



ILM 如何在对象的整个生命周期内运行 StorageGRID 11.5

NetApp
April 11, 2024

目录

ILM 如何在对象的整个生命周期内运行	1
如何载入对象	2
对象的存储方式（复制或纠删编码）	8
如何确定对象保留	16
如何删除对象	19

ILM 如何在对象的整个生命周期内运行

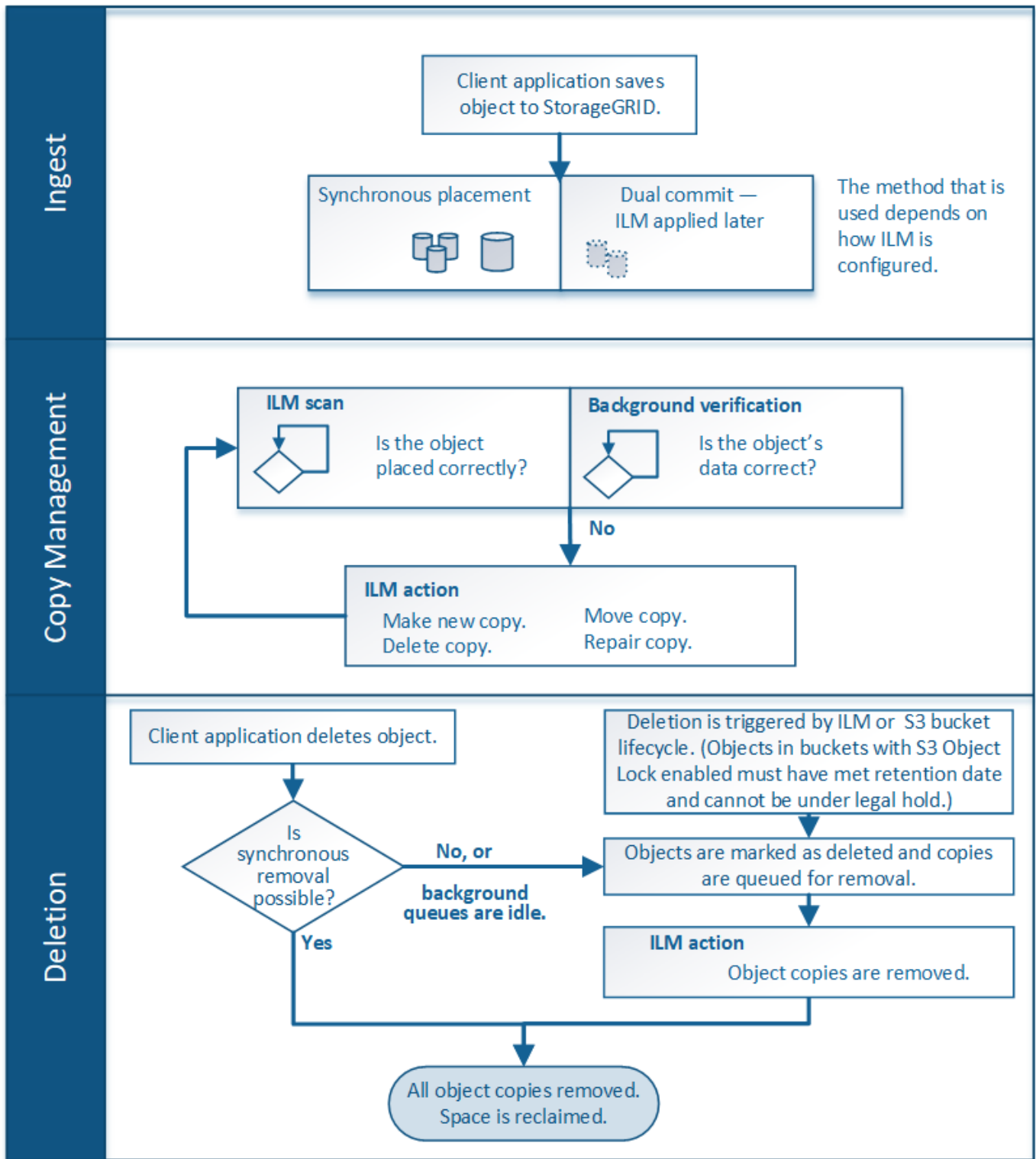
了解 StorageGRID 如何在对象生命周期的每个阶段使用 ILM 管理对象，有助于您设计更有效的策略。

