资源共享化、存储体系混合化、系统环境虚拟化等现状与趋势，在数据缓存模型、租户性能隔离、混合体系优化、资源弹性扩展等方面也给应用系统的开发、运行和管理带来新的技术需求和挑战。**

（1）现有的分布式缓存服务主要基于NoSQL数据模型，提供键值对操作以及简单的查询支持，虽然提供了良好的可扩展性，但是由于缺乏对传统关系型数据库开发模式的兼容（尤其缺乏对结构化查询语言的支持），以及较弱的一致性保证，使其应用范围受到了一定的限制。在此基础上，开发人员仅能实现简单的数据查询逻辑，并且需要在应用层保证数据一致性，不仅大幅增加了开发人员使用缓存服务的学习负担，抬高了应用部署成本。例如，针对TPC-W事务处理基准应用系统，采用现有分布式缓存方案，需要对系统进行大规模重构、甚至放弃一些复杂查询事务的支持[14]。本项目的前期预研结果也验证了这一问题，我们分别使用2款开源分布式缓存框架（Red Hat Infinispan和Hazelcast）对TPC-W基准应用进行重构，每个方案的累计代码修改量均超过了2000行，占原系统总代码量超过40%[15]。另据报道，我国12306网站采用国外分布式缓存系统VMWare GemFire、历时1年才实现余票查询和订单查询两项业务的改造[16]。**因此，如何提供高时效、易使用的数据缓存模型，降低内存计算的使用“门槛”以及应用系统的开发成本，成为大数据场景下分布式缓存服务面临的首要问题。**

（2）多租户共享是云计算模式的基本特征[17]，其基本思想是，宿主在某一服务层上的一个或多个服务实例为来自不同组织（租户）的并发用户提供服务，实现资源的共享使用，目的是提高资源利用率，降低均摊在每个租户上的基础设施与管理成本。在多租户模式下，尤其是PaaS平台广泛采用的应用实例-租户模式而言，单一内存计算服务实例（进程）同时被多个租户共享使用，尽管资源共享程度更高，但引入的资源竞争会导致部分租户的服务水平协议（SLA）无法得到满足，即所谓的性能干扰（performance interference）[18]。例如，对于数据缓存服务而言，当采用传统的数据缓存替换策略（如LRU、LFU），易导致请求内容宽泛、访问频率高的租户大量占用共享内存并获得额外的收益，而请求内容集中、访问频率低的租户则由于内存空间被挤占导致命中率下降，服务质量和利益受到损害；另一方面，不同的租户访问量、读写比、请求数据大小、缓存命中率、数据持久化策略、数据副本策略等差异同样也会带来CPU、网络、存储等资源的公平性问题。现有缓存服务未提供面向SLA保障的性能隔离机制，而采用“尽力而为（best-effort）”的策略，使得租户应用的性能变得难以预测[19][20][21]。**因此，如何针对高度共享的系统资源环境，在最大限度地提高资源利用率基础上，提供面向SLA保障的缓存服务性能隔离机制，成为当前研究的难点。**

（3）混合存储是一种集主存（DRAM）、固态盘（SSD）和机械硬盘（HDD）为一体，以大容量、高性能和低成本为目标的异构性存储应用模式。在混合存储环境下，通常可以将SSD替换机械硬盘来减少数据读取延迟，也可将其作为内存扩展来获得更大的内存操作和运行空间。当前，混合存储已成为云计算、大数据应用场景下的主要架构模式[22]。例如 Amazon EC2、Microsoft Azure、阿里云等主流的云计算厂商都提供了基于混合存储的虚拟机存储方案。面向混合存储体系的优化代表未来发展趋势，然而“混合存储感知”的技术相对存储无关的技术而言更加复杂，相关工作仍处于起步阶段，主要关注文件存储系统[23][24]、数据库管理系统[25][26]的混合存储优化，对于分布式缓存系统则鲜有相关研究工作，导致无法充分发挥新型硬件带来的诸多优势。例如，混合存储没有考虑到不同应用组件对存储的不同需求，从而无法差异化地提供存储性能保证，我们的前期项目预研及实验数据显示这一问题将最终导致数据处理或访问的延迟增加数倍；另外，当前静态的数据持久化机制无法感知负载的时变性和突发性，进而难以保证关键数据（元数据、中间数据、状态信息等）能够被及时存储。上述问题将对缓存服务的时效性和可扩展性带来严重影响。**因此，如何针对云计算环境下的混合存储体系，综合考虑大数据应用的存储需求差异以及负载的动态变化，提供应用敏感的混合存储体系优化机制，成为当前研究的热点。**

（4）云计算的弹性资源供给使得分布式缓存服务需要在满足用户SLA的基础上，提供最大限度的可伸缩性。缓存服务动态伸缩的同时，存储在缓存节点中的部分内存数据也需要在节点间完成迁移，因此，数据迁移是实现分布式数据管理系统动态扩展与弹性负载均衡的关键技术[27]。对于部署在云环境中的缓存服务而言，数据迁移过程中伴随的大量系统开销会给服务性能、可用性带来影响，如何有效降低迁移开销是需着力解决的关键问题。然而，虚拟化的软硬件环境给数据迁移带来新的挑战[28]。例如，虚拟机（VM）性能干扰和迁移节点的物理位置会对数据迁移产生无法忽略的影响[29]。已有研究工作主要针对磁盘I/O、网络I/O、CPU等资源敏感应用的虚拟机性能干扰对影响开展研究[30][31][32]，这些方法为考虑缓存服务用户的低延迟要求、访问负载的不可预知性以及系统的有状态性给数据迁移带来新的挑战。我们的前期项目预研及实验数据显示，VM性能干扰对数据迁移（包括迁移时间和性能衰减值）具有显著影响，随着VM数量的增长，迁移时间由于I/O阻塞呈非线性变化，性能衰减程度数倍于基准环境。**因此，如何在考虑虚拟化环境、用户负载变化等影响的前提下制订数据迁移规划，降低迁移开销是云模式下缓存服务弹性扩展的关键问题。**

综上分析，内存计算服务化、租户资源共享化、存储体系混合化、系统环境虚拟化等现状与趋势，要求我们以全新视野研究高时效、可扩展大数据应用场景下分布式缓存服务面临的新挑战和新问题，这也是本项目与国内外该领域的其他研究问题的核心区别。

**1.2 面向高时效、可扩展大数据应用的分布式缓存服务研究仍处于起步阶段，在数据缓存模型、租户性能隔离、混合体系优化、资源弹性扩展等方面尚未提供充分的技术支持。**

近年来，分布式内存计算成为国际分布式计算领域最为活跃的研究方向之一，大量相关研究工作对本项目具有重要借鉴意义。下面从数据缓存模型、租户性能隔离、混合体系优化、资源弹性扩展等方面深入介绍和分析该领域的最新进展。

**1）数据缓存模型研究**

典型的数据缓存模型可分为用于管理应用层缓存数据和管理中间层缓存数据（位于应用层和持久化存储层之间）两类。如表1所示，在不同层面，缓存模型均需要解决以下几个方面的问题：1）在内存中存储什么数据；2）提供何种编程模型；3）如何维护数据一致性。

（a）应用层缓存，允许开发人员管理粒度合适的应用层数据对象，如元组（tuple）、事件（event）、查询结果（query result）、会话（session）等，这些数据通常会存储在应用服务器进程中，因此应用逻辑和缓存数据之间的访问延迟最小。应用层缓存提供类似MapReduce的数据处理接口以及类似key/value的数据访问接口，开发人员可以实现特定的数据处理和查询逻辑，但同时也导致缓存数据管理逻辑与业务组件逻辑耦合在一起。另外，开发人员需要通过超时、失效、主动更新等方法，维护数据一致性。例如，Pequod等人[33]利用有序的键值对存储来缓存联合查询（join query），并提供了自动化的增量更新机制，但是开发人员仍然使用特定语法定义需要缓存的目标数据以及相应的查询视图和查询计划。连续查询是应用层缓存的一类特殊场景，主要针对连续流式数据提供实时查询支持，采用“先计算、再存储”方式对数据流进行切分和缓存，在局部有限数据范围内进行查询。该类缓存系统主要通过特定API由用户自定义的数据处理及查询逻辑（例如提供类MapReduce接口），也有部分系统提供了类SQL的应用编程接口，便于查询功能的实现[34]。由于需要权衡系统响应性和资源容量，该类缓存采用滑动窗口（slide window）机制，在指定范围（时间或数据量）的数据集合内，根据指定时间间隔，周期性的更新查询结果。在此基础上，一些研究工作从增量计算角度进行时效性优化[35][36][37]。上述工作的不足在于增加了开发人员的学习成本、开发代价，也增加了应用程序的复杂度、容易出错，并降低了业务组件复用的可能。

（b）中间层缓存，位于应用层和持久化存储层之间，保存部分或全部持久化层数据，从而降低持久化存储端服务器的负载压力。在该类模式中，缓存系统需要负责决定保存哪些数据，并维护与持久化数据的一致性。早期的代表工作包括TimesTen[38]、DBCache[39]、MTCache[40]等数据库缓存系统，这类工作存在三方面的限制：首先，由于缓存的数据对象是数据库表而不是应用层对象，因此与应用层数据缓存相比存在较大的性能开销。其次，方法将数据缓存在数据库进程，增加了进程间通信和数据转换的代价；最后，方法通常通过同步机制来保障数据一致性，因此所有的缓存数据更新操作都将在缓存层和数据库层同步，这将导致方法不适用于大规模的数据写入操作[41]。

查询结果缓存是一种简单的中间层缓存。该模式主要针对高频事务处理场景，将数据库查询结果缓存在数据库访问驱动层（例如JDBC）或对象关系映射框架（例如Hibernate），并依赖触发器机制[42][43][44]实现缓存数据的自动维护。当读写操作的并发量增加，缓存数据维护将产生大量的并发更新，由此产生数据竞争会导致出现过期数据。针对这一问题，已有工作主要通过设计相应的编程模型与一致性保障机制来满足特定的数据一致性保障要求。例如，Facebook[45]借鉴CPU级别的无锁同步指令Load-Link/Store-Conditional原理，提出基于租约（lease）的缓存数据并发维护机制，这一机制的灵活性在于可以通过在一定程度上允许读取到过期数据来获得高性能表现；Shahram等人[46]对租约机制进行扩展，为开发人员提供更多的租约操作语义，实现了强一致性（strong consistency）保障。上述工作不足在于方法主要针对只读（read only）[47]或读操作为主（workload dominated by reads）的应用场景[27]，不适用于大规模的数据写入操作[41]，并且需要显示调用特定的一致性维护接口来避免产生过期数据，另外，上述工作的系统实现需要依赖数据库触发器，甚至修改数据库引擎（如文献[44]）。

表1 典型数据缓存模型对比

|  |  |  |  |
| --- | --- | --- | --- |
| **特点** | **应用层缓存** | **中间层缓存** | |
| **缓存粒度** | 任意 | 数据库表 | 查询语句 |
| **应用场景** | 任意 | 数据库缓存 | 查询结果缓存 |
| **编程模型** | 键值对；类MapReduce；类SQL | SQL | SQL |
| **数据一致性** | 超时；人工失效；自定义更新逻辑； | 超时；自动失效/更新 | 超时；自动失效/更新 |

综上分析：应用层缓存模型属于“缓存分离”模型（cahce-aside），依赖应用程序的开发人员管理缓存数据。中间层缓存模型尽管不需要开发人员修改应用层代码，但适用场景有限（仅适用于“读多写少”的应用场景）；而对于查询结果缓存，为了避免缓存数据维护所导致的竞争，开发人员仍需使用特定的一致性维护接口来避免产生过期数据；另外，大部分工作的实现需要依赖数据库触发器，甚至修改数据库引擎。为此，我们需要综合上述方法的优势，研究更具时效性、易使用的大数据应用缓存模型。

**2）租户性能隔离研究**

国内外主流的PaaS平台如Amazon、Windows Azure、Sina App Engine、Aliyun等均提供了分布式缓存服务。对这些服务而言，每个应用系统对应一个租户。如表2所示，其多租户模型按共享程度和隔离性划分主要有三种：VM-租户模式、进程-租户模式和应用实例-租户模式[8]。在VM-租户模式中，服务器硬件被划分为若干VM实例，每个VM实例运行一个内存数据节点，每个节点承载一个租户应用的部分或全部数据，典型的如Amazon ElastiCache。在进程-租户模式中，每个VM实例运行多个节点进程，每个节点为一个租户应用服务，典型的如Gigaspaces[5]。在应用实例-租户模式中，每个VM实例运行一个节点进程，该节点承载了多个租户应用的内存数据，例如，Couchbase基于相互隔离的数据桶（bucket），允许租户配置不同的端口采用授权机制访问各自的数据[48]。

