FUNCTION **High value density knowledge discovery with graph database**

基于图论的 高价值密度知识 型数据存储系统的设计概述

摘要：随着IT信息技术的高速发展，大数据技术的使用已经深入到越来越多的领域。目前大数据的获取、存储和批量处理技术已经日臻成熟，但是如何从海量的数据集中高效的获取具有业务针对性的有用信息（`高价值密度知识`），依然是一个复杂的问题。本文介绍了一种使用图论算法与传统数据仓库概念相结合的方式；综合人类的主观判断与机器的客观数据分析来获取高价值密度知识的方法。并介绍了使用这一方法实现高价值密度知识获取数据平台的软件系统基本架构设计。

大数据的特点和使用中的挑战以及一种可行的解决方式

➜ 大数据的特点

目前大数据技术已经成为了企业构建数据资产平台的事实技术标准，它描述了一个整体信息管理战略，其中包含并集成了众多新的数据、数据管理以及传统数据类型。大数据系统中包含的数据量巨大，数据类型多样（数据间有很强的因果关系的`结构化数据`、数据间因果关系较弱的`半结构化数据`以及数据间无因果关系的`非结构化数据`），数据来源也具有多样性（不同的应用系统、各类终端设备以及互联网等），例如：

■ 交易类数据
- ERP、数据仓库、在线交易处理(OLTP)
- 传统的关系数据以及非结构化和半结构化信息
- 云计算平台的企业数据

■ 交互类数据
- 社交媒体数据、 Web 文本和点击流数据
- 设备和传感器信息、GPS 和地理定位映射数据
- 电子邮件、呼叫记录
- 海量图像文件
- 科学数据

大数据还可以用四个V来定义：

Volume（大量）：数据量。当数据量指的是更多数据时，它指独特数据的粒度化性质。大数据需要处理大量低密度、非结构化的Hadoop数据 — 也就是说，值未知的数据，如Twitter数据馈送、网页和移动应用上的点击流、网络流量、启用了传感器的设备捕获的光速数据，等等。大数据的任务就是将此类Hadoop数据转变成有价值的信息。对于某些组织，数据量可达数十TB级，还有一些组织的数据量高达数百PB。

Velocity（高速）：数据接收和操作的速度快。速度最高的数据通常直接流进内存中而非写入磁盘。有些物

联网(IoT)应用有健康运行和安全性要求，需要实时评估和操作。基于互联网的其他智能产品可实时或近似实时地运行。例如，消费类电子商务应用力求结合移动设备位置和个人偏好来开展有时效性的营销活动。在操作方面，移动应用体验的用户群体庞大、网络流量越来越高且希望立即获得响应。

Variety（多样）：新的非结构化数据类型。文本、音频和视频等非结构化和半结构化数据类型需要进行更多处理才能提取出意义和支持元数据。非结构化数据在得到理解后有着与结构化数据相同的许多要求，如汇总、来历追溯、可审核性和私密性。当已知来源的数据发生变化但没有通知时，复杂性将进一步提高。频繁或实时的模式变化对事务环境和分析环境而言都是巨大的负担。

Value（价值）：数据有内在价值，但需要被发现。可以通过各种量化技术和调查技巧发现数据的价值 — 从发现消费者偏好或舆情，到按位置开展相关营销，或者识别要发生故障的设备。技术突破已经使数据存储和计算的成本大幅降低，因此能够提供充足的数据来对整个数据集进行统计分析，而以前只是对样本进行统计分析。这种技术突破使得更精准的决策成为可能。不过，发现价值还需要新的发现过程，牵涉到机敏而有见地的分析人员、业务用户和高管。真正的大数据挑战来自人本身，包括学习如何提出正确的问题、认可合作伙伴、做有根据的假设以及对行为进行预测。

➜ 大数据使用中的挑战

由大数据的基本特性（大量，高速，多样）我们可以得知，大数据体系中包含了海量的种类丰富且高时效性的数据集合，是一座价值巨大的数据金矿。但同样是因为这些特性，使大数据无法在一定时间范围内用常规

软件工具进行捕捉、管理和处理（数据集合的规模大到在获取、存储、管理、分析方面大大超出了传统数据库软件工具能力范围）。同时大数据中的单项数据的数据价值密度又很低（一个常用的说明这一特点的例子是视频监控。在连续不间断的监控过程中,可能有用的数据仅仅有一两秒钟。这意味着在一个1，2G大小的视频文件记录中，有价值的数据可能只有1，2M），挖掘大数据中的价值类似沙里淘金。由此可见，大数据技术的战略意义不在于掌握庞大的数据信息，而在于对这些含有价值的数据进行专业化处理。换而言之，如果把大数据比作一种产业，那么这种产业实现盈利的关键，在于提高对数据的“加工能力”，通过“加工”实现数据的“增值”。需要一种全新的处理模式才能使大数据成为“ **具有更强的决策力、洞察发现力和流程优化能力来适应海量、高增长率和多样化** ”的信息资产。

目前，在技术层面，对大数据的获取，存储和管理方面已经有了一套成熟的软件技术解决方案和完善的配套生态体系。而在最核心的数据价值发现方面，目前有众多的针对不同领域和应用场景的技术方向。

在对数据的处理方式这一问题上主要有如下两个技术方向：

■ **流处理** - 数据到来立即分析运算并返回结果。主要使用的技术有：
- Twitter的Storm
- Yahoo的S4
- LinkedIn的Kafka

.....

■ **批处理** - 先将数据存储，再对批量数据统一分析运算并返回结果。主要使用的技术有：
- Google 和 Haddop的MapReduce模型框架

.....