- * 载入 *：当 S3 或 Swift 客户端应用程序建立连接以将对象保存到 StorageGRID 系统时，载入开始；当 StorageGRID 向客户端返回 "ingest successful" 消息时，载入完成。根据 ILM 要求的指定方式，在载入期间通过立即应用 ILM 指令（同步放置）或创建临时副本并稍后应用 ILM（双提交）来保护对象数据。
- * 副本管理 *：创建 ILM 放置说明中指定的对象副本数量和类型后，StorageGRID 可管理对象位置并防止对象丢失。
 - ILM 扫描和评估：StorageGRID 会持续扫描网格中存储的对象列表，并检查当前副本是否符合 ILM 要求。如果需要不同类型，数字或位置的对象副本，StorageGRID 会根据需要创建，删除或移动副本。
 - 后台验证：StorageGRID 持续执行后台验证以检查对象数据的完整性。如果发现问题，StorageGRID 会自动在满足当前 ILM 要求的位置创建一个新的对象副本或替换的擦除编码对象片段。请参见有关 StorageGRID 监控和故障排除的说明。
- * 对象删除 *：从 StorageGRID 系统中删除所有副本后，对象管理将结束。可以根据客户端的删除请求删除对象，也可以通过 ILM 删除对象或因 S3 存储分段生命周期到期而删除对象。



如果已启用 S3 对象锁定的存储分段中的对象处于合法保留状态，或者已指定保留日期但尚未满足，则无法删除这些对象。

该图总结了 ILM 在对象的整个生命周期中的运行方式。



相关信息

"监控和放大；故障排除"

如何载入对象

StorageGRID 通过执行同步放置或执行与对象匹配的ILM规则中指定的双提交来在载入期间保护对象。

当S3或Swift客户端将对象存储到网格时、StorageGRID 将使用以下两种方法之一载入对象：

- 同步放置：StorageGRID 会立即创建满足ILM要求所需的所有对象副本。创建所有副本后、StorageGRID 会向客户端发送"ingest successful"消息。

如果StorageGRID 无法立即创建所有对象副本(例如、由于所需位置暂时不可用)、则它会向客户端发送"ingest failed"消息、或者、它会回退到创建临时对象副本并稍后评估ILM、具体取决于您在创建ILM规则时所做的选择。

- 双提交：StorageGRID 会立即为该对象创建两个中间副本、每个副本位于不同的存储节点上、并向客户端发送"ingest successful"消息。然后、StorageGRID 会将对象排队以进行ILM评估。

在StorageGRID 执行ILM评估时、它会首先检查临时副本是否满足ILM规则中的放置说明。例如、这两个中间副本可能满足双副本ILM规则中的说明、但它们不符合纠删编码规则中的说明。如果临时副本不符合ILM指令、StorageGRID 将创建新的对象副本并删除不需要的任何临时副本。

如果StorageGRID 无法创建两个中间副本(例如、如果网络问题描述 阻止创建第二个副本)、则StorageGRID 不会重试。载入失败。



S3或Swift客户端可以指定StorageGRID 在载入时通过指定创建一个临时副本 REDUCED_REDUNDANCY 存储类。有关详细信息、请参见实施S3或Swift客户端的说明。

默认情况下、StorageGRID 使用同步放置在载入期间保护对象。

相关信息

["用于载入的数据保护选项"](#)

["使用 S3"](#)

["使用 Swift"](#)

用于载入的数据保护选项

创建 ILM 规则时，您可以指定以下三个选项之一来在载入时保护对象：双提交，平衡或严格。根据您的选择，StorageGRID 会创建临时副本并将对象排队，以便稍后进行 ILM 评估，或者使用同步放置并立即创建副本以满足 ILM 要求。

双提交

选择双提交选项后，StorageGRID 会立即在两个不同的存储节点上创建临时对象副本，并向客户端返回"ingest successful" 消息。对象将排队等待 ILM 评估，满足规则放置说明的副本将在稍后创建。

何时使用双提交选项

在以下任一情况下，请使用双提交选项：

- 您正在使用多站点 ILM 规则，客户端载入延迟是您的主要考虑因素。使用双提交时，您必须确保网格可以在不满足 ILM 要求的情况下执行创建和删除双提交副本的额外工作。具体而言：
 - 网格上的负载必须足够低，以防止 ILM 积压。

- 网格必须具有多余的硬件资源（IOPS，CPU，内存，网络带宽等）。
- 您正在使用多站点 ILM 规则，并且站点之间的 WAN 连接通常具有较高的延迟或有限的带宽。在这种情况下，使用双提交选项有助于防止客户端超时。在选择双提交选项之前，您应使用实际工作负载测试客户端应用程序。

严格

如果选择 **strict** 选项，则 StorageGRID 会在载入时使用同步放置，并立即创建规则放置说明中指定的所有对象副本。如果 StorageGRID 无法创建所有副本，则载入将失败，例如，因为所需存储位置暂时不可用。客户端必须重试此操作。

何时使用 **strict** 选项

如果您在操作或法规要求中要求仅将对象立即存储在 ILM 规则中所述的位置，请使用 **strict** 选项。例如，要满足法规要求，您可能需要使用 **strict** 选项和 Location Constrict 高级筛选器来保证对象永远不会存储在特定数据中心。

"示例 5：用于严格载入行为的 ILM 规则和策略"

