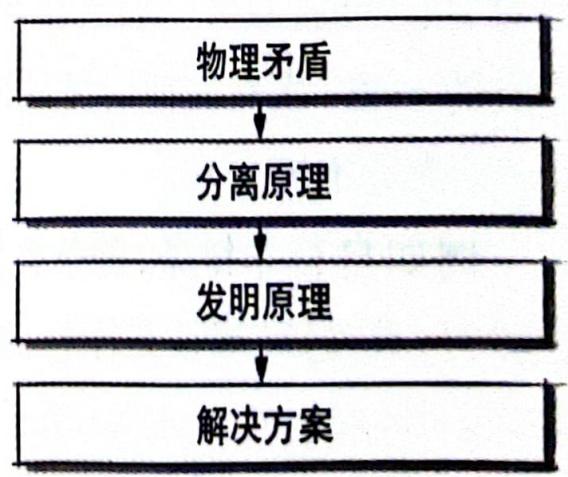
7.2.2 分离原理与发明原理的对应关系

分离原理与解决技术矛盾的40个发明原理之间有密切关联，每一个分离现有多个发明原理与之对应，对应关系如表7.2.2所示。只要确定物理矛盾及其分离原理类型，就可用发明原理及相关案例帮助设计者快速确定新方案。

表7.2.2 分离原理与发明原理关系

|  |  |  |
| --- | --- | --- |
|  | 分离原理 | 发明原理 |
| 空间分离 | | 分割原理（01）；抽取原理（02）；非对称原理（04）；嵌套原理（07）；局部特性原理（03）；多维化原理（17）；中介原理（24）；柔壳或薄膜原理（30） |
| 时间原理 | | 预先反作用原理（09）；预先作用原理（10）；预置防范原理（11）；动态化原理（15）；抛弃与再生原理（34）；振动原理（18）；周期性作用原理（19）；有效作用持续原理（20） |
| 关系分离 | | 局部特性原理（03）；多维化原理（17）；周期性作用原理（19）；多孔原理（31）；变色原理（32）；复合材料原理（40） |
| 系统分离 | | 分割原理（01）；合并原理（05）；等势原理（12）；同质原理（33）；复合材料原理（40）；多功能原理（06）；变害为利原理（22）；反馈原理（23） |
| 方向分离 | | 非对称原理（04）；复合材料原理（40）；多孔原理（31）；曲面化原理（14）；多维化原理（17）；变色原理（32）；嵌套原理（07） |

**7.2.3 利用分离原理解决物理矛盾**

解决物理矛盾步骤如图7.2.1所示。首先对问题及其背景进行分析，写出物理矛盾；然后分析矛盾分离的可能性，确定适用的分离原理；再分析发明原理，思考可能的解决方案；最后根据相关知识，确定具体的解决方案。

1．空间分离

在应用这种分离方法时，首先要分析是否图7.2.1 物理矛盾解决步骤在技术系统所有的空间位置都需要矛盾双方。通常描述此类矛盾的方式是“在哪里”，即“在哪里需要······（一种需求），在哪里需要······（另一种需求）”。

### 例7.2.1：动态负载均衡在服务器集群中的应用

#### 背景介绍

在计算机科学与技术领域，动态负载均衡是服务器集群管理中的关键技术，用于确保服务器的高效运作和资源的优化使用。然而，在实际应用中，动态负载均衡存在物理矛盾，需要解决如何在不同时间和空间满足相反的需求。

#### （1）写出物理矛盾

服务器集群的负载均衡要高效，因为需要保证资源的最优分配和响应速度； 但是，负载均衡要低效，因为需要防止单点故障和资源过度使用。

#### （2）分析适用的分离原理

服务器集群中，负载均衡的效率需求在不同的时间和空间有所不同。适用时间分离和空间分离原理。

#### （3）分析适用的发明原理（表7.2.3）

| **发明原理** | **分析** | **可能的解决方案** |
| --- | --- | --- |
| 分割原理（01） | 负载均衡机制能分开吗？ | 使用多层负载均衡架构 |
| 局部特性原理（03） | 能否对不同节点进行不同配置？ | 根据节点功能设置不同的负载均衡策略 |
| 嵌套原理（07） | 能否将负载均衡器嵌入不同的网络层次？ | 在应用层和网络层分别进行负载均衡 |
| 非对称原理（04） | 能否设计非对称的负载均衡策略？ | 对高流量和低流量节点采用不同的均衡算法 |