表2 三类常用的多租户模式

|  |  |  |
| --- | --- | --- |
| **多租户模式** | **描述** | **共享程度** |
| **VM虚拟机-租户** | 服务器硬件被VM虚拟机划分为多个基础设施实例，各运行一个平台实例，每个平台实例部署一个应用实例，每个应用实例为一个租户服务。 | 低：共享硬件基础设施 |
| **进程-租户** | 服务器硬件作为一个基础设施实例，其上运行多个平台实例，每个平台实例部署一个应用实例，每个应用实例为一个租户服务。 | 中：共享操作系统 |
| **应用实例-租户** | 服务器硬件作为一个基础设施实例，其上运行一个平台实例，部署多个应用实例，每个应用实例为一个租户服务。 | 高：共享进程 |

相比前两种租户模式，应用实例-租户模式采用线程级的资源管理机制，共享程度高，但难以精确隔离和有效调度底层物理资源，租户间性能隔离面临更多挑战。现有研究工作主要集中在各类数据缓存的性能隔离，这些数据缓存或是作为独立的云服务存在，或是作为持久化存储的数据缓存模块，例如关系型数据库的共享缓冲池（buffer pool）、非关系型数据存储的块缓存（block cache）、共享存储服务的存储缓存（storge cache）等，这些工作可分为基于改进的全局数据缓存替换的方法、基于动态内存资源分配的方法和基于请求调度的方法。

在缓存替换策略方面，现有工作主要针对传统静态缓存替换策略存在的弊端开展研究。静态缓存替换策略的基本思想是为每个租户分配声明的资源空间，并在其分配的空间内实施缓存数据替换，该策略可以最低限度的保证租户资源使用的公平性，有效防止租户间的内存侵占，然而这种固定的内存配置对负载变化缺乏足够的适应性，无法保障租户的响应时间。为此，IBM海法实验室的Chockler等人[49]利用全局数据缓存替换算法来保障租户内存资源使用的公平性，采用最小二乘法拟合租户性能曲线，并基于该性能曲线得到的预测结果为缓存替换算法提供执行依据。作者给出了两种替换策略，一是租户动态加权，以确保访问频率低的租户的数据不会很快被替换出内存；二是根据租户当前占用缓存空间是否少于目标分配空间，来确定是选取该租户还是其他租户的数据替换出缓存。该方法的不足在于缓存替换策略的粒度较粗，调整周期长。Narasayya等人[50]针对微软Azure关系型数据库云服务的共享缓冲池（buffer pool）管理问题，将传统数据库常用的分页替换算法LRU-K升级为的面向多租户的分页替换算法MT-LRU。该方法综合考虑了租户数据的大小、分布以及负载特征，实现了SLA敏感的共享缓冲池管理。方法的不足在于场景限定在单个物理节点上的共享缓冲池，而未考虑由多个物理节点组成的分布式共享架构。

在内存资源分配方面，相关研究工作主要集中于共享存储系统的存储缓存管理，其基本思想是通过动态划分缓存资源来实现性能隔离。宾夕法尼亚州立大学的Prabhakar等人[51]针对分布式共享存储系统的租户缓存资源分配问题，使用I/O延迟指标表征租户的QoS需求，最终利用线性规划方法完成每个节点内部的逻辑资源划分。山东大学Shi等人[52]将分布式共享存储系统的存储缓存管理问题定义为收益目标优化问题，以服务提供商收益最大化为目标，结合租户SLA收益模型，利用遗传算法周期性的生成缓存资源分配策略。国内华中科技大学Feng等人[53]针对异构多租户应用场景下的共享存储缓存管理问题，提出一种基于分区的缓存管理算法，其基本思想是建立综合考虑租户应用性能和公平性的效用函数，通过预测动态调整缓存分区大小，使得在保障一定程度公平性的同时各租户应用的性能下降最小。上述工作不足在于内存划分时未考虑租户负载在节点间的分布，租户的内存配额与访问负载不匹配，降低了资源的使用效率。

在请求调度方面，相关研究工作同样集中于共享存储系统的存储缓存管理，其基本思想是将共享存储服务看作一个黑盒，租户请求按照分类进入到不同的缓存队列中，采用调度策略完成优先级调度和准入控制。滑铁卢大学的Akshay等人[54]提出一种基于性能模型的存储缓存管理方法，该方法扩展了传统单一的LRU队列，构建了请求时延期限敏感（deadline-aware）的多LRU队列，并根据性能模型对请求响应时延进行预估，进而将有限的缓存资源分配给潜在的超期响应请求。该方法的不足在于无法解决由租户资源竞争导致的服务质量下降问题。

综上分析，现有研究工作主要关注租户间内存资源的竞争，而未考虑CPU、网络、存储等其它系统资源竞争带来的影响，具有一定的局限性。另一方面，同内存资源竞争相比，其它系统资源竞争相关联的因素更为复杂（如读写比、命中率和并发量等），并且CPU资源由于多租户共享进程而难以精确度量，因此现有的内存资源分配方法无法直接应用于其它系统资源竞争场景。因此，现有工作难以满足高时效、可扩展的大数据应用场景下分布式缓存的性能隔离需求，与之相对应的是，基于持久化介质的大数据存储服务性能隔离研究则成为近期热点[19][20][55][56][57][58]，这为本项目开展相关研究提供了大量可借鉴的技术、算法和机制。为此，需要综合考虑分布式缓存的功能特性、租户负载变化以及CPU、网络、存储等资源的公平分配问题，提供面向SLA驱动的性能隔离机制。

**3）混合体系优化研究**

目前，国内外的主流云计算厂商（如Amazon EC2、Microsoft Azure、阿里云等）已提供了基于混合存储的虚拟机存储方案。然而，“混合存储感知”的系统优化技术相对存储无关的技术而言更加复杂，需要了解存储的层次、容量、带宽等指标，相关研究工作仍处于起步阶段，主要分为SSD分配优化和混合存储全局资源优化两方面。

SSD分配优化技术主要将SSD作为文件存储系统的缓存进行优化。Centaur[24]通过缓存分区（Cache Partitioning），为不同的虚拟机分配独立的、不同大小的缓存，从而获得相比使用统一缓存（所有虚拟机共享同一个SSD缓存）而言更高的缓存利用率和性能。S-CAVE[59]基于测试和历史数据分析的方法，根据缓存命中率及由缓存相关指标导出的数据，如rECS（有效缓存空间比例）、已分配缓存空间、早先的分配决策等，来决定缓存的分配。这些工作存在的局限性是它们都将虚拟机独立对待，针对不同的虚拟机分配缓存，没有考虑到应用的不同组件部署在多台虚拟机上的情形，也没有考虑到虚拟机之间的关联，从而不能很好地应对多虚拟机部署的大规模数据密集型应用。

在混合存储全局资源优化方面，相关研究工作主要考虑云计算环境下的全局I/O控制。IOFlow[60]针对不同虚拟机的I/O请求进行了标记，将虚拟机的I/O请求抽象为I/O流的形式，之后通过控制I/O流在网络上的传输以实现针对I/O请求的控制。Moirai [61]则对IOFlow进行了扩展，它将缓存作为IOFlow中I/O路径的路由目标，即可将指定的I/O请求导向至缓存，从而实现全局的缓存分配和调度，提升虚拟机的性能。这些工作存在的局限性是它们都从“软件定义网络”的角度来进行I/O的流控制，因此其“全局控制”及“控制与数据分离”的概念只覆盖了I/O栈在网络传输中的部分，即I/O请求在网络上的交互过程，没有考虑到I/O请求在具体的I/O栈中的调度。因此，当应用集群规模较大、拓扑复杂，且有多个应用集群存在时，仍需要大量的人工介入，为每台虚拟机定义特定规则，覆盖I/O栈的各方面，方可保证控制的准确性。

现有的混合存储体系优化工作主要面向“SSD + HDD”的混合模式，关注文件存储系统、数据库管理系统的混合存储优化，针对分布式缓存等内存计算系统的相关研究工作较少。对于分布式缓存服务，如何实现用户透明的系统优化、及时有效的将缓存系统中的关键数据（元数据、中间数据、状态信息等）持久化到非易失的混合存储介质，是提升分布式缓存数据存储管理的时效性和可靠性的关键。然而，现有面向大数据应用的分布式框架仍依赖传统存储架构，无法发挥新型硬件带来的诸多优势。Strom的Trident[62]允许节点状态以增量的方式存入远程数据库中；Samza平台[63]提供本地状态持久化策略并支持远程备份，该方案实现中间状态数据的存储管理。Hazelcast[64]为用户提供了数据持久化接口，该方案支持用户配置特定策略来实现缓存数据的同步或异步持久化。上述方案的不足在于存储管理策略简单、固化，无法应对缓存系统负载和系统状态的动态变化。

综上分析，现有云计算环境下的混合存储策略缺乏对虚机与虚机之间、组件与组件之间关联性的感知，缺乏对多租户异构应用I/O需求差异的考虑，缺乏对缓存系统负载和系统状态时变性的考虑，具有一定的局限性。为此，需要统筹考虑“DRAM + SSD + HDD”的混合存储体系架构特征：一方面，针对DRAM与SSD的“混合内存”，需要能够感知分布式缓存中关键数据产生速率变化以及系统资源状态变化，自适应的调整分布式缓存的数据存储管理策略；另一方面，针对SSD与HDD的存储系统优化，需要一种能够从整体上把握应用相关性，从全局视角感知应用的服务质量需求和优先级需求，同时统筹混合存储体系全局I/O路径的调度机制，实现合理的资源调度和优先级选择。因此，需要研究如何针对云计算环境下的混合存储体系，综合考虑大数据应用的存储需求差异以及负载的动态变化，提供应用敏感的混合存储体系优化机制。

**4）资源弹性扩展研究**

数据迁移是实现分布式系统节点动态扩展与负载均衡的关键技术，一些研究工作关注如何利用控制论、性能模型等方法实现数据迁移过程的服务质量保障。Lim等人[64]采用控制论方法解决数据迁移中网络I/O的控制问题。文献[65]主要针对具有严格服务质量约束的存储系统的数据迁移问题，作者首先采用回归方法构建一个线性分类模型，该模型可预测任意存储节点在某一负载下是否违背SLA约束，控制器基于该模型与各分区统计信息采用贪心方法制定数据迁移计划。Chiu等人[66]提出了一种基于贪心法的数据迁移策略。上述工作的不足在于仅关注数据迁移前后的系统状态，而未考虑迁移开销问题。针对迁移过程中产生的系统开销，Kari 等人[67]针对异构存储系统在满足不同传输能力约束前提下如何降低数据迁移时间这一问题开展研究。作者将该问题规约为图着色（Multi-edge coloring problem）问题，考虑通过优化数据迁移的调度，使得迁移消耗的时间片最少，即使用的颜色数量最少。Das等人[68][69]的工作主要针对多租户场景下数据库集群的数据迁移问题，作者提出了一种轻量级的、基于迭代复制的数据迁移方法，目标是尽可能降低迁移开销。迁移过程中使用一种类似两阶段提交协议（Two-Phase Commit）的机制保障数据操作的原子性。迭代复制时，源节点持续提供服务，复制结束后迁移数据的控制权转移至目标节点。

云计算模式下，虚拟化的软硬件环境给数据迁移带来新的挑战[28][29]。具体来讲，迁移数据的发送与接收是异步完成，并与虚拟机调度机制相关，而现有虚拟机调度算法在I/O调度处理方面存在一定缺陷，主要表现为调度时仅关注每个domain的状态，并一味追求资源分配的公平性而忽略每个domain I/O处理的实时性需求。为此，一些研究工作通过收集系统实时数据，对性能干扰问题进行检测和避免。Kambadur 等人[70]主要利用分布式环境中大量相似的任务并发运行，将这些相似任务的IPC（instructions per cycle）正态分布曲线作为性能干扰指标，当收集到的在线服务IPC过度偏离其正态分布曲线期望值时，则认为该服务出现了性能干扰。Novakovic等人[71]利用硬件计数器进行性能干扰检测，作者通过收集不同维度的硬件计数器参建立状态空间，利用聚类算法将应用的正常状态进行聚类，没有落入聚类的状态则被认定为干扰状态。Zhang等人[72]同样利用类似于IPC的CPI参数建立正态分布曲线，并且进一步通过对出现性能干扰的服务器上的应用CPI进行排序，找到最有可能干扰他人的应用进行资源限制。另一类研究工作通过建立性能干扰模型，预测性能干扰的影响。Kundu 等人[73]通过实验发现，随机I/O和顺序I/O对性能干扰带来不同程度的影响，进而细化了I/O参数输入，对比分析了机器学习方法中支持向量机模型和神经网络模型对性能干扰的建模效果。华中科技大学的Xu等人[74]针对虚拟机迁移问题，利用各种基准应用负载测试分析了迁移时和迁移后的两类性能干扰问题，通过设计多资源供需模型（demand-supply model）实现性能干扰的综合评估，在此基础上实现了虚拟机干扰敏感的迁移机制。本项目团队的前期研究工作[75]针对分布式缓存数据迁移的性能衰减和迁移时间进行权衡，建立了基于面积的迁移开销模型以及开销敏感的数据迁移算法。该工作的不足在于未考虑节点资源异构、突发性负载以及缓存服务的SLA约束等因素。