不同的数据处理方式只是解决了怎样对获取到的大数据进行操作的问题，而针对已经获取到的数据，如何从中发现数据价值，又是一个全新的领域。目前在这一领域中主要使用 ***知识发现*** 技术来解决从数据中获取知识价值的问题。**知识发现是大数据处理流程的核心,也是大数据价值生成的主要途径**。目前在技术层面，知识发现主要使用 `数据挖掘` , `机器学习` , `统计学` 等技术方法来获取知识价值。同时由于大数据领域中数据集合的大量，高速，多样性，知识发现也面临着大量的挑战（例如算法实时性,可扩展性,并行计算的实现等方面）。

从以上事实中我们可以发现，对于用户来说，大数据系统的真正价值不是大数据中的数据集合本身，而是这些海量的数据集合中的高价值密度数据（有价值的知识）。同时获取高价值密度数据的过程（知识挖掘）有相当的技术和现实方面的挑战。由此引申我们还可以发现以下问题：

1. 使用知识发现来发现知识型数据（高价值密度数据），需要付出一定的技术代价。但知识发现的效果并不一定能超过人类意识的主观判断，大多数的数据挖掘和机器学习算法只是在大规模的数据集合上执行可以量化统计的数学计算，对于具有与人类主观判断特点的分类（例如判断艺术作品或建筑的风格）并没有良好的效果。

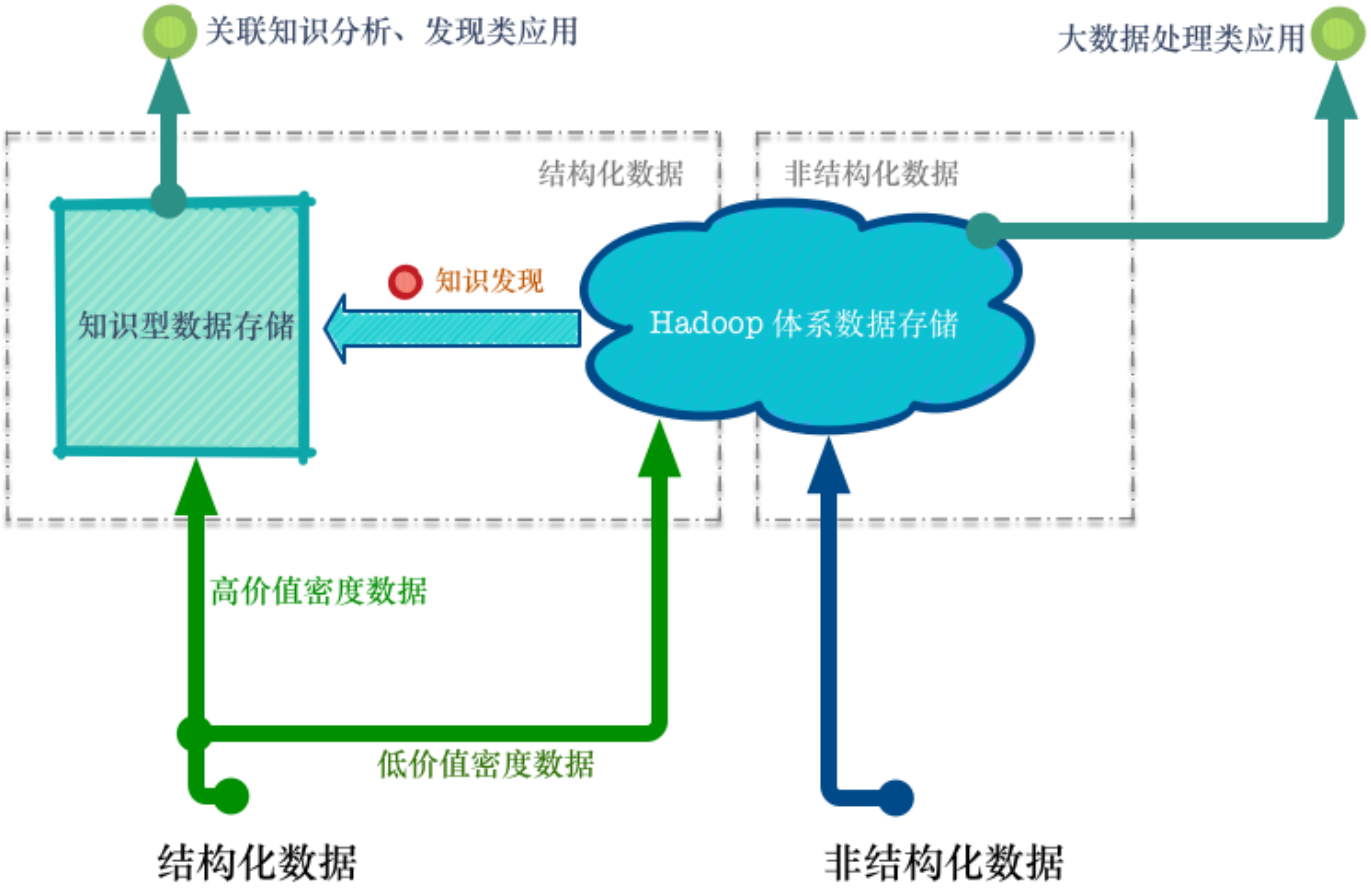
2. 一项数据的价值密度高低与否不是一个客观的绝对量值，而是主观的因人而异的评价值（例如，一个建筑的外立面效果图对于建筑设计师来说是高价值密度数据，而对于结构工程师来说可能是低价值密度数据）。