平衡

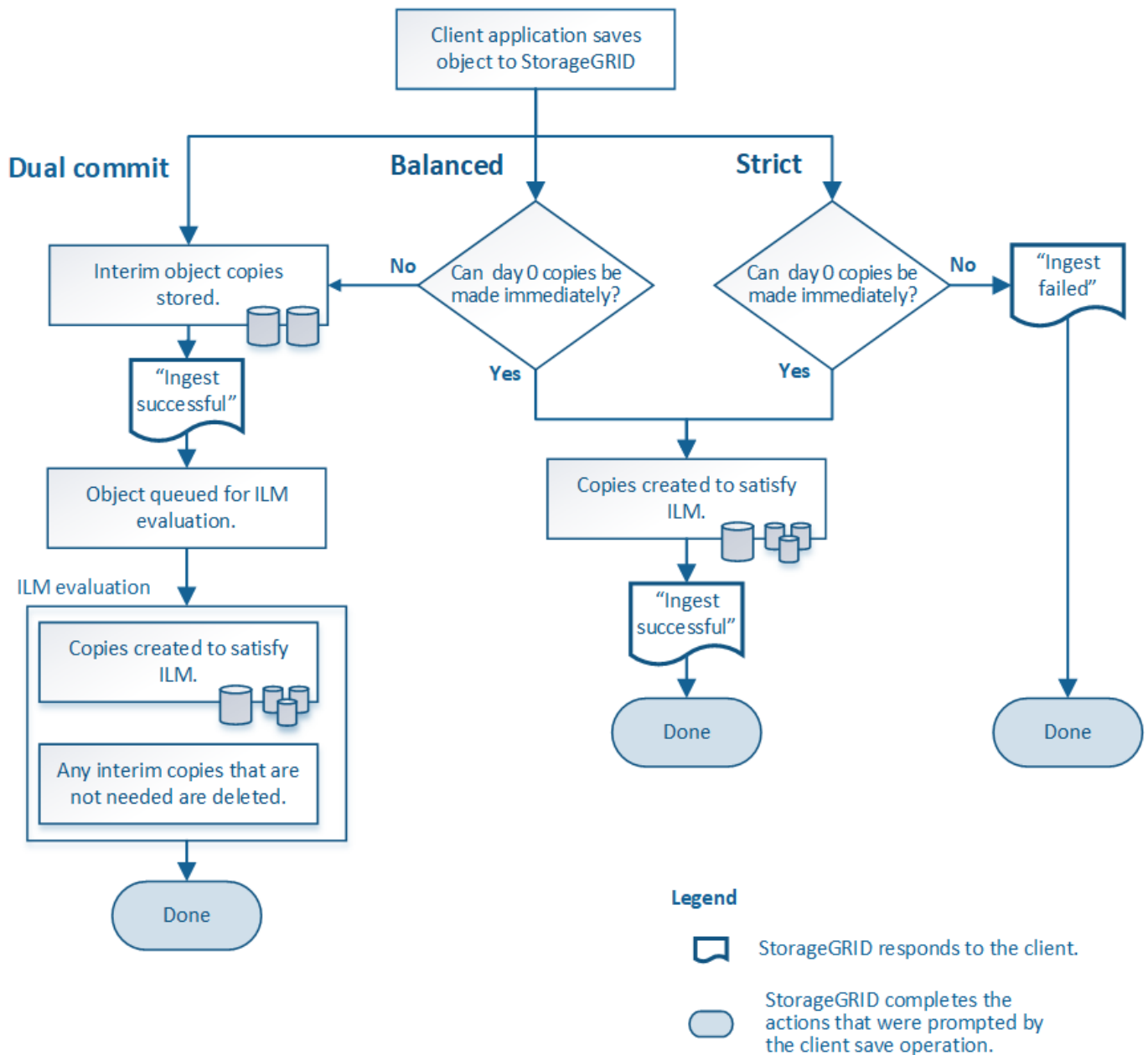
选择 **Balified** 选项时，StorageGRID 还会在载入时使用同步放置，并立即创建规则放置说明中指定的所有副本。与 **strict** 选项不同，如果 StorageGRID 无法立即创建所有副本，则会改用双提交。

何时使用平衡选项

使用平衡选项可实现数据保护，网格性能和载入成功的最佳组合。平衡是 ILM 规则向导中的默认选项。

三个载入选项的流程图

此流程图显示了使用这些载入选项之一的 ILM 规则对对象进行匹配时会发生什么情况。



相关信息

"如何载入对象"

数据保护选项的优势，劣势和限制

了解在载入时保护数据的三个选项（平衡，严格或双重提交）中每一个选项的优缺点有助于您确定要为 ILM 规则选择哪个选项。

平衡而严格的选项的优势

与在载入期间创建临时副本的双提交相比，这两个同步放置选项具有以下优势：

- **更好的数据安全性**：对象数据会按照 ILM 规则放置说明中的说明立即受到保护，可以对其进行配置，以防止出现多种故障情况，包括多个存储位置发生故障。双提交只能防止丢失一个本地副本。