综上分析，现有工作的不足在于主要针对磁盘I/O、网络I/O、CPU等资源密集型应用的虚拟机性能干扰开展研究，这些方法未考虑高时效、可扩展的大数据应用场景下，缓存服务的低延迟、负载不可预知和时变性以及系统有状态性等给数据迁移带来新的挑战。同时，虚拟化环境下的数据迁移操作可能受网络因素的影响导致操作失效，服务中断时用户请求需要重定向，多用户并发操作数据的多个副本易产生数据一致性问题等，这些都是需要迫切解决的问题。此外，访问过程中产生的热点数据区同样会成为系统瓶颈，需要对热点数据进行重均衡处理，优化系统资源利用率。因此，需要研究如何刻画底层虚拟化环境的性能干扰对缓存数据迁移产生的影响，综合考虑迁移时间、性能衰减等多种因素，提供虚拟化敏感的弹性扩展机制。

**1.3 综合分析上述技术与挑战，迫切需要针对内存计算服务化、租户资源共享化、存储体系混合化、系统环境虚拟化的现状与趋势，建立面向高时效、可扩展大数据应用的分布式缓存服务技术体系，为应用系统的开发、运行和管理提供必要的关键技术支撑，并遵循现有技术标准开发出具有实际应用价值的支撑系统。**

本项目研究解决基于云计算模式的高时效、可扩展大数据应用场景下分布式缓存服务面临的新挑战和新问题，符合云计算、大数据应用环境下软件技术发展的迫切需求，具有重要理论技术价值和广阔应用前景。

从国家需求来看，随着信息技术网络化、泛在化、智能化的发展，信息化正在进入“人-机-物”深度融合阶段，着力关注下一代信息技术，力争抢占国际科技和产业竞争的战略制高点，对于提升我国科技和产业竞争力意义重大。《国家中长期科技发展规划纲要（2006-2020）》指出：“信息技术将继续向高性能、低成本、普适计算和智能化等主要方向发展，寻求新的计算与处理方式和物理实现是未来信息技术领域面临的重大挑战”。本项目关注的分布式缓存服务是云计算环境下提升应用性能的关键手段，是支撑智慧城市、物联网以及重要领域信息化系统场景下的极端事务处理、流式数据处理等数据密集型应用系统的基础软件技术，项目成果的应用方向和应用前景符合国家信息化建设需求，具有重大现实意义和长远战略意义。

在技术理论方面，本项目针对内存计算服务化、租户资源共享化、存储体系混合化、系统环境虚拟化的现状与趋势，建立面向高时效、可扩展大数据应用的分布式缓存服务技术体系。相关关键技术问题的提出符合云计算、大数据应用环境下软件技术发展的迫切需求，主要基于以下判断：

1）内存计算已成为云计算模式下的核心关键服务，如果失去分布式缓存这一关键技术的支撑，云计算的潜能将十分有限；

2）多租户共享是云平台服务的基本特征，高共享程度的多租户模式是技术发展趋势；

3）基于非易失闪存的混合内存体系将成为云计算平台的“标准配置”，分布式缓存等内存计算技术需要做出必要变化与之相适应；

4）现有虚拟化技术仍面临性能干扰难题，需要结合应用特征，在平台层统一协调处理。

在软件产业方面，IBM、Oracle、Microsoft、SAP、VMware、Red Hat等国外基础软件厂商正积极布局内存计算产品和技术的研制，并开始在中国市场积极推广，国产基础软件厂商难以跟上国外企业的技术产品研发节奏，跨代差距逐渐拉大。本项目的相关技术及系统成果可转化到国产基础软件企业，可为我国基础软件产业跨越式发展提供重要的技术推动力。

**参考文献：**

1. 程学旗, 靳小龙, 王元卓, 郭嘉丰, 张铁赢, 李国杰. 大数据系统和分析技术综述. 软件学报, 2014(9):1889-1908.
2. 孟小峰, 慈祥. 大数据管理:概念、技术与挑战. 计算机研究与发展, 2013, 50(1):146-169.
3. 孙大为, 张广艳, 郑纬民. 大数据流式计算:关键技术及系统实例. 软件学报, 2014, 25(4):839-862.
4. 崔星灿, 禹晓辉, 刘洋, 吕朝阳. 分布式流处理技术综述. 计算机研究与发展, 2015, 52(2):318-332.
5. Gigaspaces. XAP In-Memory Computing Platform. http://www.gigaspaces.com
6. Apache Spark Streaming. http://spark.apache.org/streaming
7. Apache Storm. http://storm.apache.org
8. 秦秀磊, 张文博, 魏峻, 王伟, 钟华, 黄涛. 云计算环境下分布式缓存技术的现状与挑战. 软件学报, 2013, 24(1):50-66.
9. Shamsi J, Khojaye M A, Qasmi M A. Data-Intensive Cloud Computing: Requirements, Expectations, Challenges, and Solutions. Journal of Grid Computing, 2013, 11(2):281-310.
10. Ranjan R. Streaming Big Data Processing in Datacenter Clouds. IEEE Cloud Computing, 2014, 1(1):78-83.
11. Massimo Pezzini, Roxane Edjlali, Nick Heudecker. Gartner: Cool Vendors in In-Memory Computing Technologies, 2015, Report.
12. Fabrizio Biscotti, Massimo Pezzini. Gartner: Hype Cycle for In-Memory Computing Technology, 2015, Report.
13. Rick Greenwald, Eric Knipp. Gartner: Architect Cloud-Native Solutions With In-Memory Computing, 2016, Report.
14. Donald Kossmann, Tim Kraska, Simon Loesing, An evaluation of alternative architectures for transaction processing in the cloud, Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, June 06-10, 2010, Indianapolis, Indiana, USA.
15. ExBench4Q for NoSQL. Trustie Project. http://www.trustie.net/projects/project/show/ExBench4Q\_for\_NoSQL
16. 12306 & Gemfire. http://www.csdn.net/article/2015-02-10/2823900
17. Multitenancy. Wikipedia. http://en.wikipedia.org/wiki/Multitenancy
18. Shue D, Freedman MJ, Shaikh A. Performance isolation and fairness for multi-tenant cloud storage. In：Proc. of the 10th USENIX conference on Operating Systems Design and Implementation (OSDI '12). 2012. 349-362.
19. Lang W, Shankar S, Patel J M, et al. Towards multi-tenant performance slos. Knowledge and Data Engineering, IEEE Transactions on, 2014, 26(6): 1447-1463.
20. Mace J, Bodik P, Fonseca R, et al. Retro: Targeted resource management in multi-tenant distributed systems. 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15). 2015: 589-603.
21. Floratou A, Megiddo N, Potti N, et al. Adaptive Caching Algorithms for Big Data Systems. IBM Research Report. September 25, 2015.
22. Dong M, Lit H, Ota K, et al. HVSTO: Efficient privacy preserving hybrid storage in cloud data center. Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on. IEEE, 2014: 529-534.
23. Koller, R., et al., Write policies for host-side flash caches, in Proceedings of the 11th USENIX conference on File and Storage Technologies. 2013, USENIX Association: San Jose, CA. p. 45-58.
24. Koller, R., A.J. Mashtizadeh, and R. Rangaswami. Centaur: Host-Side SSD Caching for Storage Performance Control. in Autonomic Computing (ICAC), 2015 IEEE International Conference on. 2015.
25. Liu X, Salem K. Hybrid storage management for database systems. Proceedings of the Vldb Endowment, 2013, 6(8):541-552.
26. Jin P, Yang P, Yue L. Optimizing B+-tree for hybrid storage systems. Distributed & Parallel Databases, 2015, 33(3):1-27.
27. Trushkowsky B, Bodik P, Fox A. The SCADS Director: Scaling a Distributed Storage System under Stringent Performance Requirements. In：Proc. of USENIX Conf. on File and Storage Technologies (FAST '11). 2011. 163-176.
28. Xu F, Liu F, Jin H, et al. Managing Performance Overhead of Virtual Machines in Cloud Computing: A Survey, State of the Art, and Future Direction. Proceedings of the IEEE, 2014, 102(1):11-31.
29. Mei Y, Liu L, Pu X, Sivathanu S, Dong X. Performance analysis of network I/O workloads in virtualized data centers. IEEE Trans. on Service Computing, 6(1):48-63, 2013.
30. Xu F, Liu F, Liu L, et al. iAware: Making Live Migration of Virtual Machines Interference-Aware in the Cloud. IEEE Transactions on Computers, 2014, 63(12):3012-3025.
31. Kundu S, Rangaswami R, Gulati A, et al. Modeling Virtualized Applications using Machine Learning Techniques. Acm Sigplan Notices, 2012, 47(7):3-14.
32. Zhang X, Tune E, Hagmann R, et al. CPI 2 : CPU performance isolation for shared compute clusters. ACM European Conference on Computer Systems. 2013:379-391.
33. Kate B, Kohler E, Kester M S, et al. Easy freshness with Pequod cache joins. 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14). 2014: 415-428.
34. Hoppe A, Gryz J. Stream processing in a relational database: A case study. In: Proc. of the 11th Int’l Database Engineering and Applications Symp. (IDEAS 2007). Banff: IEEE Press, 2007. 216−224.
35. Tangwongsan K, Hirzel M, Schneider S, et al. General incremental sliding-window aggregation. Proceedings of the Vldb Endowment, 2015, 8(7):702-713.
36. Pramod Bhatotia, Umut A. Acar, Flavio P. Junqueira, and Rodrigo Rodrigues. Slider: incremental sliding window analytics. In Proceedings of the 15th International Middleware Conference (Middleware '14). ACM, New York, NY, USA, 2014, 61-72.
37. Zaharia M, Das T, Li H, et al. Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters. Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing. USENIX Association, 2012:10-10.
38. The TimesTen Team. Mid-tier caching: The TimesTen approach. In Proc. of the SIGMOD, Madison, WI, 2002.
39. Bornh¨ovd, C., Altinel, M., Mohan, C., Pirahesh, H., Reinwald, B. 2004. Adaptive database caching with DBCache. IEEE Data Eng. Bull. 27(2), 11–18.
40. Larson, P.-A, Goldstein, J., Zhou, J. 2004. MTCache: transparent mid-tier database caching in SQL server. In Proceedings of the 20th International Conference on Data Engineering. ICDE '04, Boston, MA, 177-188.
41. Kristian F. D. Rietveld and Harry A. G. Wijshoff. 2013. To cache or not to cache: a trade-off analysis for locally cached database systems. In Proceedings of the ACM International Conference on Computing Frontiers. CF '13, ACM, New York, NY, USA, Article 31, 8 pages.
42. S. Ghandeharizadeh, J. Yap, and S. Barahmand. COSAR-CQN: An Application Transparent Approach to Cache Consistency. In Proceedings of the 21st International Conference on Software Engineering and Data Engineering, LA, California, June 27-29 2012.
43. Gupta, P., Zeldovich, N., Madden, M. A trigger-based middleware cache for ORMs. In Proceedings of the 12th ACM/IFIP/USENIX international conference on Middleware. Middleware'11, Fabio Kon and Anne-Marie Kermarrec (Eds.). Springer-Verlag, Berlin, Heidelberg, 2011, 329-349.
44. Ports D R K, Clements A T, Zhang I, et al. Transactional Consistency and Automatic Management in an Application Data Cache. OSDI. 2010, 10: 1-15.
45. R. Nishtala, H. Fugal, S. Grimm, M. Kwiatkowski, H. Lee, H. C. Li, R. McElroy, M. Paleczny, D. Peek, P. Saab, D. Staord, T. Tung, and V. Venkataramani. Scaling Memcache at Facebook. In NSDI, pages 385-398, Berkeley, CA, 2013. USENIX.
46. Ghandeharizadeh S, Yap J, Nguyen H. Strong consistency in cache augmented SQL systems. Proceedings of the 15th International Middleware Conference. ACM, 2014: 181-192.
47. Eyal I, Birman K, van Renesse R. Cache serializability: Reducing inconsistency in edge transactions. Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on. IEEE, 2015: 686-695
48. Couchbase. http://www.couchbase.com
49. Chockler G, Laden G, Vigfusson Y. Data caching as a cloud service. In：Proc. of the 4th Int’l Workshop on Large Scale Distributed Systems and Middleware (LADIS '10). 2010.
50. Narasayya V, Menache I, Singh M, et al. Sharing buffer pool memory in multi-tenant relational database-as-a-service. Proceedings of the VLDB Endowment, 2015, 8(7): 726-737.
51. Prabhakar R, Srikantaiah S, Kandemir M, Patrick CM, Kandemir M. Adaptive Multi-Level Cache Allocation in Distributed Storage Architectures. In：Proc. of the 24th ACM Int’l Conf. of Supercomputing (ICS '10). 2010. 211-221.
52. 史玉良, 王捷. 一种多租户云数据存储缓存管理机制. 计算机研究与发展, 2014, 51(11):2528-2537.
53. Li Y, Feng D, Shi Z. Heterogeneous-aware cache partitioning: Improving the fairness of shared storage cache. Parallel Computing, 2014, 40(10): 710-721.
54. Singh A K, Cui X, Cassell B, et al. MicroFuge: A middleware approach to providing performance isolation in cloud storage systems[C]//Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on. IEEE, 2014: 503-513.
55. Luo Y, Zhou S, Guan J. LAYER: a cost-efficient mechanism to support multi-tenant database as a service in cloud. Journal of Systems and Software, 2015, 101: 86-96.
56. Zeng J, Plale B. Workload-aware resource reservation for multi-tenant nosql. Cluster Computing (CLUSTER), 2015 IEEE International Conference on. IEEE, 2015: 32-41.
57. Zhang N, Tatemura J, Patel J, et al. Re-evaluating designs for multi-tenant OLTP workloads on SSD-basedI/O subsystems. Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM, 2014: 1383-1394.
58. Das S, Narasayya V R, Li F, et al. CPU sharing techniques for performance isolation in multi-tenant relational database-as-a-service. Proceedings of the VLDB Endowment, 2013, 7(1): 37-48.
59. Luo, T., et al., S-CAVE: effective SSD caching to improve virtual machine storage performance, in Proceedings of the 22nd international conference on Parallel architectures and compilation techniques. 2013, IEEE Press: Edinburgh, Scotland, UK. p. 103-112.
60. Thereska, E., et al., IOFlow: a software-defined storage architecture, in Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles. 2013, ACM: Farminton, Pennsylvania. p. 182-196.
61. Stefanovici, I., et al., Software-defined caching: managing caches in multi-tenant data centers, in Proceedings of the Sixth ACM Symposium on Cloud Computing. 2015, ACM: Kohala Coast, Hawaii. p. 174-181.
62. Storm Trident. http://storm.apache.org/documentation/Trident-tutorial.html
63. Apache Samza. http://samza.incubator.apache.org
64. Lim HC, Babu S, Chase JS. Automated control for elastic storage. In：Proc. of the 7th Int’l Conf. on Autonomic computing (ICAC '10). 2010. 1-10.
65. Trushkowsky B, Bodik P, Fox A. The SCADS Director: Scaling a Distributed Storage System under Stringent Performance Requirements. In：Proc. of USENIX Conf. on File and Storage Technologies (FAST '11). 2011. 163-176
66. Chiu D, Shetty A, Agrawal G. Elastic Cloud Caches for Accelerating Service-Oriented Computations. In：Proc. of the ACM/IEEE Int’l Conf. for High Performance Computing, Networking, Storage and Analysis (SC '10). 2010. 1-11.
67. Kari C, Kim Y, Russell A. Data Migration in Heterogeneous Storage Systems. In：Proc. of the 31st Int’l Conf. on Distributed Computing Systems (ICDCS '11). 2011. 153-160.
68. Das S, Nishimura S, Agrawal D, Abbadi AE. Albatross: Lightweight Elasticity in Shared Storage Databases for the Cloud using Live Data Migration. In：Proc. of the 37th Int’l Conf. on Very Large Data Bases (VLDB '11). 2011. 494-505.
69. Elmore AJ, Das S, Agrawal D, Abbadi AE. Zephyr: Live Migration in Shared Nothing Databases for Elastic Cloud Platforms. In：Proc. of the annual ACM SIGMOD Conference (SIGMOD '11). 2011. 301-312.
70. Kambadur M, Moseley T, Hank R, Kim MA. Measuring interference between live datacenter applications. In: Proc. of the Int’l Conf. on High Performance Computing, Networking, Storage and Analysis. ACM Press, 2012. 1−12.
71. Novaković D, Vasić N, Novaković S, et al. DeepDive: transparently identifying and managing performance interference in virtualized environments. Usenix Technical Conference. 2013.
72. Zhang X, Tune E, Hagmann R, et al. CPI2: CPU performance isolation for shared compute clusters. ACM European Conference on Computer Systems. 2013:379-391.