#### （4）具体解决方案

**应用分割原理（01）**： 将负载均衡机制分开，可以设计成多层负载均衡架构。例如，前端使用Nginx进行初步负载均衡，后端使用HAProxy进行深度负载均衡。

**图7.2.2 多层负载均衡架构**

[客户端]

|[Nginx]

|[HAProxy]

|[服务器集群]

**应用局部特性原理（03）**： 对不同节点进行不同配置，根据节点的功能和角色设置不同的负载均衡策略。例如，对数据库服务器和应用服务器分别采用适应性负载均衡算法。

**图7.2.3 分布式配置**

[负载均衡器]

/ \[数据库服务器] [应用服务器]

**应用嵌套原理（07）**： 将负载均衡器嵌入不同的网络层次，例如在应用层和网络层分别进行负载均衡，在不同层次进行优化。

**图7.2.4 网络层和应用层负载均衡**

css

复制代码

[网络层负载均衡]

|

[应用层负载均衡]

|

[服务器集群]

**应用非对称原理（04）**： 设计非对称的负载均衡策略，对高流量和低流量节点采用不同的均衡算法。例如，高流量节点采用加权轮询算法，低流量节点采用随机算法。

**图7.2.5 非对称负载均衡策略**

[高流量节点] - 加权轮询算法[低流量节点] - 随机算法

### 结论

通过应用TRIZ剪裁工具中的分离和发明原理，我们可以有效解决服务器集群动态负载均衡中的物理矛盾，优化系统性能并提高资源利用效率。每种解决方案都针对特定的矛盾，通过合理分离需求和功能，实现系统的最佳配置和运行。

### 例7.2.2：计算机存储系统中的动态存储空间管理

#### 背景介绍

在计算机科学与技术领域，存储系统的管理是一个关键问题。在某些应用场景中，存储空间需要在使用时尽可能大，以存储更多的数据；而在不使用时，希望存储空间占用尽可能小，以提高系统的灵活性和效率。

#### （1）写出物理矛盾

存储空间要大，因为需要存储更多的数据； 但是，存储空间要小，因为在不使用时需要节省资源，提高系统效率。

#### （2）分析适用的分离原理

存储空间使用时要大；不使用时要小。适用时间分离原理。

#### （3）分析适用的发明原理（表7.2.4）

| **发明原理** | **分析** | **可能的解决方案** |
| --- | --- | --- |
| 动态化原理（15） | 存储空间能否动态调整？ | 动态分配存储（Dynamic Memory Allocation） |
| 抛弃与再生原理（34） | 存储空间用完后能否释放？ | 自动垃圾回收（Garbage Collection） |

#### （4）具体解决方案

**应用动态化原理（15）**： 存储系统可以设计为动态调整存储空间。例如，采用动态内存分配技术，根据应用需求实时分配和释放存储空间。

**图7.2.4 动态内存分配示意图**

[内存池]

|[动态分配器] <---> [应用程序1]

| |[动态分配器] <---> [应用程序2]

动态内存分配器根据应用程序的需要，动态分配和释放内存空间，确保在需要时有足够的存储空间，而在不需要时释放多余的存储空间。

**应用抛弃与再生原理（34）**： 存储空间用完后可以自动释放。例如，采用垃圾回收机制（Garbage Collection），在程序运行过程中自动检测和回收不再使用的存储空间。

**图7.2.5 自动垃圾回收示意图**

[应用程序]

|[内存分配]

|[垃圾回收器] <---> [内存池]