- * 更高效的网格操作 *：每个对象仅在载入时进行一次处理。由于 StorageGRID 系统不需要跟踪或删除中间副本，因此处理负载较低，数据库空间占用较少。
- * (平衡) 建议 *：平衡选项可提供最佳 ILM 效率。除非需要严格的载入行为或网格满足用于双提交的所有标准，否则建议使用 `balanced` 选项。
- * (严格) 对象位置的确定性 *：`strict` 选项可确保根据 ILM 规则中的放置说明立即存储对象。

平衡而严格的选项的缺点

与双提交相比，平衡和严格选项存在一些缺点：

- * 客户端载入时间更长 *：客户端载入延迟可能更长。使用平衡和严格的选项时，在创建和存储所有纠删编码的片段或复制的副本之前，不会向客户端返回 `"ingest successful"` 消息。但是，对象数据很可能会更快地到达最终放置位置。
- * (严格) 更高的载入失败率 *：使用 `strict` 选项时，只要 StorageGRID 无法立即创建 ILM 规则中指定的所有副本，载入就会失败。如果所需存储位置暂时脱机或网络问题发生原因在站点之间复制对象时出现延迟，则可能会出现较高的载入失败率。
- 在某些情况下 * (严格) S3 多部分上传放置可能与预期不同 *：严格地说，您希望对象按照 ILM 规则的说明放置，或者载入操作失败。但是，如果使用 S3 多部分上传，则会在载入对象时对 ILM 的每个部分进行评估，并在多部分上传完成后对整个对象进行评估。在以下情况下，这可能会导致放置方式与您预期不同：
 - * 如果在 S3 多部分上传过程中 ILM 发生变化 *：由于每个部件都是根据在载入部件时处于活动状态的规则放置的，因此在多部分上传完成后，对象的某些部分可能不符合当前的 ILM 要求。在这些情况下，对象的载入不会失败。相反，未正确放置的任何部件将排队等待 ILM 重新评估，并稍后移动到正确的位置。
 - * 当 ILM 规则按大小筛选时 *：在评估某个部件的 ILM 时，StorageGRID 会按部件的大小进行筛选，而不是按对象的大小进行筛选。这意味着，对象的某些部分可以存储在在不满足整个对象的 ILM 要求的位置。例如，如果规则指定所有 10 GB 或更大的对象都存储在 DC1 中，而所有较小的对象存储在 DC2 中，则在载入时，10 部分多部分上传的每个 1 GB 部分都存储在 DC2 中。评估对象的 ILM 时，对象的所有部分都会移至 DC1。
- * (严格) 更新对象标记或元数据后，如果无法进行新要求的放置，则载入不会失败 *：严格地说，您希望按照 ILM 规则的说明放置对象，或者使载入失败。但是，在更新网格中已存储的对象的元数据或标记时，不会重新载入该对象。这意味着，更新触发的任何对象放置更改都不会立即进行。如果通过正常后台 ILM 流程重新评估 ILM，则会进行放置更改。如果无法进行所需的放置更改（例如，由于新需要的位置不可用），则更新后的对象将保留其当前放置位置，直到可以进行放置更改为止。

具有平衡或严格选项的对象放置限制

对于具有以下任一放置说明的 ILM 规则，不能使用平衡或严格选项：

- 第 0 天放置在云存储池中。
- 在第 0 天放置在归档节点中。
- 当规则将用户定义的创建时间作为参考时间时，将放置在云存储池或归档节点中。

之所以存在这些限制，是因为 StorageGRID 无法同步向云存储池或归档节点创建副本，并且用户定义的创建时间可以解决为当前问题。

ILM 规则和一致性控制如何交互以影响数据保护

ILM 规则和一致性控制选项都会影响对象的保护方式。这些设置可以进行交互。

例如，为 ILM 规则选择的载入行为会影响对象副本的初始放置，而存储对象时使用的一致性控制则会影响对象元数据的初始放置。由于 StorageGRID 需要访问对象的元数据及其数据来满足客户端请求，因此为一致性级别和载入行为选择匹配的保护级别可以提供更好的初始数据保护和更可预测的系统响应。

下面简要总结了 StorageGRID 中提供的一致性控制：

- * 全部 *：所有节点均立即接收对象元数据，否则请求将失败。
- * 强 - 全局 *：对象元数据会立即分发到所有站点。保证所有站点中所有客户端请求的写入后读一致性。
- * 强站点 *：对象元数据会立即分发到站点上的其他节点。保证站点内所有客户端请求的写入后读一致性。
- * 读后新写入 *：为新对象提供读后写入一致性，并最终为对象更新提供一致性。提供高可用性和数据保护保证。
- * 可用 *（机头操作的最终一致性）：与 read-after-new-write 一致性级别相同，但仅为机头操作提供最终一致性。