Kundu S, Rangaswami R, Gulati A, et al. Modeling Virtualized Applications using Machine Learning Techniques. Acm Sigplan Notices, 2012, 47(7):3-14.

1. Xu F, Liu F, Liu L, et al. iAware: Making Live Migration of Virtual Machines Interference-Aware in the Cloud. IEEE Transactions on Computers, 2014, 63(12):3012-3025.
2. Xiulei Qin, Wei Wang, Wenbo Zhang, Jun Wei, Xin Zhao, Hua Zhong, Tao Huang. PRESC2: efficient self-reconfiguration of cache strategies for elastic caching platforms, Computing, 2014, 96(5): 415-451.

2．**项目的研究内容、研究目标，以及拟解决的关键科学问题**（此部分为重点阐述内容）**；**

**2.1 研究目标**

本项目针对云计算、大数据应用环境下软件技术发展的迫切需求，研究建立面向高时效、可扩展大数据应用的分布式缓存服务技术体系，着力解决内存计算服务化、租户资源共享化、存储体系混合化、系统环境虚拟化的现状与趋势带来的新挑战和新问题。在本研究团队已有工作积累的基础上，项目预期达到以下研究目标：

1. 提出一种新型数据缓存模型，支持具有内外存融合特征的高时效、易使用的缓存数据访问，降低传统分布式缓存系统的使用学习成本和开发代价；
2. 提出一种SLA驱动的多租户性能隔离技术，针对高共享程度的多租户模式，在保障租户间性能隔离的同时，满足各租户相互独立的SLA需求；
3. 提出一种应用敏感的混合存储体系优化技术，针对云计算环境下的混合存储架构，感知大数据应用的存储需求差异以及负载的动态变化，实现时延敏感、自适应的分布式缓存数据存储管理；
4. 提出一种虚拟化环境敏感的资源弹性扩展技术，针对虚拟机性能干扰问题，综合考虑影响缓存节点间数据迁移代价的多种因素，实现可持续服务、开销敏感的节点动态扩展。

在上述目标基础上，我们将结合现行技术和标准，研发相关技术的原型系统，并利用实际应用系统、基准应用系统进行案例和实证研究，对技术方案的可行性进行分析和评估，最终形成具有实际应用价值的新型分布式缓存服务系统。上述成果表现形式包括高质量的研究论文、可执行的软件系统、技术发明专利以及软件著作权等。

**2.2 研究内容**

**1）高时效、易使用的新型数据缓存模型**

现有的应用层缓存模型，属于“缓存分离”的cache-aside模型，依赖应用程序的开发人员管理内存数据，存在严重的学习成本和开发代价；中间层缓存模型虽然具有较高的易用性，但由于存在缓存结果失效、内外存数据同步代价等问题，仅适用于“读多写少”的应用场景，无法满足不同读写比例应用场景的“广谱”高时效需求。我们在项目预研阶段，利用TPC-W国际事务处理基准应用系统，针对上述缓存模型进行了经验式研究。结果显示，采用应用层缓存模型的分布式缓存方案，需要对系统进行大规模的重构、甚至放弃了一些复杂查询事务的支持（如Bestseller等），累计重构代码量占原系统代码量超过40%；另一方面，采用中间层缓存模型的查询结果缓存方案，在三种TPC-W基准测试的三种负载模式（浏览、购物、订单三种负载模式，分别代表不同读写比例的应用负载）下的性能表现差异显著，购物和订单模式由于写操作比例较高，其基准测试结果没有得到明显提升。

本课题将综合上述现有缓存模型的优势，研究兼顾时效性、易用性的新型数据缓存模型。首先，针对“缓存分离”问题，课题拟研究一种支持内外存融合的数据迁移转换机制，实现应用层数据对象模型与持久化存储层数据模型（如关系型数据库）的双向高效转换；同时，针对内外存数据的一致性管理，区别于中间层缓存模型提供的一致性同步机制，我们假设不同的租户、应用系统和业务组件具有差异化的数据一致性要求，并依此研究提供更灵活、可定制的数据一致性保障策略；另外，课题组将借鉴连续查询技术采用的滑动窗口机制以及增量计算思路，对上述机制进行优化以满足高时效数据处理与访问需求。其次，在编程模型方面，课题拟研究兼容SQL的分布式缓存编程模型，为开发人员提供符合现行技术标准的结构化查询编程接口（如JDBC API），这需要研究如何将SQL操作翻译转换为具有NoSQL特征的分布式内存数据操作；同时，拟借鉴关系型数据库SQL引擎的相关优化技术，对SQL的翻译、执行过程进行优化。最后，我们将利用实际应用系统、基准应用系统进行案例和实证研究，对技术方案的可行性进行分析和评估。综上，本部分的主要研究内容包括：

* 支持内外存融合的数据迁移转换机制；
* 兼容SQL的分布式缓存编程模型；
* 案例和实证研究，对方案的可行性进行分析和评估。

**2）SLA驱动的多租户性能隔离技术**

在高共享程度的多租户模式下，所有租户共享单一分布式缓存服务实例，并将其视为黑盒，引入的系统资源（内存、CPU、网络、存储）竞争会导致部分租户的服务质量需求无法得到满足。现有PaaS平台未提供面向SLA保障的性能隔离机制，而采用“尽力而为（best-effort）”的策略，这使得租户应用的性能变得难以预测。现有研究工作主要关注租户间内存资源的竞争，而未考虑CPU、网络、存储等其它系统资源竞争带来的影响，而与内存资源相比，其它系统资源竞争相关联的因素更为复杂、资源重配或调度带来的性能隔离时效性更强，同时，已有工作在设计性能隔离方法时未充分考虑各租户SLA需求可能无法同时满足以及不同负载模式对资源需求的差异性等问题。我们在项目预研阶段，采用雅虎的云服务基准测试工具Yahoo! Cloud Serving Benchmark（简称YCSB）对上述问题进行了实验分析，发现租户的并发规模、负载模式以及缓存命中率等因素都会引起高度共享的CPU、网络I/O、存储I/O资源产生竞争，进而发生租户间的性能干扰。我们的实验数据显示，租户性能受干扰影响下降高达300%。

针对上述问题，本课题拟首先研究面向多租户的分布式缓存性能模型动态构造技术，对上下文系统环境进行动态评估，综合考虑内存、CPU、网络I/O、存储I/O等内存计算敏感的系统资源变化以及租户负载变化对租户性能的影响。同时，研究系统资源的动态调配机制，实现白盒式、细粒度的资源动态管理，这需要针对内存资源设计一种新的内存空间划分机制和租户敏感的缓存替换算法，而在CPU、网络I/O、存储I/O资源分配方面，需要提出一种基于租户请求调度的资源调配算法。上述系统资源调配机制相辅相成，互为补充，弥补现有针对内存资源的性能隔离机制的滞后性。在上述研究内容基础上，设计SLA驱动的多租户性能隔离机制，依据租户SLA，利用性能模型动态求解各租户的资源容量阈值，然后利用系统资源的动态调配机制，动态调整租户资源容量，避免违背租户SLA。综上，本部分的主要研究内容包括：

* 面向多租户的分布式缓存性能模型动态构造；
* 内存计算敏感的系统资源动态调配机制；
* SLA驱动的多租户性能隔离算法。

**3）应用敏感的混合存储体系优化技术**

现有的混合存储体系主要面向“SSD + HDD”的两层混合体系架构，重点关注文件存储系统和数据库存储系统的优化，而针对“DRAM + SSD + HDD”的多层混合体系架构在分布式缓存系统中的优化方案缺乏相关研究工作，也未能充分考虑大数据应用以及云计算部署环境的特点。