3. 很多结构化数据在特定的业务领域中天然具有较高的价值密度（例如某地区的人口统计数据对于该城市的总人口统计来说天然就是高价值密度数据），无需通过大数据体系进行知识挖掘。

4. 在相当多的领域中，高价值密度数据不是标量的客观数值本身，而是可以由这些数值本身分析发现的一些与外部世界的相互关联。

➜ 一种解决目前问题的方式

分析上述大数据领域中面临的使用问题，可以得出以下结论。即在现实应用中，需要一个如下图所示的独立于大数据系统的 **知识型数据存储**系统。在使用中，可以直接将结构化数据中的高价值密度数据以知识型数据的形式存储入库，同时可以通过知识发现技术从大数据系统中获得更多的知识型数据。



当用户执行关联知识分析以及发现类的应用时，直接从该知识型数据存储系统中获取所需的数据，并且根据自身的业务需要再加工、细化获取到的知识数据。该知识型数据存储系统还应当具有以下特性:

■ **可调整型** - 用户应当可以通过自身的主观判断来调整、修改知识数据的属性，或者细化知识数据的分类。经过调整后，针对该用户的操作，该知识数据的价值密度也应当相应变化。

■ **自适应性** - 可以针对不同使用者的主观判断来定义什么是高价值密度数据。对于不同的用户的主观操作，同一条知识数据应该具有不同的价值密度（例如在使用中，作为知识型数据，一个建筑的外立面效果图对于建筑设计师和结构工程师应该具有不同的价值）。

■ **高可用维度性** - 用户可以从自身主观意愿出发，从任意的维度（此处的维度是指用户看待问题的角度，或者知识数据的某些属性）以单独或者关联的方式来查询知识型数据，从而最大可能性的发掘出知识的价值。

■ **自主判断性** - 可以针对用户的历史操作记录推断出用户使用系统的意图以及倾向性，从而以推荐的方式自动的帮助用户获取到相关联或者相似的知识数据。

■ **与现有技术标准兼容性** - 系统应当具有与现有的数据库事实技术标准（关系数据库系统）相一致的功能，具有标准的数据访问接口，能够使用行业标准的方式（例如使用SQL语言）进行数据操作。

■ **高性能，高可用性** - 由于知识型数据存储系统实质上是大数据系统的外延，所以它应当能够支持海量数据，提供快速高效的数据访问，并且能够通过透明的硬件水平扩展来提高整体的吞吐能力和性能。

使用图论数据库作为知识型数据存储系统的核心

➜ 为什么需要使用图论实现知识型数据存储系统

为了从庞大的数据集合中检索出需要的数据，需要建立一个数据模型将这些数据集合放入其中，再通过该数据模型提供的访问方式获取数据。在众多不同的数据模型里，关系数据模型在实际应用中占据了统治地位。很多数据库产品是对关系模型做出的实现（例如Oracle、MySQL和DB2等），这些数据库统称为关系数据库管理系统（RDBMS）。关系模型本身是为了解决对具有强业务关联，强事务性要求的数据集合的访问而设计的，它本身并没有关注对海量数据的可访问性这一要求。由于关系型数据库根植于关系模型，他们具有以下两个突出的问题：

1. 数据建模复杂，缺乏灵活性，一旦关系数据库中的表结构建立并投入使用。很难再根据业务需要灵活的改变他们。
2. 大数据量和多服务器之上进行水平伸缩存在诸多限制，单台关系数据库服务器系统无法承载海量的数据集合，而对多台关系数据库服务器系统进行集群化的水平扩展技术难度较大。

在当前的大数据时代的背景下，各种数据源产生的数据量呈指数增长，而数据的多样性和灵活性使得数据之间的相互依赖和复杂度的也急剧的增加。在这个趋势前，关系数据库所具有的突出的问题又被空前的放大，在真实的大数据背景下的数据系统中产生了更多的问题。

为应对这一问题，在近几年出现了很多新项目，它们被统称为NOSQL数据库（NOSQL-databases），NOSQL数据库试图使用新的数据模型来替换关系模型，从而解决关系数据库所面临的问题。按照它们的数据模型，NOSQL数据库可以分成4类：

■ **`键-值存储库（Key-Value-stores）`** - 最小的建模单元是键-值对。可以通过key快速查询到其value。

■ **`BigTable实现（BigTable-implementations）`** - 最小建模单元是包含不同个数属性的元组。BigTable是非关系型数据库，是一个稀疏的、分布式的、持久化存储的多维度排序Map。

■ **`文档库（Document-stores）`** - 最小单元是文档。文档存储一般用类似json的格式存储，存储的内容是文档型的。这样也就有机会对某些字段建立索引，实现关系数据库的某些功能。

■ **`图形数据库（Graph Database）`** - 整个数据集合建模成一个大型稠密的网络结构。使用图论来构建数据存取的模型。

键-值存储库，***BigTable*** 以及 ***文档库*** 这三种NOSQL实现方式只是为了解决特定条件下关系模型带来的问题而做的一些技术简化（键-值存储库）或微调（BigTable，文档库）。他们并没有在大数据的尺度下提供一个对关系模型主要功能的替代。而图论提供了一种全新的数据模型，这个模型可以在特定的领域内替代关系模型在大数据的尺度下提供全面的功能替代。而在当前的技术发展和数据应用阶段，产生了大量的对于相互关联的 `关联数据 Relational Data` （例如接触、联络、关系、群体依附和集结等）的查询和探索的需求，例如：