在选择一致性级别之前，请阅读有关创建 S3 或 Swift 客户端应用程序的说明中的这些设置的完整问题描述。在更改默认值之前，您应了解其优势和限制。

一致性控制和 ILM 规则如何交互的示例

假设您有一个双站点网络，其中包含以下 ILM 规则和以下一致性级别设置：

- * ILM 规则 *：创建两个对象副本，一个在本地站点，一个在远程站点。此时将选择严格的载入行为。
- * 一致性级别 *：strong-global（对象元数据会立即分发到所有站点。）

当客户端将对象存储到网格时，StorageGRID 会创建两个对象副本并将元数据分发到两个站点，然后再向客户端返回成功。

在载入成功消息时，此对象将受到完全保护，不会丢失。例如，如果本地站点在载入后不久丢失，则远程站点上仍存在对象数据和对象元数据的副本。此对象完全可检索。

如果您改用相同的 ILM 规则和 strong-site 一致性级别，则在将对象数据复制到远程站点之后，在将对象元数据分发到该远程站点之前，客户端可能会收到一条成功消息。在这种情况下，对象元数据的保护级别与对象数据的保护级别不匹配。如果本地站点在载入后不久丢失，则对象元数据将丢失。无法检索此对象。

一致性级别和 ILM 规则之间的关系可能很复杂。如需帮助，请联系 NetApp。

相关信息

["什么是复制"](#)

["什么是纠删编码"](#)

["什么是纠删编码方案"](#)

["示例 5：用于严格载入行为的 ILM 规则和策略"](#)

["使用 S3"](#)

["使用 Swift"](#)

对象的存储方式（复制或纠删编码）

StorageGRID 可以通过存储复制的副本或存储经过纠删编码的副本来防止对象丢失。您可以在 ILM 规则的放置说明中指定要创建的副本类型。

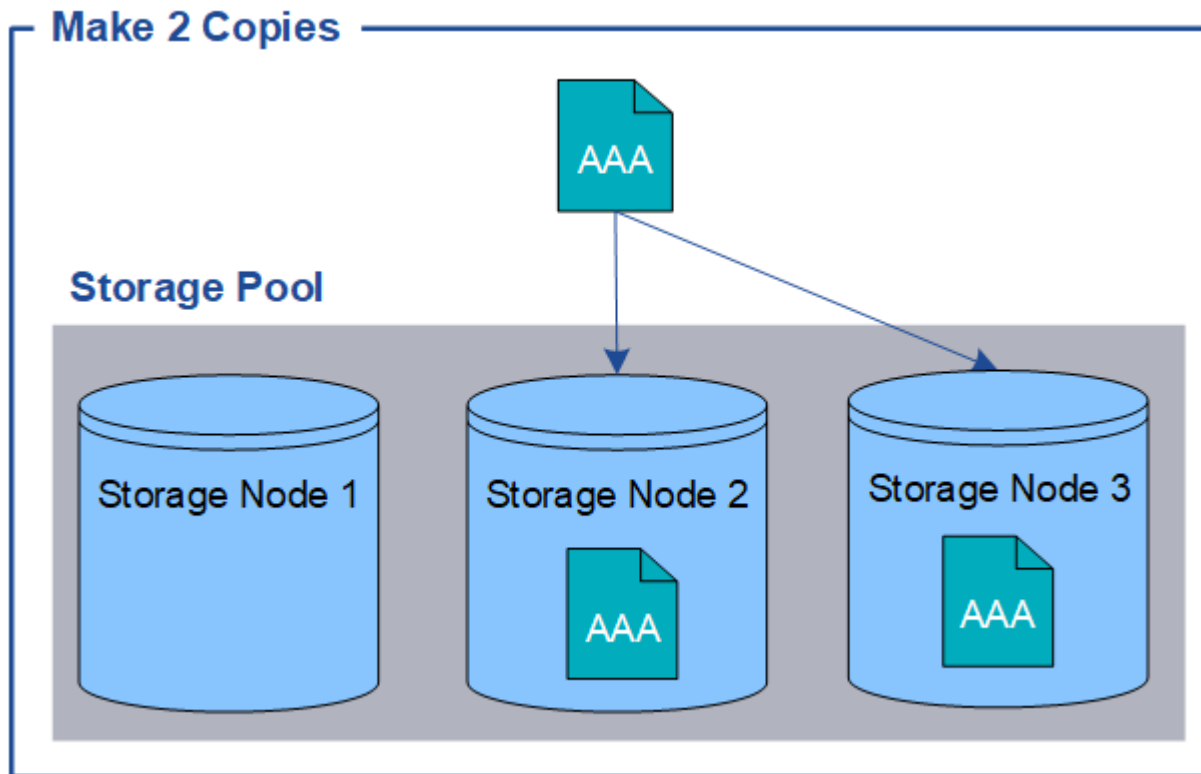
- "什么是复制"
- "为什么不应使用单副本复制"
- "什么是纠删编码"
- "什么是纠删编码方案"
- "纠删编码的优势，劣势和要求"