垃圾回收器在后台运行，自动回收不再使用的内存空间，确保内存资源的高效利用。

### 总结

通过应用TRIZ剪裁工具中的时间分离和发明原理，我们可以有效解决计算机存储系统中的动态存储空间管理问题。具体的解决方案包括动态内存分配和自动垃圾回收，这些方法能够在存储空间使用时提供足够的容量，在不使用时释放多余的资源，从而优化系统性能和资源利用效率。

### 例7.2.3：存储与计算在数据库系统中的应用

#### 背景介绍

在计算机科学与技术领域，数据库系统需要既高效存储数据，又快速处理查询。在实际应用中，存储和计算之间存在矛盾：为了高效存储，数据应紧凑排列，减少存储空间；但为了快速处理查询，数据应分散排列，便于快速访问。

#### （1）写出物理矛盾

数据存储要紧凑，因为需要节省存储空间； 但是，数据存储要分散，因为需要快速处理查询。

#### （2）分析适用的分离原理

对于存储，数据应紧凑；对于查询，数据应分散。适用关系分离原理。

#### （3）分析适用的发明原理（表7.2.5）

| **发明原理** | **分析** | **可能的解决方案** |
| --- | --- | --- |
| 局部特性原理（03） | 存储和查询能否在不同的结构中处理？ | 分层存储结构，分别优化存储和查询性能 |
| 复合材料原理（40） | 数据存储能否使用复合技术？ | 使用混合存储技术，结合内存和磁盘存储 |

#### （4）具体解决方案

**应用局部特性原理（03）**： 存储和查询可以在不同的结构中处理。例如，采用分层存储结构，分别优化存储和查询性能。

**图7.2.6 分层存储结构**

[高速缓存 (RAM)]

|[固态硬盘 (SSD)]

|[传统硬盘 (HDD)]

高速缓存（RAM）用于快速处理查询，固态硬盘（SSD）用于高效存储，传统硬盘（HDD）用于长期数据保存。这种分层存储结构结合了不同存储介质的优点，满足了存储和查询的不同需求。

**应用复合材料原理（40）**： 数据存储可以使用混合技术，结合内存和磁盘存储。例如，采用混合存储技术，结合内存和磁盘存储，将频繁访问的数据保存在内存中，其他数据保存在磁盘中。

**图7.2.7 混合存储技术**

[内存存储 (In-Memory Storage)]

|[磁盘存储 (Disk Storage)]

内存存储用于快速访问频繁查询的数据，磁盘存储用于高效存储大量数据。这种混合存储技术可以同时满足高效存储和快速查询的需求。

### 总结

通过应用TRIZ剪裁工具中的关系分离和发明原理，我们可以有效解决数据库系统中存储与计算的物理矛盾。具体的解决方案包括分层存储结构和混合存储技术，这些方法能够在不同的子系统中分别优化存储和查询性能，从而提升整个数据库系统的效率和性能。

例7.2.4：自行车传动链条（系统和子系统分离）。

（1）写出物理矛盾。

传动链条需要软，因为传动链要可靠连接曲柄链轮和后链轮；

但是，传动链条需要硬，因为传动链要传递动力。

（2）分析适用的分离原理。

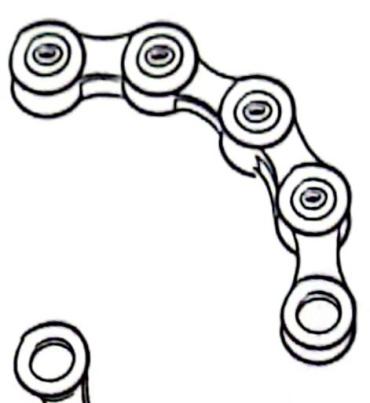
从整体上（系统）看，传动链条应该是软的，便于连接；从局部（子系统或组件）看，传动链应该是硬的，可以传递动力，工作可靠。适用系统分离原理。