- 发现社交网络关系中的关键影响因素、桥接实体和群体。
- 智能发现商品亲和力以增强客户体验并给出更明智、更简单的建议。
- 发现指示欺诈行为的模式及关系。

针对这类“多对多对多”的关系查询和遍历计算以及定量分析，图论模型更是具有先天的优势。

➜ 图论的基本概念和算法

图论（Graph Theory）是数学的一个分支。它以图为研究对象。图论中的图是由若干给定的点及连接两点的线所构成的图形，这种图形通常用来描述某些事物之间的某种特定关系，用点代表事物，用连接两点的线表示相应两个事物间具有这种关系。图G=（V,E）是一个二元组（V,E）使得E⊆[V]的平方，所以E的元素是V的2-元子集。集合V中的元素称为图G的定点（或节点、点），而集合E的元素称为边（或线）。通常，描绘一个图的方法是把定点画成一个小圆圈，如果相应的顶点之间有一条边，就用一条线连接这两个小圆圈，如何绘制这些小圆圈和连线时无关紧要的，重要的是要正确体现哪些顶点对之间有边，哪些顶点对之间没有边。

图形是关系规范化的一种替代技术。图形中的点对应关系模型中的实体，边（具有方向和类型，即标记和标向）对应关系模型中实体见的关系，而点和边上的属性则对应实体和关系上面的属性。除了替代关系模型外，现实世界中的大部分领域实际上都可以建模成图。例如社交类系统，推荐类系统，关联关系发现类系统等都可以便捷的使用的图形建
模。 `而本文所讨论的知识型数据存储系统，也是一个使用图论模型的绝佳场景` 。

图论的具有丰富的用途和应用场景，它跟不同领域的很多问题都有关联。针对图论数据模型，有很多相关的图论算法，例如：
- 各种类型的最短路径计算。
- 测地线（Geodesic Path）。
- 集中度测量（如PageRank、特征向量集中度、亲密度、关系度、HITS等）。

通过使用这些算法，可以非常容易的实现在关系模型中难以轻易实现的运算，例如使用两点间的最短路径算法可以轻易的实现SQL查询语言中较复杂的经典的航班最短飞行问题（如何用最短的班次实现在两个不同地点之间的航班飞行）。

➜ 现有主流图形数据库产品概况

目前已经有相当多的NOSQL数据库产品实现了基于图论的数据模型，其中应用范围最广的有几下五种：

■ ***Neo4j*** - Neo4j是一个用Java实现、高度可扩展的本地纯图形数据库。数据以一种针对图形网络进行过优化的格式保存在磁盘上。Neo4j的内核是一种极快的图形引擎，具有数据库产品期望的所有特性，如恢复、两阶

段提交、符合XA等。Neo4j既可作为无需任何管理开销的内嵌数据库使用；也可以作为单独的服务器使用，在这种使用场景下，它提供了广泛使用的REST接口，能够方便地集成到基于PHP、.NET和JavaScript的环境里。Neo4j是目前适用范围最为广泛的图形数据库产品。

■ ***OrientDB*** - OrientDB是一种双模式的数据库，它是一种兼5具文档数据库的灵活性和图形数据库管理关联关系能力的可深层次扩展的文档-图形数据库管理系统。可选无模式、全模式或混合模式下。支持许多高级特性，诸如ACID事务、快速索引，原生和SQL查询功能。可以JSON格式导入、导出文档。OrientDB还是一个分布式数据库系统，可以组成主主数据库集群，实现透明的数据库水平扩展。OrientDB是一个用户活跃度仅次于Neo4j的图形数据库

■ ***Titan*** - Titan 是一个在服务器集群中搭建的分布式的图形数据库，特别为存储和处理大规模图形而优化。集群很容易扩展以支持更大的数据集，Titan有一个很好的插件式性能，这个性能让它搭建在一些成熟的数据库技术上像 Apache Cassandra、Apache HBase、 Oracle BerkeleyDB。插件式索引架构可以整合 ElasticSearch 和Lucene技术。内置实现 Blueprints graph API，支持 TinkerPop所有的技术。

■ ***Virtuoso*** - Virtuoso 是一种多模式的数据库，它支持关系数据库，图形数据库以及文档库三种数据模型。Virtuoso提供了复杂的SQL\XML\RDF数据库管理功能。支持工业标准的交互查询协议、API 和数据格式。包括：ODBC, JDBC, OLE-DB, ADO.NET, XMLA, SQL, SPARQL, XQuery, SOAP, HTTP, WebDAV, SyncML, Atom (Publishing and Syndication), RSS, RDF等。

■ ***ArangoDB*** - ArangoDB 是一种多模式的数据库，它同时支持文档库，图形数据库和键-值存储库三种数据模型。ArangoDB是一个高性能的数据库,支持类SQL的查询语言AQL以及JavaScript和Ruby扩展。由于ArangoDB是模式自由的元数据模式，跟其它文档型数据库相比，ArangoDB占用的存储空间更少。ArangoDB支持主从集群，可以构建数据库集群。