什么是复制

复制是 StorageGRID 用于存储对象数据的两种方法之一。当对象与使用复制的 ILM 规则匹配时，系统会为对象数据创建精确副本，并将这些副本存储在存储节点或归档节点上。

在配置创建复制副本的 ILM 规则时，您可以指定应创建多少个副本，这些副本应放置在何处以及应将这些副本存储在每个位置的时间长度。

在以下示例中，ILM 规则指定将每个对象的两个复制副本放置在包含三个存储节点的存储池中。



当 StorageGRID 将对象与此规则匹配时，它会为该对象创建两个副本，并将每个副本放置在存储池中的不同存储节点上。这两个副本可以放置在三个可用存储节点中的任意两个上。在这种情况下，规则会将对象副本放置在存储节点 2 和 3 上。由于有两个副本，因此，如果存储池中的任何节点出现故障，可以检索此对象。



StorageGRID 只能在任何给定存储节点上存储一个对象的一个复制副本。如果您的网格包含三个存储节点，并且您创建了一个 4 副本 ILM 规则，则只会创建三个副本—每个存储节点一个副本。系统将触发 * 无法实现 ILM 放置 * 警报，以指示无法完全应用 ILM 规则。

相关信息

["什么是存储池"](#)

["使用多个存储池进行跨站点复制"](#)