首先，针对DRAM与SSD的“混合内存”体系，由于分布式缓存系统中关键数据（如中间数据、状态信息等）以及系统资源状态具有时变性，而现有缓存系统的数据存储管理策略无法及时感知这种变化，并且动态做出相应调整。我们在项目预研阶段，采用YCSB对上述问题进行了实验分析，发现当分布式缓存中的关键数据产生速率增加1至10倍时，数据存储速率仅提升20%-100%，导致数据存储速率与数据产生速率不匹配，严重情况下还会造成数据无法及时被持久化而发生内存泄露数据丢失情况。其次，针对SSD与HDD的存储系统优化，传统方法通常将每一台虚拟机作为独立的个体对待，未考虑到应用本身的特点和部署应用的各虚拟机之间的关联，并且，基于“先来先服务”模式SSD分配策略无法保障应用系统的服务质量。在项目预研中，我们利用TPC-W基准系统进行高频事务处理负载场景下的混合存储策略优化研究，通过调整应用系统内不同组件的SSD资源配额，可造成性能差异在2倍以上。

针对上述问题，本课题拟统筹考虑“DRAM + SSD + HDD”的混合存储体系架构特征，研究应用敏感的混合存储体系优化技术，建立“混合存储感知”的性能保障机制。首先，研究分布式缓存数据的自适应持久化策略，建立基于机器学习的持久化策略优化模型，通过在线感知分布式缓存中关键数据产生速率变化以及系统资源状态变化，持续调整模型，进而实现持久化策略的自适应调优。其次，研究应用敏感的SSD资源分配策略，通过自动感知应用系统架构、组件组成及行为，在线监测应用组件及执行环境对I/O性能的需求，进而实现SSD资源的分配优化。最后，考虑到不同应用对I/O需求的差异，拟研究一种混合存储全局资源优化策略，从全局视角感知不同应用的服务质量需求和优先级需求，统筹混合存储体系全局I/O路径的调度机制，实现合理的资源调度和优先级选择。综上，本部分的主要研究内容包括：

* 分布式缓存数据的自适应持久化策略；
* 应用敏感的SSD资源分配策略；
* 混合存储全局资源优化策略。

**4）虚拟化环境敏感的资源弹性扩展技术**

分布式缓存在弹性伸缩的同时，存储在缓存节点中的部分内存数据也需要在节点间完成迁移，此外，数据访问过程中产生的热点数据区同样会成为系统瓶颈，也需要对热点数据进行均衡迁移。对部署在云环境中的分布式缓存服务而言，数据迁移过程中伴随的虚拟机性能干扰、副本状态同步等问题会给系统性能带来严重影响，因此，如何有效降低迁移开销并保障数据迁移过程中的内存计算服务可用性和数据一致性成为新的挑战，而现有工作主要针对磁盘I/O、网络I/O、CPU等资源密集型应用的数据迁移问题开展研究，这些方法未考虑高时效、可扩展的大数据应用场景下，缓存服务的低延迟、负载不可预知和时变性以及系统有状态性等给数据迁移带来新的挑战。我们在项目预研阶段，采用YCSB对上述问题进行了实验分析，数据显示，虚拟机之间产生的性能干扰对分布式缓存数据迁移（包括迁移时间和性能衰减）具有显著影响，随着虚拟机数量的增长，迁移时间由于I/O阻塞呈非线性变化，性能衰减程度数倍于基准环境。

针对上述问题，我们首先需要研究建立数据迁移的开销模型，综合考虑迁移时间、性能衰减值等因素的影响，刻画底层虚拟化环境的性能干扰对数据迁移产生的影响，为数据迁移决策提供量化分析依据。另一方面，为及时消除热点数据对系统性能及服务的可用性造成的影响，本课题将在数据迁移机制基础上设计一种热点数据重均衡算法，当分布式缓存节点动态扩展时，将热点数据迁移到新的节点以均衡热点数据区的负载压力；同时，在选择热点数据和迁移目标节点时，需要研究面向资源优化的负载整合算法，在实现热点数据重均衡的同时兼顾各节点系统资源的整体利用率优化。综上，本部分的主要研究内容包括：

* 数据迁移与开销控制机制；
* 热点数据重均衡算法，均衡热点数据分布；
* 面向资源利用率优化的负载整合算法。

**2.3 拟解决的关键科学问题**

**1）数据缓存模型**

现有的数据缓存模型存在学习成本高、开发代价大的问题，同时，适用的负载场景有限，无法满足不同读写比例应用场景的“广谱”、高时效需求。如何提供高时效、易使用的数据缓存模型，降低系统的开发成本，是分布式缓存服务面临的首要问题。其中，如何实现内外存融合的数据迁移转换？如何实现高时效、自动化的外存数据加载以及内存数据持久化？如何实现系统开销敏感的数据一致性保障策略？如何提供兼容SQL的编程模型以及如何优化SQL语句的处理和执行过程？都是本课题在缓存模型方面需要解决的关键技术问题。

**2）租户性能隔离**

在高共享程度的多租户模式下，现有PaaS平台未提供面向SLA保障的性能隔离机制，而采用“尽力而为(best-effort)”的策略，这使得租户应用的性能变得难以预测。如何在最大限度地提高系统资源利用率的基础上，提供面向SLA保障的性能隔离机制，是本课题需要解决的重要问题。其中，如何动态构造分布式缓存的多租户性能模型，在线评估各类系统资源以及租户负载对租户性能的影响？如何针对内存计算敏感的系统资源，实现白盒式、细粒度的资源动态管理？如何设计SLA驱动的多租户性能隔离机制？这些都是本课题在性能隔离方面需要解决的关键技术问题。

**3）混合体系优化**

现有的混合存储体系优化研究工作主要关注文件存储系统、数据库管理系统，对于分布式缓存系统研究较少，无法充分发挥新型硬件带来的诸多优势。如何针对分布式缓存中关键数据和系统资源的时变性，实现数据的自适应持久化策略？如何感知云计算环境下虚拟机之间、应用组件之间的关联性，设计应用敏感的SSD资源分配策略？如何综合考虑多租户异构应用对I/O需求的差异，从全局视角感知不同应用的服务质量需求，实现混合存储全局资源优化？这些都是本课题在混合体系优化方面需要解决的关键技术问题。

**4）资源弹性扩展**

对部署在云环境中的分布式缓存服务而言，弹性扩展过程中伴随的数据迁移、虚拟机性能干扰、副本状态同步等问题会给系统性能带来严重影响。如何有效降低系统开销并保障扩展过程中的内存计算服务可用性和数据一致性是本课题需要解决的重要问题。其中，如何建立虚拟化环境下的数据迁移开销量化分析模型？如何利用硬件计数器选取相关性高的指标进行虚拟机性能干扰估算；如何定位热点数据，并兼顾资源利用率优化，选择恰当的迁移目标节点？是本课题在弹性扩展方面需要解决的关键技术问题。

3．**拟采取的研究方案及可行性分析**（包括研究方法、技术路线、实验手段、关键技术等说明）；

**3.1 研究方法及技术路线**

课题组将在充分理解吸收当前国内外主流软件技术的前提下，结合项目团队在多租户共享系统资源管理、分布式系统性能建模、分布式内存数据管理、系统测试验证等方面的研究成果，研究上文提出的研究内容和难点问题。项目将以“模型-支撑框架-关键技术”为研究思路，结合现行技术和标准，研发面向高时效、可扩展大数据应用的分布式缓存服务系统，按照“现有分布式缓存系统——支持新型数据缓存模型的系统——多租户、混合存储敏感的系统——虚拟化环境敏感的系统”的技术路线，逐步进行关键技术问题攻关。

本项目的总体研究方案的结构如图1所示，将分为以下4个阶段：

* **数据缓存模型研究**：开展新型数据缓存模型的研究，设计内外存数据迁移转换框架以及兼容SQL的数据访问框架，在内外存数据模型转换、数据一致性保障、SQL语句翻译及执行优化等关键技术方面取得突破。
* **性能隔离机制研究**：研究分布式缓存多租户性能模型，设计租户系统资源监测与管理框架，在性能模型动态构造、系统资源动态调配、租户SLA保障等关键技术方面取得突破。
* **混合体系优化研究**：研究混合存储性能保障模型，设计应用敏感的混合存储I/O性能框架，在分布式缓存数据的自适应持久化、应用敏感的SSD资源分配、混合存储全局资源优化等关键技术方面取得突破。
* **弹性扩展机制研究**：研究数据迁移开销模型，设计节点状态监测及数据迁移框架，在数据迁移控制、热点数据重均衡、负载整合优化等关键技术方面取得突破。

同时，结合现行技术和标准研发原型系统，并利用实际应用、基准应用进行案例和实证研究，对技术方案的可行性进行分析评估，最终形成具有实际应用价值的新型分布式缓存服务系统。

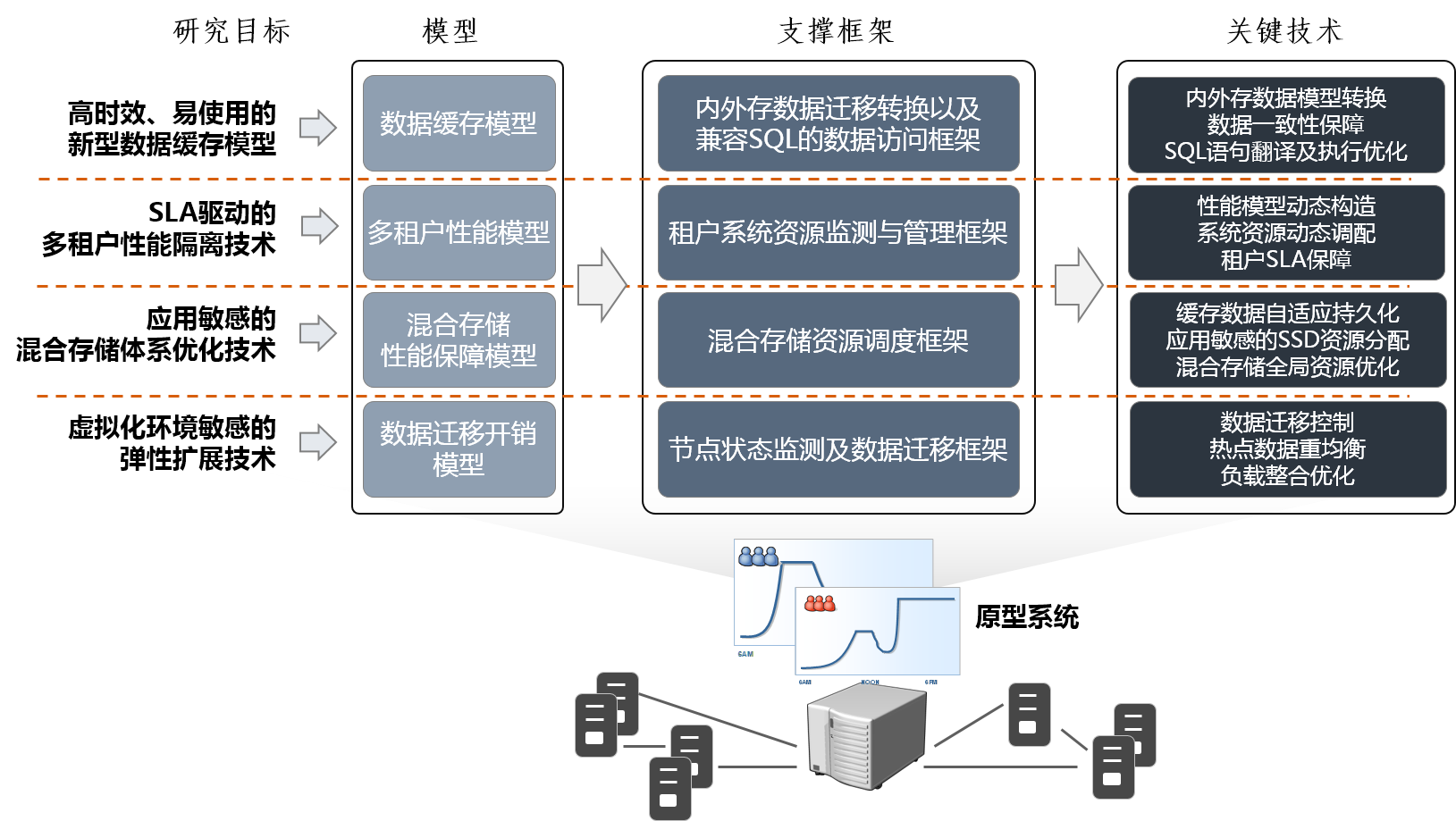


图1 研究方案

具体的研究方案与技术路线包括以下4个方面`：

**1）数据缓存模型**

针对现有缓存模型的不足，本课题拟提出一种兼顾时效性、易用性的新型数据缓存模型，降低现有分布式缓存系统的使用学习成本和开发代价。具体思路如图2所示。

首先，针对“缓存分离”问题，课题拟研究一种支持内外存融合的数据迁移转换机制，通过应用层数据对象模型与关系型数据库数据模型的双向转换，将关系型数据库表的行数据（外存数据）自动翻译为应用层数据对象，保存在距离应用层业务逻辑最近的应用服务器内存里。同时，在数据一致性管理方面，我们将假设应用系统的用户可以容忍一些复杂的分析型查询操作，并针对内存中执行的增、删、改数据操作，提供write/through、write/behind等多种数据一致性保障策略（前者提供严格数据一致性保障，后者提供最终数据一致性保障），支持开发人员根据应用需求设定一致性保障策略。