结合图论数据库与传统数据仓库概念实现知识型数据存储系统

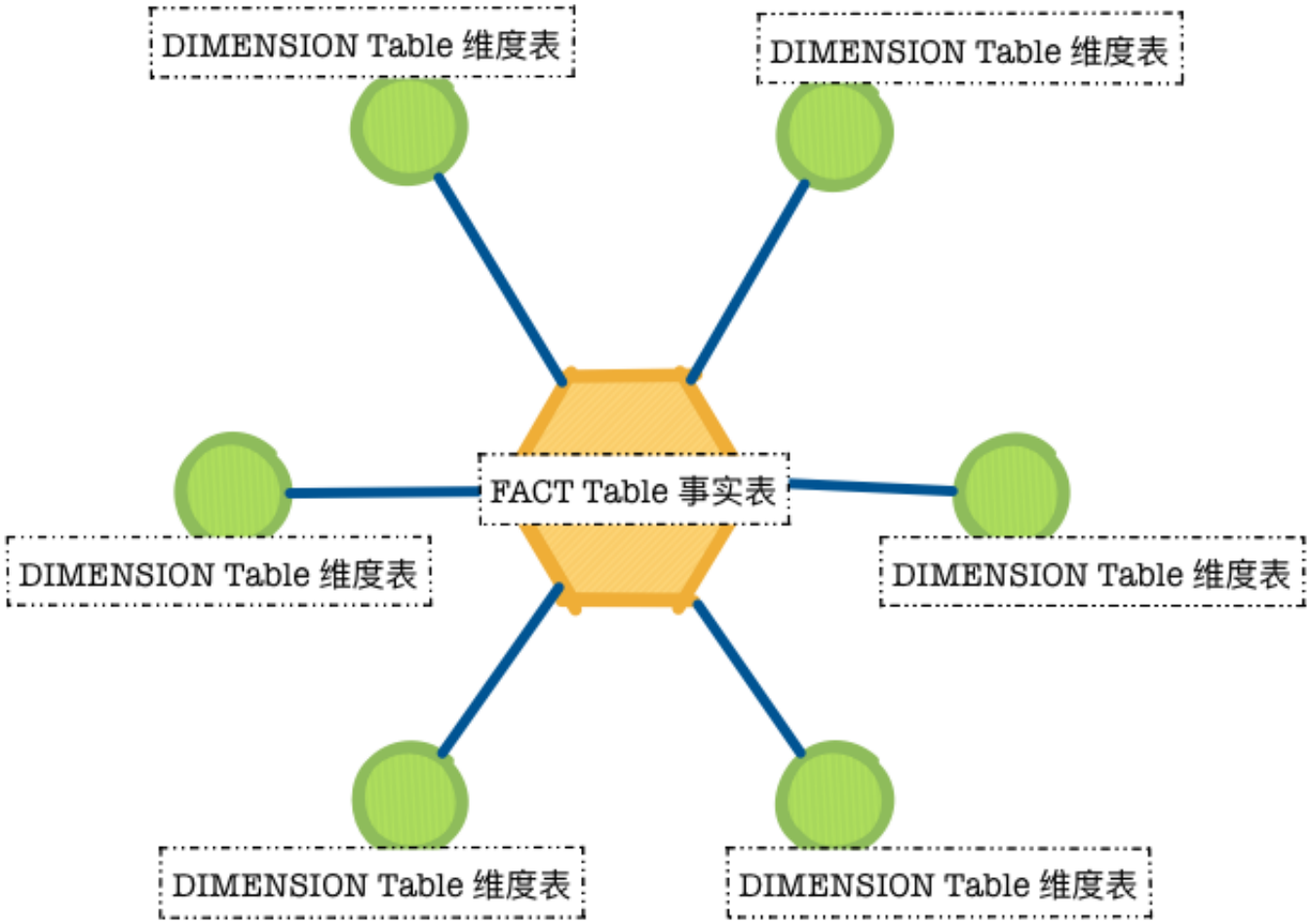
➜ 结合使用图形数据模型和数据仓库星型模式概念来设计知识型数据存储系统

根据我们对知识型数据存储系统的定义可以得知：知识型数据存储系统的主要目标是为用户提供高价值密度的知识数据，而知识数据价值密度的高低取决于不同用户的主观需求，并不是一个客观的确定数值。为了获取所需的知识数据，用户必须输入若干约束条件做为筛选知识的依据。**用户能够设定的约束条件组合形式越灵活多变，知识型数据存储系统的准确性就越高。通过指定的知识数据获取相关联的其他知识数据（高价值密度数据）的能力越强，知识型数据存储系统的效率就越高，而这一点正是图形数据模型所擅长的(多节点，多重路径的关联计算)。** 但另一方面， 存储在系统中的知识数据本身是一些标量数据的组合，实质上是一些客观存在的事实。由于知识数据集合本身是客观数值，同时又是所有的用户共享的，这意味着用来筛选知识数据的约束条件也应该是一个可描述的，有限范围内的度量的集合。为了逻辑性、系统性的在图形数据模型中实现这一特点，我们可以借鉴 `传统数据仓库` (基于关系数据库存储原理的数据仓库系统) 技术中 `星型模型` 这一概念。以下是传统数据仓库中和星型模型相关的几个概念的简要定义：

■ ***事实（Fact）*** - 事实是数据仓库中的信息单元，也是多维空间中的一个单元，受分析单元的限制。事实存储于一张表中（当使用关系数据库时）或者是多维数据库中的一个单元。每个事实包括关于事实（例如收入、价值、满意记录等）的基本信息，并且与维度相关。

■ ***维度（Dimension）*** - 维度是绑定由坐标系定义的空间的坐标系的轴线。数据仓库中的坐标系定义了数据单元，其中包含事实。坐标系的一个例子就是带有 x 维度和 y 维度的 Cartesian（笛卡尔）坐标系。在数据仓库中，时间总是维度之一，此外具有业务含义的描述性信息也是定义维度的主要属性。