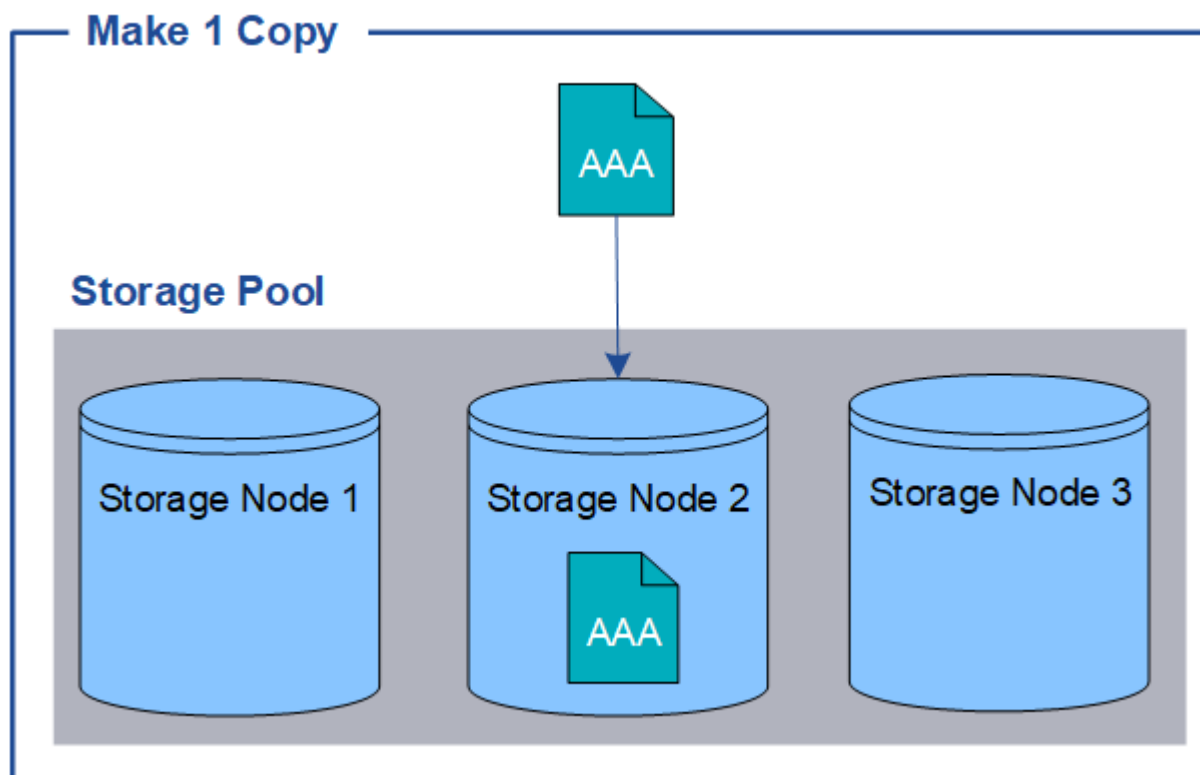
为什么不应使用单副本复制

在创建 ILM 规则以创建复制副本时，您应始终在放置说明中指定至少两个副本，以便在任意时间段内使用。



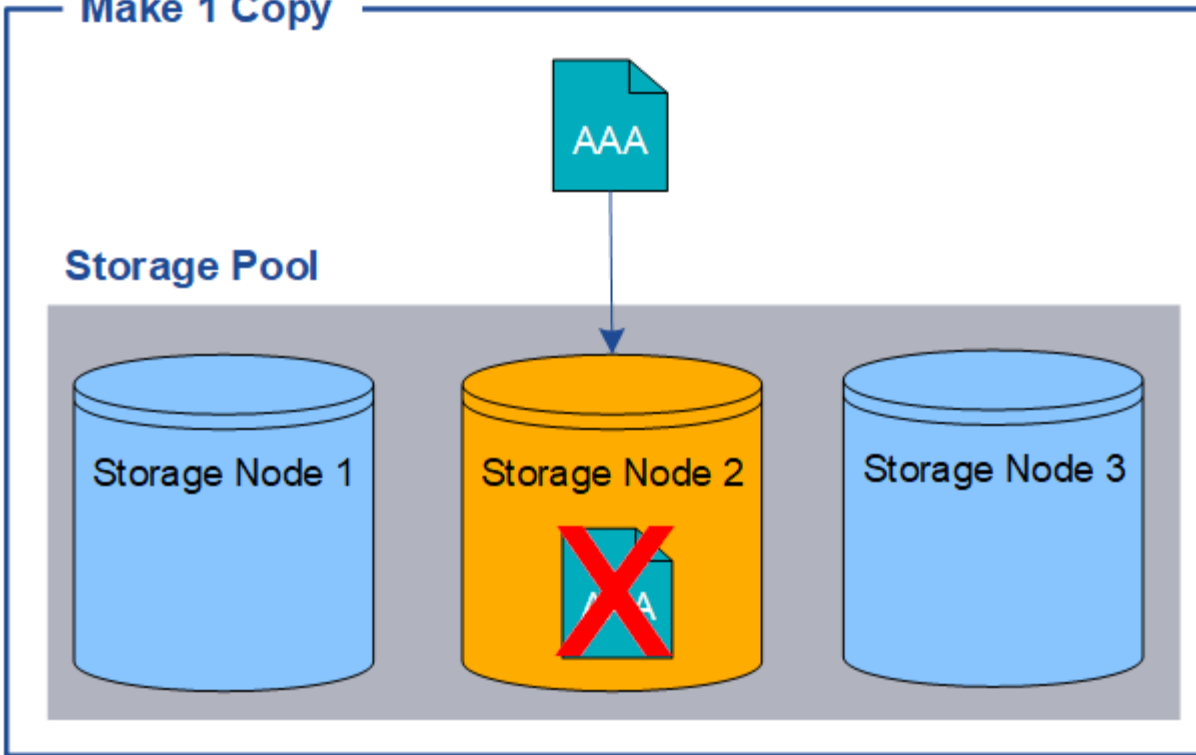
请勿使用仅在任意时间段创建一个复制副本的 ILM 规则。如果某个对象只存在一个复制副本，则在存储节点出现故障或出现严重错误时，该对象将丢失。在升级等维护过程中，您还会暂时失去对对象的访问权限。

在以下示例中，Make 1 Copy ILM 规则指定将对象的一个复制副本放置在包含三个存储节点的存储池中。如果载入的对象与此规则匹配，则 StorageGRID 仅会在一个存储节点上放置一个副本。



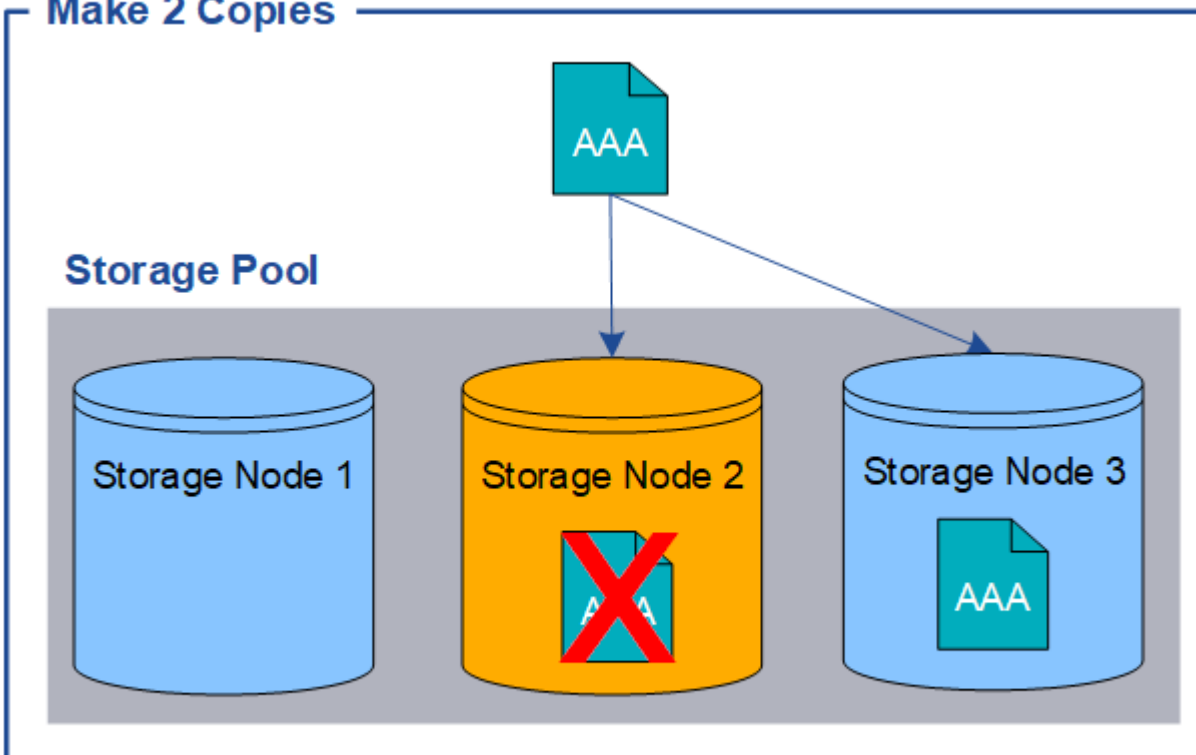
如果 ILM 规则仅创建一个对象的一个复制副本，则在存储节点不可用时，此对象将无法访问。在此示例中，只要存储节点 2 脱机，例如在升级或其他维护操作步骤期间，您将暂时无法访问对象 AAA。如果存储节点 2 发生故障，您将完全丢失对象 AAA。