其次，针对上述机制的时效性问题，课题组将建立数据一致性保障开销模型，着重权衡策略协议在性能、一致性水平和通信开销方面的差异，基于开销分析和制定的规则实现对差异化需求自适应选取不同的一致性水平，目标是最小化开销。同时，借鉴连续查询技术采用的滑动窗口机制以及增量计算思路，对数据一致性保障策略的数据传输和通信机制进行优化。

最后，在编程模型方面，课题组通过提供标准的关系数据库访问驱动（JDBC），避免开发人员修改原有业务逻辑以及编写缓存数据访问逻辑，同时，建立灵活的SQL分离机制（SQL splitting）。我们认为具有NoSQL特征的分布式缓存适合执行增、删、改，而对于具有Ad-hoc特征的查询语句，则需要通过建立SQL执行代价评估模型，权衡分布式执行和集中执行的代价，通过SQL分离机制自适应的选择由分布式缓存执行SQL查询还是由后台关系型数据库执行。同时，由于需要解析SQL语句，我们将借鉴传统关系数据库的解析缓存、执行序列优化、查询结果缓存等优化机制，降低缓存的数据访问性能代价。

在上述工作基础上，课题组将利用团队承担的多个采用分布式缓存方案的实际大数据应用（交通领域、电力领域）以及基准应用进行案例和实证研究，对方案的可行性进行分析和评估。

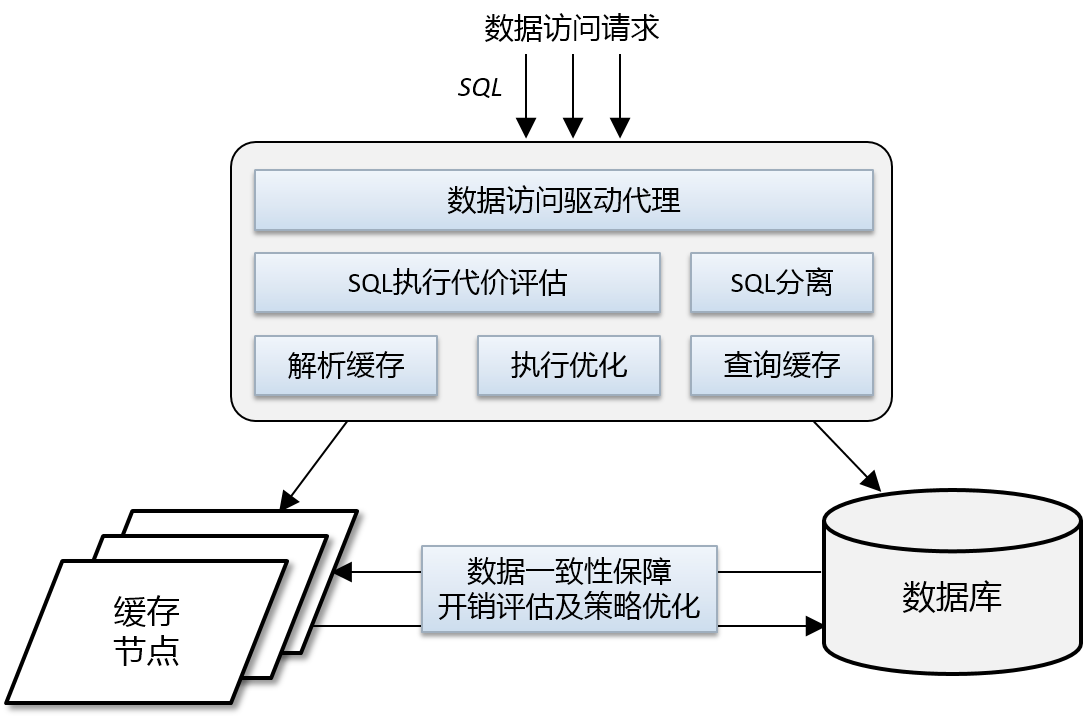


图2 内外存融合的数据缓存模型（示意图）

**2）多租户性能隔离**

本课题拟要研究如何在最大限度地提高系统资源利用率的基础上，综合考虑租户负载变化以及内存、CPU、网络、存储等系统资源的公平分配问题，提供SLA驱动的性能隔离机制。具体思路如图3所示。

首先，研究面向多租户的分布式缓存性能模型动态构造技术，利用多租户上下文信息，对共享进程内的租户线程进行分类和标记，动态建立租户与系统资源的映射关系，进而在线构造多租户性能模型。其中，与已有工作区别在于，我们将结合直接度量与统计分析方法，对内存、CPU、网络、存储等系统资源进行动态追踪评估，以适应不同应用、不同租户的负载模式变化，实现动态、精确的多租户资源剖面分析，进而提高性能模型的精确度与适应性。

其次，在系统资源动态调配方面，本课题方法综合考虑负载倾斜、时变以及资源访问瓶颈等问题，以节点为单位动态调整租户间的系统资源配额，重点保障租户间内存、CPU、网络I/O、存储I/O等系统资源的公平分配。我们拟建立内外两层资源控制环，外层控制环解决租户缓存空间如何公平分配的问题，内层控制环解决租户CPU、网络I/O、存储I/O资源竞争问题。其中，租户缓存空间划分机制基本思想是，将存在访问瓶颈的租户从其他缓存空间利用率较低的租户处暂借一定的资源配额，使之与负载量相匹配，而存在“债务”的租户需在其他节点归还这部分配额，这样分配给每个租户的系统资源总量保持不变；租户CPU、网络I/O、存储I/O资源分配机制与缓存空间划分机制互为补充，其基本思想是利用请求响应时间与访问量、读写比、数据大小间等度量的关系，设计请求调度队列及调度算法，通过调节各租户请求到达率、队列任务优先级等方式实现资源配额的动态调整。

最后，在SLA保障方面，基于曲线拟合的方法建立各租户SLA约束与资源元素的关系曲线，基于该关系曲线在线检测约束冲突。将冲突消解与资源分配规约为组合优化问题，资源分配以最大化各租户SLA的联合效用函数为目标，采用启发式算法求解租户资源分配方案。



图3 SLA驱动的多租户缓存服务性能隔离方案（示意图）

**3）混合体系优化**

本课题拟要研究在“DRAM + SSD + HDD”的混合存储体系架构下，综合考虑大数据应用的存储需求差异以及负载的动态变化，提供一种应用敏感的混合存储体系优化方案。具体思路如图4所示。

首先，针对分布式缓存数据的自适应持久化，研究一种支持自适应调整的分布式缓存持久化策略。我们将建立基于机器学习的持久化策略优化模型及框架，通过实时采集系统信息，在线感知分布式缓存中关键数据产生速率变化以及系统资源状态变化，并持续改进模型，进而选择最优的缓存数据持久化方案，同时根据方案的实施效果对模型进行反馈，在迭代的过程中不断的优化模型，提高持久化策略的自适应性。

其次，针对应用敏感的SSD资源分配，研究一种能够自动感知应用系统架构、组件组成及行为，从而动态分配SSD缓存资源的机制。我们将建立应用数据处理和访问过程的追踪机制，并重点扩展I/O请求感知模块，对应用中不同组件的I/O需求进行度量和评估；同时建立全局控制器，根据不同组件的I/O需求，确定同一应用部署的多台虚拟机的权重，最终反馈给数据映射模块进行具体的调整。

最后，针对混合存储全局资源优化，研究一种综合考虑多租户异构应用对I/O需求的差异性，并优先保障高优先级应用性能的机制和策略。我们将在前述工作基础上，扩展I/O请求感知模块，实现I/O请求标记；同时接管虚拟机内部的I/O调度器，扩展数据映射模块，建立统筹混合存储体系全局I/O路径的调度机制，从全局视角保障不同应用的服务质量需求和优先级需求。



图4 基于机器学习的混合体系优化方案（示意图）

**4）资源弹性扩展**

本课题拟针对虚拟化环境下普遍存在的虚拟机性能干扰问题，研究建立数据迁移开销模型，实现可持续服务、开销敏感的节点动态扩展。具体思路如图5所示。

首先，为了系统的评价数据迁移开销，拟采用统计学习方法构建性能干扰敏感的预测模型，预测每个数据迁移操作的迁移时间、性能衰减等因素，为数据迁移决策提供量化分析依据。该模型需要包含虚拟机层关键性能参数，系统反映底层操作的细节并刻画虚拟机性能干扰程度，为此我们拟利用底层硬件计数器在线记录程序运行期间产生的硬件事件信息，分析虚拟机性能干扰与硬件计数器之间的关系，并选取相关性高的指标进行虚拟机性能干扰估算。对于预测模型的构建，拟选取多种常用的回归分析方法进行预测效果评估，包括线性逐步回归法、主成分分析法和非线性回归法等。

然后，课题组将数据迁移问题纳入负载均衡、节点动态扩展等场景，拟构建一个基于MAPE（Monitor-Analyze -Plan-Execute）控制环的数据迁移框架，包含状态监测、租户SLA检测、数据迁移决策和执行等组成分。其中，监测模块负责收集虚拟机、缓存性能、负载均衡度等状态数据；数据迁移决策模块首先负责迁移行为的触发决策，当负载均衡度低于某一阈值或缓存性能不能满足租户SLA时便会触发数据迁移操作，其次，该模块还负责制定迁移计划，包括迁出迁入节点的选取、迁移带宽和迁移数据量的控制等；执行模块位于每个缓存节点，负责数据迁移的执行。

在上述数据迁移框架基础上，对于每个可能的数据迁移操作，首先采用预测模型预测其迁移时间和性能衰减值，然后计算迁移开销，迁移决策算法根据均衡度和迁移开销值选取一系列最优的数据迁移操作。在迁移数据的选择方面，综合考虑热点数据的重均衡和资源利率优化，即将热点数据分区装入低负载节点中，这一问题可以被规约为一个NP难的装箱(bin-packing)问题。考虑节点的异构性，迁移节点的选取拟采用基于动态权值调整的方法，为每个节点设置一个初始权值，建立服务节点性能模型以刻画节点运行时负载状况，包括服务请求状况、CPU内存使用状况和网络负载等，运行时基于该性能模型动态调整权值。



图5 虚拟机性能干扰敏感的缓存数据迁移方案（示意图）

**5）原型系统研制及技术验证**

在上述阶段的技术研究过程中，项目组将结合现行技术和标准，研发相关技术的原型系统，并利用项目团队承担的多个采用内存计算方案的实际应用系统（交通领域、电力领域）以及TPC-W、YCSB等国际权威基准测试进行案例和实证研究，对技术方案的可行性进行分析和评估，最终形成具有实际应用价值的新型分布式缓存服务系统。

**3.2 可行性分析**

基于上述研究方法，我们认为本项目具有良好的可行性，理由在于：

（1）大数据应用环境下，利用分布式缓存等分布式内存计算方案提供低时延、高可扩展的数据访问加速服务，是云平台提升应用性能的重要手段，也是支撑重要领域、行业信息系统的关键技术。现有的相关商业软件、开源软件与实际需求仍存在很大差距，相关技术的学术研究处于起步阶段。本项目的研发具有良好的技术发展机遇。

（2）本项目关注的方法和技术是相关领域的研究热点和难点，国内外研究人员已经对相关技术展开了积极的研究（参见（一）“立项依据”）。本项目研究内容也是在现有工作基础上，研究解决基于云计算模式的高时效、可扩展大数据应用场景下分布式缓存服务面临的新挑战和新问题。已有的相关研究成果为本项目的研究提供了一个极为有利的外部环境。

（3）项目申请人及主要研究人员在网络化软件运行支撑和在线管理技术、多租户共享系统资源管理、分布式系统性能建模、分布式内存数据管理、系统测试验证等与本项目相关的技术方面进行了比较深入的研究（参见（二）“研究基础”），保持对该领域最新学术动态的准确掌握，在研究中积累了许多宝贵经验，部分成果已经在本领域知名期刊和会议发表，**代表论文近3年国际他引>100，被国内外知名研究机构持续关注和引用，提出的细粒度系统资源评估方法成为国际知名学者建立的系统资源评估库三类代表方法之一。**这些工作为本项目研究内容的深入开展打下了良好的基础。