■ ***星型模型（Star Schema）*** - 星型模型是一种使用关系数据库实现多维分析空间的模式。它是传统数据仓库领域中一种主要的多维分析的建模方式。星型模型是一种多维的数据关系，由一个事实表（Fact Table）和一组维表（Dimension Table）组成。每个维表都有一个维作为主键，所有这些维的主键组合成事实表的主键。事实表的非主键属性称为事实（Fact），它们一般都是数值或其他可以进行计算的数据；而维大都是文字、时间等类型的数据，按这种方式组织好数据就可以按照不同的维（事实表主键的部分或全部）来对这些事实数据进行聚集计算（例如求和、求平均、计数，百分比等）。这样就可以从不同的角度数字来分析业务主题的情况。



上图是一个星型模型的结构概念图示，由此图我们可以发现，在宏观上星型模型内部的组成部分实质上具有图形的特点，只是在具体的数据查询和获取操作中传统数据仓库的星型模型使用了关系模型。由此启发，我们可以使用图形数据模型来设计一个"改良版"的的基于图形数据库的星型模型来服务于知识型数据存储系统的需要。以下是该模型的主要技术特征：

1. 设计逻辑层面使用 ***事实（Fact）*** 来代表知识型数据存储系统中的 ***知识数据*** 这一概念。
2. 设计逻辑层面使用 ***维度（Dimension）*** 来代表知识型数据存储系统中用来筛选知识数据的 ***约束条件*** 这一概念。

3. 系统实现层面使用图论模型中的 `节点` 来实现事实，***`事实节点`*** 用来标识在知识数据中客观存在的标量数值，这些数值以属性的形式存储在事实节点中。

4. 系统实现层面使用图论模型中的 `节点` 来实现维度， ***`维度节点`*** 用来标识用户在获取高价值密度知识数据时所使用的约束条件。维度节点的属性中只储存与使用约束条件相关的数据和信息，不储存任何与知识数据相关的数据和信息。

5. 系统实现层面使用图论模型中的 `边` 来描述事实与维度间的 ***`关系`*** 。通过在边上设置方向（有向边）和不同的属性，可以简洁的使用图论算法筛选出 ***`事实节点`*** 与 ***`维度节点`*** 之间的复杂关联关系。

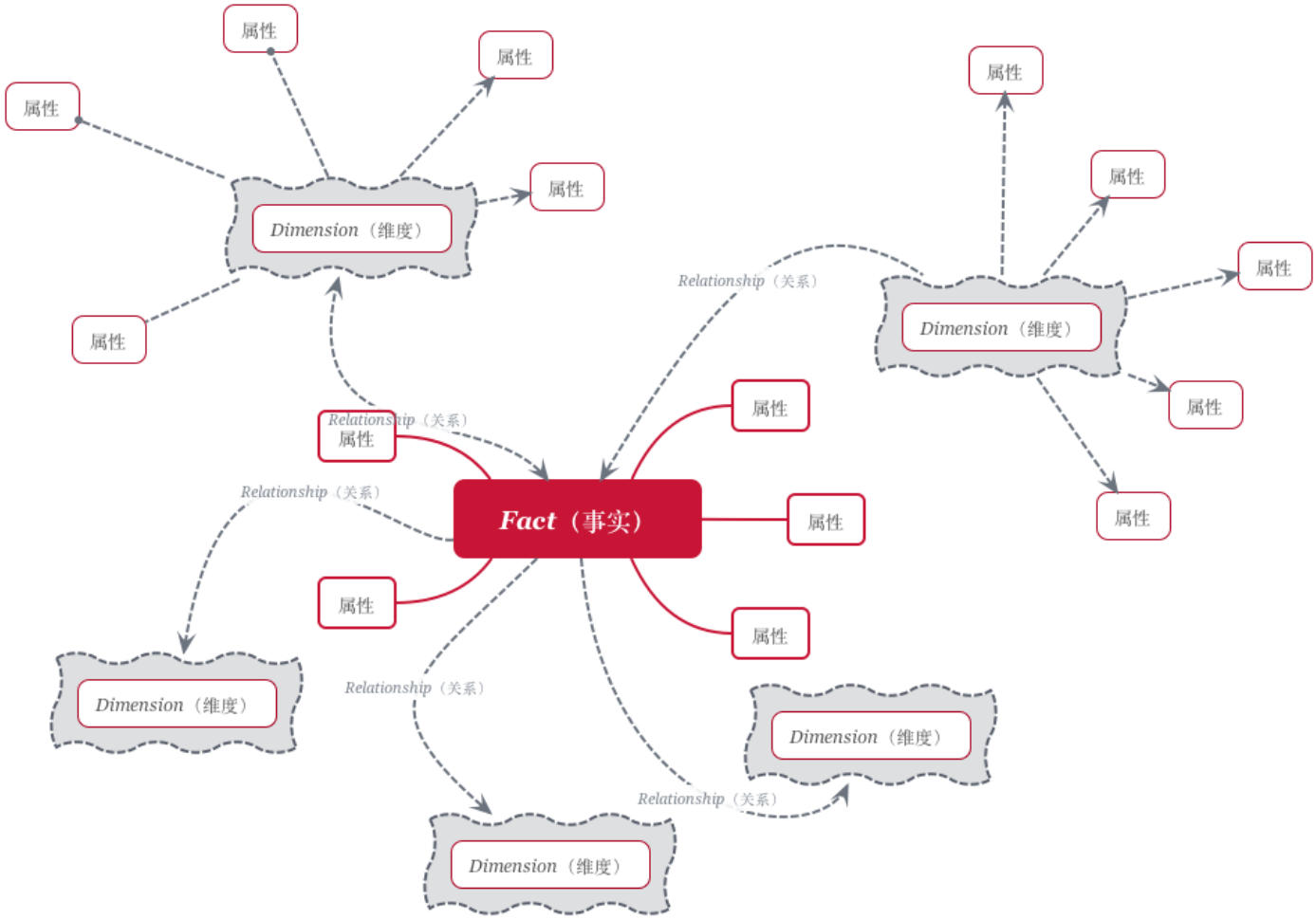
6. 用户可以通过对特定的 ***`维度节点`*** 执行关联发现操作（图论算法中的多节点，多重路径的关联计算）来获取所需的高价值密度知识数据（图论算法优先获取到的 ***`事实节点`*** 代表了高价值密度的知识数据）。