（3）分析适用的发明原理（表7.2.6）。

表7.2.6 例7.2.4分析表

|  |  |  |
| --- | --- | --- |
| 发明原理 | 分析 | 可能的解决方案 |
| 分割原理（01） | 传动链能否分成相互独立的部分？ | 分割成独立的部分 |
| 合并原理（05） | 能否将相似的部分合并？ | 将独立部分柔性连接 |
| : | .  ·  · | ·  .  . |

（4）具体解决方案。

如图7.2.7所示，根据分割原理（01）和合并原理（05）的启示，把传动链分成相互独立的部分：链板、链销和滚圈，它们的材料都是刚性的；组合后各链板可相对运动，链条整体是柔性的。

**键销（轴）**

图7.2.7 链条及其组成

5．方向分离

在应用这种分离方法时，首先要分析是否在技术系统所有的方向上都需要矛盾双方。如果矛盾一方在某一方向存在，而另一方可在其他方向存在，就可考虑使用方向分离。

通常描述此类矛盾的方式是“在哪个方向”，即“在什么方向······（一种需求），在另一方向······（另一种需求）”。

### 例7.2.4：计算机处理器中的热管理（系统和子系统分离）

#### 背景介绍

在计算机科学与技术领域，处理器的热管理是一个重要问题。处理器需要有效散热，以保证稳定运行；但同时需要保持一定的热量，以确保高性能运行。这里存在一个物理矛盾。

#### （1）写出物理矛盾

处理器需要散热，因为要防止过热； 但是，处理器需要保持一定的热量，因为要保证高性能运行。

#### （2）分析适用的分离原理

从整体上（系统）看，处理器应该散热，以防过热；从局部（子系统或组件）看，处理器应该保持一定的热量，以保证高性能运行。适用系统分离原理。

#### （3）分析适用的发明原理（表7.2.6）

| **发明原理** | **分析** | **可能的解决方案** |
| --- | --- | --- |
| 分割原理（01） | 散热和保温能否分成相互独立的部分？ | 分割成独立的热管理组件 |
| 合并原理（05） | 能否将相似的部分合并？ | 将散热和保温组件柔性集成 |

#### （4）具体解决方案

如图7.2.7所示，根据分割原理（01）和合并原理（05）的启示，可以将处理器的热管理系统分成相互独立的部分：散热片、风扇和热管。散热片和风扇负责快速散热，热管负责将热量均匀分布。组合后，各组件可以相对独立工作，整体实现高效热管理。

**图7.2.7 处理器热管理系统及其组成**

[处理器]

|[热管] <---> [散热片]

|

[风扇]

**散热片和风扇**：

* 1. 散热片由高导热材料制成，吸收处理器热量，并通过风扇将热量散发到空气中。
  2. 风扇可以根据处理器温度自动调节转速，以保证高效散热。

**热管**：

* 1. 热管负责将处理器产生的热量均匀分布到散热片上，以提高散热效率。
  2. 热管内部填充高导热材料，确保快速传导热量。

### 总结

通过应用TRIZ剪裁工具中的系统分离和发明原理，我们可以有效解决计算机处理器中的热管理问题。具体的解决方案包括分割和合并不同的热管理组件，如散热片、风扇和热管，这些方法能够在不同的子系统中分别优化散热和保温功能，从而提升处理器的性能和稳定性。

**7.3 其他物理矛盾解决方法**

7.2节介绍了利用分离原理解决物理矛盾的方法。当遇到分离原理不能解决物理矛盾时，可以尝试其他两种方法：满足矛盾需求或绕过矛盾需求。

**7.3.1 满足矛盾需求**

满足矛盾需求，即在同时满足矛盾对立需求，从而解决物理矛盾。如图7.3.1所示，阴影部分为同时满足对立需求。

适用于此种方法的发明原理如下：

反向作用原理（13），相变原理（36），热膨胀原理（37），替代机械系统原理（28），改变状态原理（35），强氧化原理（38），惰性环境原理（39）。