（4）在项目前期预研阶段，项目团队针对分布式缓存现有计算模型存在的学习成本高、开发代价大等问题，**利用TPC-W国际事务处理基准应用搭建了系统的实验测试床软硬件环境，使用RedHat Infinispan、Hazelcast、Ehcache等多款开源分布式缓存产品对TPC-W基准应用进行重构，开展了深入的经验式研究，并以开源软件项目的形式发布了初步研究结果（www.trustie.net/projects/project/show/ExBench4Q\_for\_NoSQL），该项目获OW2国际程序竞赛二等奖，**为进一步评估项目所提出方法的可行性和有效性提供了较为全面的实验平台。

（5）项目申请人牵头研制的国产分布式缓存中间件EasyCache，**将国际权威事务处理基准测试性能提升2个数量级，同时，针对自主可控CPU服务器、数据库进行适配优化，显著提升了自主可控软硬件环境下的数据访问效率。目前，EasyCache已经在北京市交通委、中国能源建设集团等重要领域的信息化项目中得到应用，其关键技术转化到国产中间件领先厂商——北京东方通科技股份有限公司，并被作为核心技术产品纳入工信部“国产中间件产品及技术未来10年发展路线图”**。这些工作为本项目的研究提供了坚实的系统工作基础和丰富的系统应用经验。

（6）本项目的申请人及主要研究人员曾参与或主持完成多项科研课题，包括国家重点基础研究发展规划项目（973）课题“面向可信和服务质量的网构软件中间件模型及运行机制”（课题编号：2009CB320704）、国家自然科学基金“面向多租户的弹性平台云的资源管理机制研究”（课题编号：No.61100068）、国家863项目课题“网构化软件运行支撑和在线管理技术与机制”（课题编号：2012AA011204）等。通过参与这些课题的研究，本项目组的主要研究人员积累了完成同类科研项目的经验。

综上所述，本项目的研究计划和具体方案无论是在客观上还是在主观方面都有着很好的可行性。

4．**本项目的特色与创新之处；**

本项目针对内存计算服务化、租户资源共享化、存储体系混合化、系统环境虚拟化的现状与趋势，研究解决基于云计算模式的高时效、可扩展大数据应用场景下分布式缓存服务面临的新挑战和新问题，相关关键技术问题的提出符合云计算、大数据应用环境下软件技术发展的迫切需求，具有很强的新颖性和极大的研究价值，所采用的技术方案较之同类工作有比较鲜明的特色，可望对分布式内存计算的相关基础理论和关键技术取得突破性进。

1）内存计算已成为云计算模式下的核心关键服务，如果失去分布式缓存这一关键技术的支撑，云计算的潜能将十分有限。然而，现有的应用层数据缓存模型依赖应用程序的开发人员管理内存数据，存在严重的学习成本和开发代价，而中间层数据缓存模型虽然具有较高的易用性，但无法满足不同读写比例场景的高时效需求。本课题综合现有数据缓存模型的优势，提出一种兼顾时效性、易用性的新型数据缓存模型，降低现有分布式缓存系统的学习成本和开发代价。

2）多租户共享是云平台服务的基本特征，高共享程度的多租户模式是技术发展趋势。然而，多租户模式引入的系统资源竞争会导致租户服务质量需求难以保障。本课题针对高共享程度的多租户模式，提出一种SLA驱动的性能隔离技术，在实现白盒式、细粒度的租户性能隔离的同时，避免各租户相互独立的SLA违约。

3）基于非易失闪存的混合内存体系将成为云计算平台的“标准配置”，分布式缓存等内存计算技术需要做出必要变化与之相适应。然而，现有关于混合存储优化的相关研究工作主要针对文件存储系统和数据库管理系统，而针对分布式缓存系统的研究较少。本课题面向“DRAM + SSD + HDD”的混合存储体系架构，综合考虑大数据应用的存储需求差异以及负载的动态变化，提供应用敏感的混合存储体系优化技术，实现“混合存储感知”的性能保障。

4）云计算的弹性资源供给使得分布式缓存服务需要在满足用户SLA的基础上，提供最大限度的可伸缩性。然而，现有虚拟化技术仍面临性能干扰难题，需要结合应用特征，在平台层统一协调处理。本课题针对缓存资源弹性扩展过程中面临的虚拟机性能干扰问题，提出一种虚拟化环境敏感的数据迁移技术，综合考虑影响缓存节点间数据迁移代价的多种因素，实现可持续服务、开销敏感的节点动态扩展。

5．**年度研究计划及预期研究结果**（包括拟组织的重要学术交流活动、国际合作与交流计划等）。

**5.1 年度研究计划**

本项目总研究时间为四年，其年度计划及进展预计如下：

**2017年度——**针对现有数据缓存模型的不足，研究提出一种兼顾时效性、易用性的新型数据缓存模型，降低现有分布式缓存系统的学习成本和开发代价。具体包括：开展新型数据缓存模型的研究，设计内外存数据迁移转换框架以及兼容SQL的数据访问框架，在内外存数据模型转换、数据一致性保障、SQL语句翻译及执行优化等关键技术方面取得突破。

**2018年度——**研究如何在最大限度地提高系统资源利用率的基础上，综合考虑租户负载变化以及网络I/O、CPU等系统资源的公平分配问题，提供面向SLA驱动的性能隔离机制。具体包括：研究和设计面向PaaS的分布式缓存多租户性能模型以及租户系统资源监测与管理框架，在性能模型动态构造、系统资源动态调配、租户SLA保障等关键技术方面取得突破。同时，针对混合存储体系架构，综合考虑大数据应用的存储需求差异以及负载的动态变化，提供应用敏感的混合存储体系优化机制。具体包括：研究建立“混合存储感知”的性能保障模型，在分布式缓存数据的自适应持久化、应用敏感的SSD资源分配、混合存储全局资源优化等关键技术方面取得突破。

**2019年度——**针对虚拟化环境下普遍存在的虚拟机性能干扰问题，研究建立数据迁移开销模型，实现可持续服务、开销敏感的节点动态扩展。具体包括：研究建立数据迁移开销模型，设计节点状态监测及数据迁移框架，在数据迁移控制、热点数据重均衡、负载整合优化等关键技术方面取得突破。

**2020年度——**主要进行研究成果、方法和系统的完善，结合现行技术和标准，形成面向实际应用的平台系统，开展课题关键技术的运用并力争取得一定的社会和经济效益。

在项目研究期间，将计划参加若干与项目目标紧密相关的高水平国际学术会议，预计5-8 人次。

**5.2 预期研究成果**

主要体现为：

1. 在国内外重要学术期刊、会议发表 SCI/EI 检索论文8-10篇；
2. 申请发明专利和软件著作权4项以上；
3. 通过本项目研究，培养一支高水平内存计算研究队伍，培养博士研究生2-3名、硕士研究生6-10名；
4. 创建开源项目1项，提供技术文档与源代码。

**（二）研究基础与工作条件**

1．**研究基础**（与本项目相关的研究工作积累和已取得的研究工作成绩）；

项目团队人员由中科院软件所骨干力量组成，长期从事分布式系统的软件模型、体系结构、运行支撑、开发测试等技术的研究，获2009年度北京市科技进步一等奖、2010年度国家科技进步二等奖、2013年度湖南省技术发明一等奖、2015年度国家技术发明二等奖。

近年来，本团队针对网络化软件运行支撑和在线管理技术、多租户共享系统资源管理、分布式系统性能建模、分布式内存数据管理、系统测试验证等与本项目相关的技术展开深入研究，具有一定的成果和技术积累，近在JSS、Computing、FCS、中国科学、CIKM、ICWS、CLUSTER、COMPSAC、CLOUD等本领域知名期刊和会议发表论文30余篇，代表论文近3年国际他引>100，获发明专利授权12项，形成了自己的优势和特色，在国内外均产生了积极影响。相关成果被IBM、SAP、CA、VMWare、HITACHI等国际知名企业以及卡内基梅隆大学、多伦多大学、卡尔斯鲁厄理工学院、帝国理工学院、法国信息与自动化研究所等国际知名研究机构关注和引用。相关系统成果在众多领域以及国家重大工程中得到运用，多项关键技术转化到我国基础软件企业，推动了企业技术发展。

**1）网络化软件运行支撑和在线管理技术**

项目组作为核心成员参加了国家863 计划课题“网构化软件运行支撑和在线管理技术与机制（课题编号：2012AA011204）”的研究和开发工作，针对一切皆服务（XaaS）、多承租等新型服务化应用模式的技术需求，重点研究网构软件的云化支撑、应用引擎、在线管理、生产开发等技术和机制，建立了新型网构化软件的运行管理和生产支撑技术体系，研制了延时可控、线性扩展和高可靠的网构化软件运行和管理平台系统及工具。相关系统成果在互联网、电信、交通、能源、教育等众多领域，以及国家“核高基”重大专项、国家卫生部“脑卒中筛查与防治工程”等重大工程中得到运用，多项关键技术转化到中创软件、东方通科技、CSIP、蚁坊软件、苏州博纳讯动等我国软件企业及机构，推动了企业技术发展。**成果获2013 年湖南省技术发明一等奖。**形成多项软件开源项目，发布在OW2、Trustie、GitHub等开源社区，累计贡献源码超过10万行，下载量超过3万次，并多次在国际、国内程序竞赛获奖。**OW2 执行主席Cedric Thomas称赞“This is a remarkable set of projects supporting cloudware”。**

**2）应用级度量、负载敏感的多租户共享系统性能异常诊断技术**

在面向多租户的共享资源管理方面，开展了应用级度量、负载敏感的系统性能异常诊断等关键技术的探索与研究，**成果发表在国际知名会议COMPASAC(CCF B)、Cloud等，近3年国际他引>50，国际同行给予高度评价，并对研究成果进行了扩展**。卡尔斯鲁厄理工学院Descartes Research Group负责人、SPEC Research Group主席Ing. Samuel Kounev教授发表在SCI期刊和国际知名会议ICWE上的系列论文认为我们的工作**提供了多租户资源评估和性能隔离的新方法，并成为Samuel教授、FP7项目负责人Casale教授、VMWare资深工程师Xiaoyun Zhu女士联合建立的系统资源评估方法库(LibReDE)的三类代表方法之一**。慕尼黑工业大学信息系统系主任、AIS主席Helmut Krcmar 教授以及FORTISS研究所资深研究员WILLNECKER博士发表在EPEW2015的文章中**将我们的方法作为主要对比对象进行分析，认为我们的方法具有更低的数据收集需求，具有更广泛的应用价值（“can be applied to a broad variety of technologies as the requirements for data collection are lower”）。**欧盟FP7项目负责人、葡萄牙科英布拉大学的Silva教授认为我们的**“方法对系统性能问题的快速检测与调优非常重要(very important)”**。欧盟FP7项目(MODAClouds)负责人、英国帝国理工学院Casale教授在其发表在软件工程领域国际顶级期刊IEEE Transactions on Software Engineering和国际知名会议IEEE 21st MASCOTS以及FP7项目报告中将我们的方法作为一类代表性的工作进行了系统介绍。**世界最大企业应用软件供应商SAP的研究者（Karlsruhe研究中心、Belfast研究中心、HANA应用研究中心）持续关注和引用我们的工作**：Karlsruhe研究中心研究员Manuel Loesch发表在CLOSER2014的论文中将我们的方法作为相关方法的唯一典型代表进行重点介绍，并特地发邮件咨询，认为**成果对SAP非常有吸引力(“very interesting to us”)**；HANA应用研究中心的研究员Rouven Krebs发表在国际知名会议CCGrid2014的论文中**在我们的方法基础上提出新的多租户应用资源控制方法。**

**3）应用级、细粒度的分布式系统性能建模技术**

在分布式系统性能建模方面，开展了应用级、细粒度性能建模关键技术的探索与研究。**成果发表在COMPASAC(CCF B)、QSIC等，近3年国际他引>20。**卡内基梅隆大学著名学者、Drew D. Perkins首席教授HYONG Kim教授在其发表在分布式计算领域国际顶级期刊IEEE Transactions on Parallel and Distributed Systems的文章中认为我们的**方法是合理和有效的，并作为主要对比对象进行分析**。多伦多大学Middleware Systems Research Group负责人、Bell University Laboratory首席教授Hans-Arno Jacobsen在其发表在软件工程领域国际顶级期刊IEEE Transactions on Software Engineering的文章中，将我们的工作作为一类代表性的工作进行了介绍。温伯尔尼理工大学CeCSES与ESRLab负责人Jun Han教授认为我们的工作**解决了大型系统性能预测的关键问题**。IBM(德国研究中心、苏黎世研究中心、东京基础研究所)、CA (世界最大IT管理软件供应商) 等国际知名企业持续关注和引用我们的工作。