7. 用户使用中可以通过对 ***`事实节点`*** 的属性值进行检索的方式来筛选出需要的知识数据（类似传统的关系数据库中的数据查询）。

8. 当获取到特定的 ***`事实节点`*** 后，用户可以根据自身的主观判断来调整、优化与其相关的 ***`维度节点`*** 。从而提高知识数据在特定约束条件下的价值密度的准确性。

9. 通过分析用户对 `维度节点` 和维度-事实间的 `关系` 的使用情况，可以推测出用户对使用系统的意图以及倾向性，从而为后续的数据推荐等操作提供数据和技术支撑。

下图以一个 ***`事实节点`*** 为中心演示了基于图形数据库的星型模型的概念



使用上述的基于图形数据库的星型模型来设计实现的知识型数据存储系统具有如下的优点：

■ 通过使用基于图形数据库的实现方式，可以消除关系模型在大规模数据集合和复杂数据类型及关联关系的应用场景下所无法解决的实现复杂性以及性能不足的问题。为在大数据环境下获取高价值密度知识数据提供可靠的技术保障。

■ 基于图数据模型的特点以及图论算法的支持，对多维度条件下的关联信息查找和发现（获取高价值密度知识数据的过程）提供了比传统关系模型更加强大的功能支持以及更加快捷的性能支持。

■ 通过借鉴已经久经实践验证的数据仓库领域的概念和设计理念，降低了系统设计的理论和实践风险。同时使用了行业熟悉的术语和操作方式来获取高价值密度知识数据数据，有利于系统的大规模推广和使用。

高价值密度知识获取数据平台的架构设计概述

➜ 高价值密度知识获取数据平台的功能

高价值密度知识获取数据平台是一个运行在大数据环境下，对各种异构数据源中的数据进行集中分析并将分析结果使用 `知识型数据存储库` 知识化并持久化保存的数据分析体系结构。它是一套异构的基于TCP/IP网络传输协议的信息系统。 该系统使用 Apache Kafka 分布式消息系统作为核心信息传输、交换枢纽。它提供了一套标准化的数据输入方式和封装格式,通过使用标准化的数据封装格式，实现了与操作系统无关，编程实现