**4）分布式内存数据策略管理及优化技术**

在分布式内存数据管理方面，开展了缓存策略重配优化、数据迁移优化等关键技术的探索与研究。**成果发表在国际知名会议Springer Computing**(SCI&SCIE，CCF B)**、CIKM(CCF B)、CLUSTER(CCF B)等，近3年国际他引>20，国际同行给予高度评价，并对研究成果进行了扩展**。布京理工大学计算机系主任Lars Lundberg教授在其发表在国际知名期刊Journal of Systems & Software的文章中认为我们的**方法是检测集群节点负载均衡问题的基础，并基于我们的工作提出系统自适应调优的新方法。**西苏格兰大学计算机系主任Christos Grecos教授在其发表在服务计算领域顶级期刊IEEE Transactions on Services Computing的文章中**以我们的方法为核心技术建立了服务计算扩展框架WSF。**

**5）系统测试验证技术**

在分布式系统负载测试方面，开展了自动化负载生成、大规模测试资源管理等关键技术的探索与研究，成果发表在SOSE、Internetware等，**形成开源项目1项，发布在OW2、Trustie、GitHub等开源社区，累计下载近2万，国际下载近1万，获2015年OW2国际程序竞赛第1名、2013年中国云-移动互联网大奖赛商业创新类第2名、2013年NASAC软件研究成果原型竞赛第2名。**加拿大女王大学NSERC/BlackBerry Software Engineering主席Ahmed E. Hassan教授在其发表在国际软件工程领域顶级期刊IEEE Transactions on Software Engineering的综述文章中将我们的方法作为一类代表性的工作进行了介绍。**成果应用在：IBM分布式缓存产品eXtreme Scale、中国能源建设集团的电网流式数据处理平台等系统评测工作**；东方通、金蝶、中创等专业中间件厂商的产品测试任务；国家“核高基”重大专项成果验收、评估等测试任务；“国家软件公共服务平台”建设工程等。性能基准测试的相关成果内容被纳入项目团队参与完成的中国电子行业标准《信息技术 交易中间件 性能测试规范》，该行业规范是**国内首个软件性能测试相关的行业标准规范**。

**6）国产分布式缓存中间件系统**

项目申请人主持研制的国产分布式缓存中间件EasyCache，该系统兼容国产基础软硬件，可适应云计算、物联网等新型计算环境，满足高响应性、高并发、高可扩展的数据处理和访问需求。TPC-W国际事务处理基准应用测试结果显著提升，**平均响应速度提升2个数量级、关键业务响应速度提升3个数量级**。EasyCache针对国产CPU服务器、操作系统、中间件、数据库进行适配优化，**显著提升了自主可控软硬件环境下的数据访问效率。**EasyCache已经在交通、能源等领域得到实际运用，例如，在北京市交通委的出租车动态运行监测系统，**基于国产基础软硬件平台实现了6.67万辆出租车的实时GPS数据处理和路况动态分析**；在中国能源建设集团的电网海量数据管理系统，**实现了百万量级电表用户实时数据处理和查询**。多项关键技术转化到国产中间件领先厂商——北京东方通科技股份有限公司，推动了企业技术发展。

2．**工作条件**（包括已具备的实验条件，尚缺少的实验条件和拟解决的途径，包括利用国家实验室、国家重点实验室和部门重点实验室等研究基地的计划与落实情况）；

本课题依托中国科学院软件研究所下属的计算机科学国家重点实验室。办公地点位于中科院软件园区新落成的5号科研楼，总共1300平方米，具有先进的办公环境、设备资源和网络环境，新购置刀片服务器集群、高端服务器集群等设备，为课题研究工作顺利实施提供了充分的能力支撑。

在国内外相关研究机构中，项目组与北京大学、北京航空航天大学、南京大学、国防科技大学、俄亥俄州立大学、多伦多大学、慕尼黑工业大学、香港科技大学等有密切的合作和交流关系。项目组与国际系统软件知名学者俄亥俄州立大学计算机系秦峰教授建立了起长期深入的合作关系，秦教授是分布式系统软件领域的国际知名专家，项目组与秦教授在分布式内存数据一致性管理方面的合作研究成果发表在相关领域顶级会议SC2014。以此为基础，项目组能够很好地跟踪和了解国际先进技术，并开展创新研究。

在队伍组织方面，项目组长期以国家自然基金项目和973、863课题为牵引，在分布式计算和软件工程方向形成经验丰富、梯队合理、成员稳定的研发团队，整个队伍知识结构合理，研究基础好、工程能力强，为本项目顺利开展提供了保障。

3．**正在承担的与本项目相关的科研项目情况**（申请人和项目组主要参与者正在承担的与本项目相关的科研项目情况，包括国家自然科学基金的项目和国家其他科技计划项目，要注明项目的名称和编号、经费来源、起止年月、与本项目的关系及负责的内容等）；

无。

4．**完成国家自然科学基金项目情况**（对申请人负责的前一个已结题科学基金项目（项目名称及批准号）完成情况、后续研究进展及与本申请项目的关系加以详细说明。另附该已结题项目研究工作总结摘要（限500字）和相关成果的详细目录）。

申请人承担了国家自然科学基金青年基金项目“面向多租户的弹性平台云的资源管理机制研究”(编号：61100068，起止时间为：2012.01-2014.12)的研究，该项目已于2014 年结题。

项目组针对多租户弹性平台云的资源管理需求，构建了面向高共享程度的多租户模式、支持资源垂直供给的资源管理模型，突破基于统计分析的多租户资源剖面分析、面向时变上下文环境的QoS控制等关键技术，研发了相应的支撑框架和系统，并在基于云计算模式的实际系统中进行了应用验证。其研究成果如图6所示。

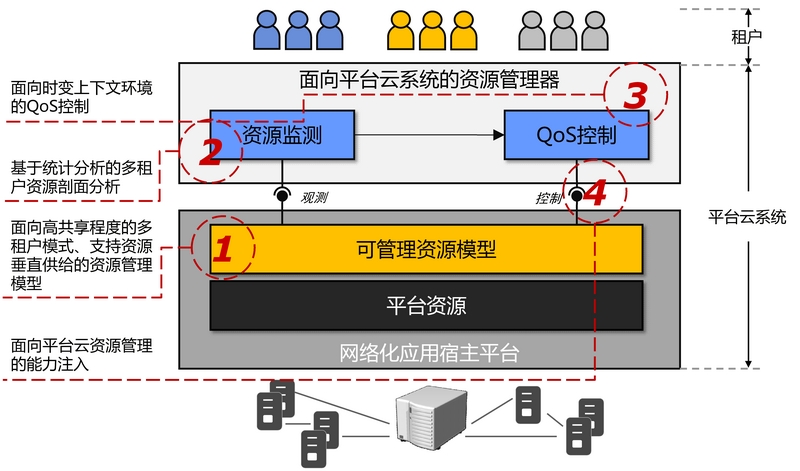


图6 青年科学基金项目已经取得的研究成果

该项目为申请项目中拟开展的SLA驱动的多租户性能隔离技术提供了良好的研究基础和系统框架基础。主要表现为：在多租户共享系统资源管理技术方面，已完成的青年科学基金项目建立了面向多租户系统的应用级资源管理模型及其并发控制及容量动态调整算法，提出了支持应用级度量、负载敏感的多租户共享系统性能异常诊断方法以及性能模型自动构造机制，设计实现开源原型系统Tenant Resource Estimator for Java Servers（TRE4J），这些成果为本项目的研究提供了丰富的理论依据和系统框架基础。申请项目将基于上述模型及方法，结合分布式共享内存数据管理系统的系统特征和多租户管理需求，进行更具体和深入的机制研究和扩展工作，包括综合考虑内存数据访问负载变化以及内存、网络I/O、CPU等系统资源的公平分配问题，研究建立面向分布式共享内存的多租户性能模型，在性能模型动态构造、系统资源动态调配、租户SLA保障等关键技术方面取得突破。

已完成项目的研究工作总结摘要如下：

平台云是介于硬件基础设施和网络化应用之间的抽象层，是云计算模式的重要类型之一，它为解决多租户共享、资源弹性供给等云计算特性所带来的资源管理技术挑战提供了新的切入点。本课题研究提出一套较为系统化的多租户资源管理关键技术，为云计算模式的发展提供创新技术支持，取得的创新成果包括：1）建立了一个基于事务处理链的应用级多租户资源管理模型，基于信号量机制给出了模型的并发控制算法， 支持事务级、分阶段的多租户Web应用资源管理，并避免因修改交互协议而产生的再工程代价；2）提出了一种基于统计分析的多租户资源剖面分析方法，针对基于共享Java中间件的多租户Web应用的细粒度系统资源状态进行动态评估，并给出了解决non-determinism问题和multicollinearity问题的技术方案；3）提出了一种多租户Web应用自动化性能建模方法，考虑了用户行为与系统中不同服务之间的关联，动态构造与应用实际状态及上下文环境相符的性能模型，并动态更新模型参数；4）提出一种资源垂直扩展开销模型，可有效感知多租户性能干扰状况，权衡扩展时间与性能衰减值；进一步，提出一种开销敏感的垂直扩展算法，基于开销模型与均衡度制订扩展计划，选取最优的扩展操作。

已完成项目发表学术论文10篇，其中SCI/EI索引7篇，近3年他引>50，被SAP、VMWare等企业以及ICL、KIT等国际知名研究机构关注和引用，**提出的多租户系统资源评估方法成为国际知名学者建立的系统资源评估库三类代表方法之一，获2014年度湖南省技术发明一等奖、2012年度OW2国际程序竞赛二等奖**。

代表性成果包括：

1. 王怀民、魏峻、尹刚、王伟、周明辉、刘江宁, 基于网络的可信软件大规模协同开发环境核心技术及其应用, 湖南省人民政府, 湖南省技术发明奖, 一等奖, 2014/1/20.
2. Xiulei Qin, Wei Wang, Wenbo Zhang, Jun Wei, Xin Zhao, Hua Zhong, Tao Huang. PRESC2: efficient self-reconfiguration of cache strategies for elastic caching platforms. Computing, 2014, 96(5):415-451.
3. Qiang Gao, Wei Wang, Guoquan Wu, Xuan Li, Jun Wei, Hua Zhong. Migrating Load Testing to the Cloud: A Case Study. Proceedings of the 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering. IEEE Computer Society, 2013:429-434.
4. 王伟, 黄涛, 魏峻, 钟华, 宋云奎. 面向多租户web应用的性能隔离方法. 中国科学：信息科学, 2013 (1), 45-59.
5. Wei Wang, Xiang Huang, Xiulei Qin, Wenbo Zhang, Jun Wei, Hua Zhong, Application-Level CPU Consumption Estimation: Towards Performance Isolation of Multi-tenancy Web Applications. in Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on , vol., no., pp.439-446, 24-29 June 2012.
6. Xiulei Qin, Wenbo Zhang, Wei Wang, Jun Wei, Xin Zhao. Towards a Cost-Aware Data Migration Approach for Key-Value Stores. 2012 IEEE International Conference on Cluster Computing IEEE, 2012:551-556.
7. 王伟, 黄翔, 张文博, 魏峻, 钟华,黄涛. 多租户web应用的CPU资源动态评估方法. 计算机学报, 2012, 34(12), 2292-2304.
8. 黄翔, 王伟, 张文博, 魏峻, 黄涛. 面向性能剖析的Web应用自动性能建模方法. 软件学报, 2012, 34(4), 786-801.
9. 黄涛, 钟华, 魏峻, 王伟, 李萱. 一种面向分布式系统性能测试的测试资源管理方法, 2013/12/4, 11, 中华人民共和国知识产权局, CN201310376714.8.
10. 王伟(指导老师), 李萱, 江浩亮, 闫东旭. Bench4Q as a service, 国际OW2 开源编程大赛二等奖, 2012.

**（三）其他需要说明的问题**

1. 申请人同年申请不同类型的国家自然科学基金项目情况（列明同年申请的其他项目的项目类型、项目名称信息，并说明与本项目之间的区别与联系）。

无。

2. 具有高级专业技术职务（职称）的申请人或者主要参与者是否存在同年申请或者参与申请国家自然科学基金项目的单位不一致的情况；如存在上述情况，列明所涉及人员的姓名，申请或参与申请的其他项目的项目类型、项目名称、单位名称、上述人员在该项目中是申请人还是参与者，并说明单位不一致原因。

无。

3. 具有高级专业技术职务（职称）的申请人或者主要参与者是否存在与正在承担的国家自然科学基金项目的单位不一致的情况；如存在上述情况，列明所涉及人员的姓名，正在承担项目的批准号、项目类型、项目名称、单位名称、起止年月，并说明单位不一致原因。

无。

4. 其他。

无。