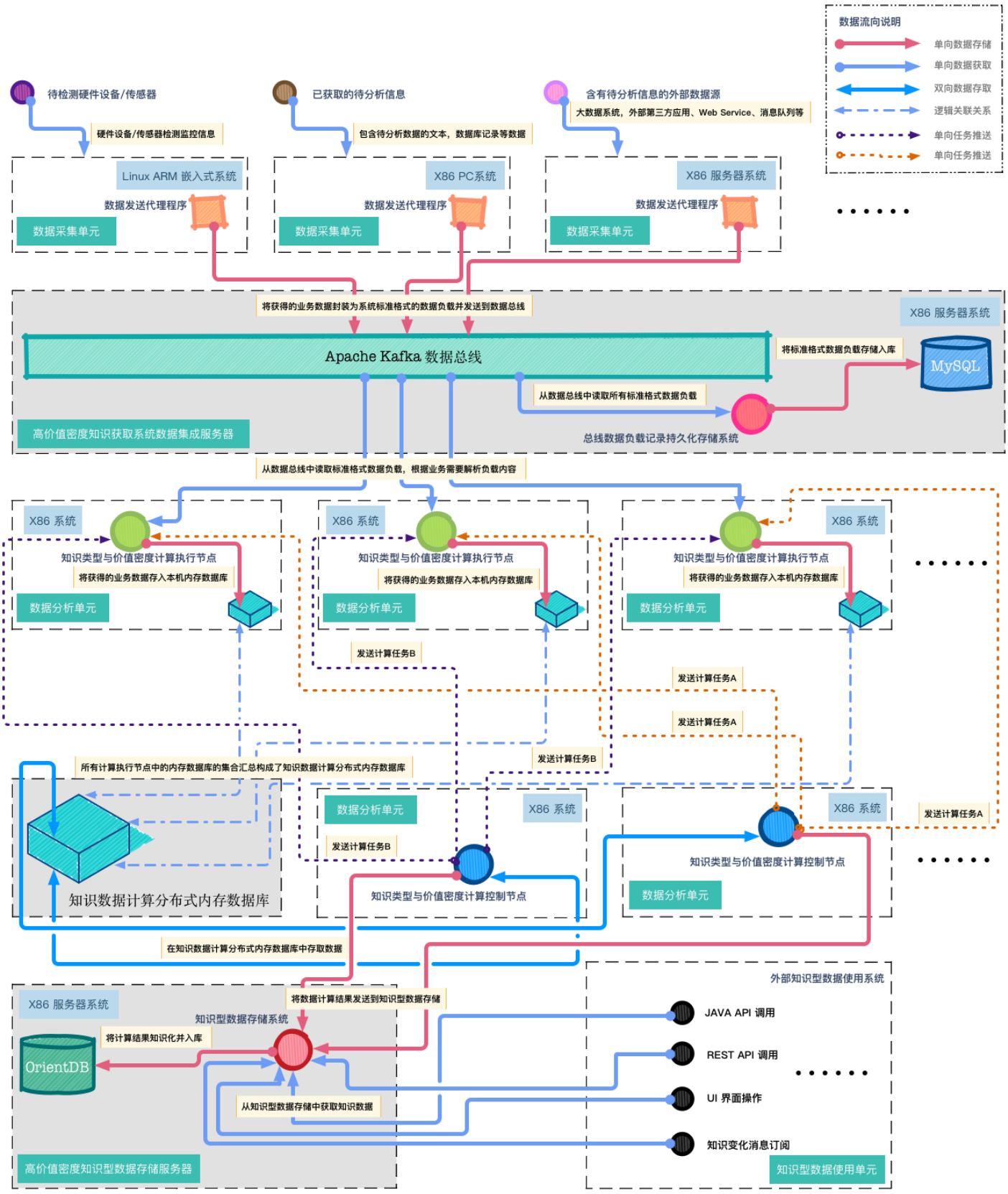
无关的数据交换体系。

各种类型的数据源（ `数据采集单元` ）实时使用这一标准的数据输入方式和封装格式将不同类型和结构的业务数据封装成标准格式的 `数据负载` 输入到数据平台的 Kafka 数据总线中。为各类不同业务内容服务的数据分析和计算系统（ `数据分析单元` ）实时监听Kafak消息系统中相关的消息主题，使用统一的技术标准从数据总线中获取标准格式的 `数据负载` 并根据需要解析负载获取特定业务信息。

当数据分析单元对 `数据负载` 中的业务信息完成分析计算后，知识型数据存储系统可以将分析计算的结果知识化（应用图形理论存储实体之间的关系信息，使用 `维度` 、 `事实` 和 `关系` 等概念以分解关联标签的形式将分析计算的结果泛化成为图形数据库中的节点并与业务相关的其他节点相互关联）。通过使用知识型数据存储系统，各种类型的外部知识型数据使用系统可以处理大量复杂、互连接、低结构化的知识型数据记录，从而获取到使用关系型数据库无法获取到（查询复杂、缓慢、超出预期）的业务领域高价值密度知识。

➜ 系统架构概述

下图为高价值密度知识获取数据平台系统架构框图：



➜ 数据采集单元

运行在各类信息系统硬件环境中的各种类型的为高价值密度知识获取数据平台提供输入数据的数据源称为数据采集单元。数据采集单元通过内置的标准化数据发送代理程序使用标准的数据输入方式和封装格式将不同类型和结构的业务数据封装成标准格式的 `数据负载` 并输入到数据平台的 Kafka 数据总线中。

➔ 数据分析单元

数据分析单元的主要功能是从 Kafka 数据总线中获取数据负载，并将数据负载中的业务数据信息分析、计算以产生最终判断结论，并将该结论发送到知识型数据存储系统中以创建领域知识型数据记录。数据分析单元具有群组通讯功能，每一个数据分析单元原则上都运行在一个单独的计算机硬件环境中。若干个数据分析单元共同构成一个数据分析单元网络集群，该集群内的数据分析单元可以互相识别，当新的数据分析单元启动或旧有数据分析单元关闭时，该集群内的所有其他分析单元都会自动获取相应消息提示信息。

数据分析单元分为以下两类节点：

■ **知识类型与价值密度计算执行节点**：计算执行节点的主要功能是构建分析计算所用数据集合并为数据分析提供计算能力。每一个计算节点都会从 Kafka 数据总线中获取所需的部分数据负载并将该负载解析成为数据记录，用本机物理内存构建内存数据库并将数据记录存储在该内存数据库中。同一个数据分析单元网络集群中的所有计算节点中的内存数据库会自动的组合构成一个分布式的基于网络的 `知识数据计算内存数据库` 。集群中的所有计算执行节点和计算控制节点都可以透明的操作该分布式内存数据库中的所有数据。计算节点的另一个主要功能是分布式的执行计算控制节点发送的计算任务。

■ **知识类型与价值密度计算控制节点**：计算控制节点的主要功能是根据业务需求统筹执行数据分析计算并将执行结果发送到知识型数据存储系统中。计算控制节点能够根据业务需求选择所需的算法程序逻辑，并将该算法代码作为一个计算任务发送到若干适合的计算执行节点中。在这些计算执行节点中会并行的执行该计算任务，当所有计算执行节点中的运算操作完成后，计算结果集合会返回到计算控制节点中（有同步返回和异步返回两种方式）。计算控制节点可以再次提炼加工这一计算结果集合，并将最终产出结果发送到知识型数据存储系统中创建最终的领域知识型数据记录。

➔ 数据集成服务器

是高价值密度知识获取数据平台的数据中转枢纽服务器。该服务器中部署 Kafka 消息系统以及 MySQL 数据库系统。高价值密度知识获取数据平台中的所有数据采集单元和数据分析单元都访问该中央数据集成服务器以获取数据访问服务。将 Kafka 标准格式的数据负载持久化存储入 MySQL 数据库的总线数据负载记录持久化存储系统也运行在数据集成服务器中。由于数据集成服务器是系统核心数据处理节点和潜在的性能瓶颈，可以通过水平扩展的方式建立服务器集群以提高高价值密度知识获取数据平台的整体性能。

➔ 高价值密度知识型数据存储服务器

是本文中讨论的**使用图形数据库存储引擎并参考数据仓库星型模型所设计的知识型数据存储系统**实际部署的服务器系统。该服务器系统存储来自数据分析单元的知识型数据信息。并为外部的知识型数据使用单元提供各种接口类型的数据服务。