Make 1 Copy



为了避免丢失对象数据，您应始终为要通过复制保护的所有对象创建至少两个副本。如果存在两个或更多副本，则在一个存储节点出现故障或脱机时，您仍可以访问此对象。

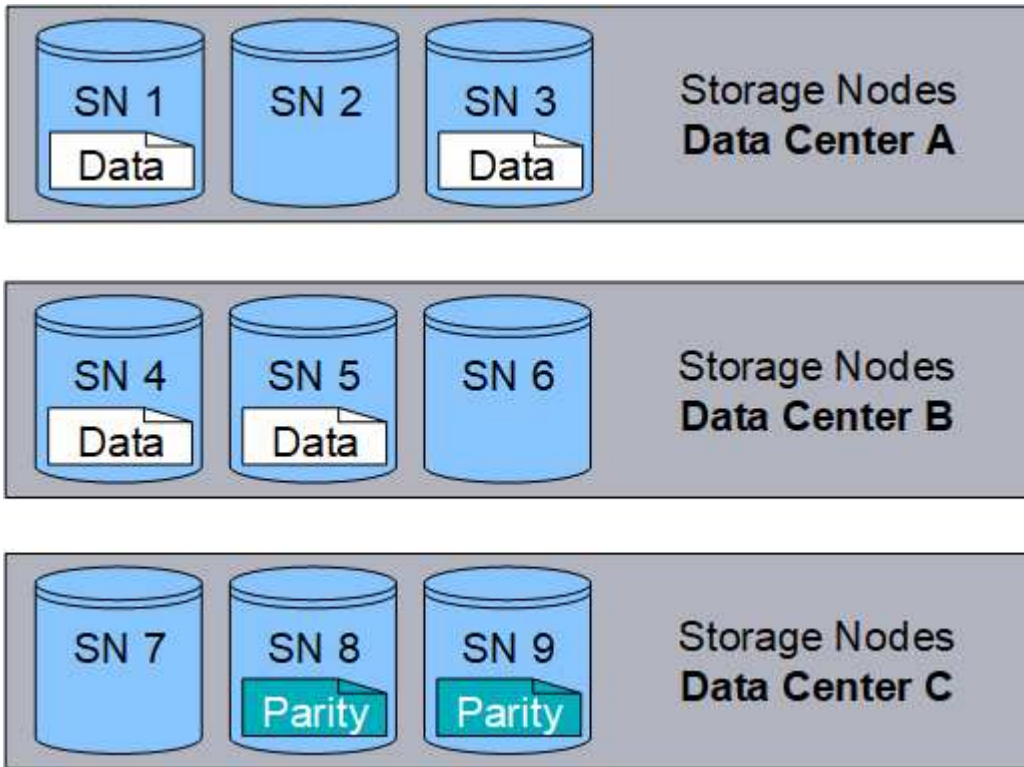
Make 2 Copies



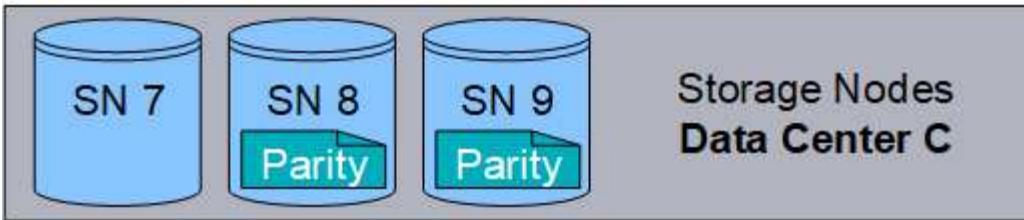
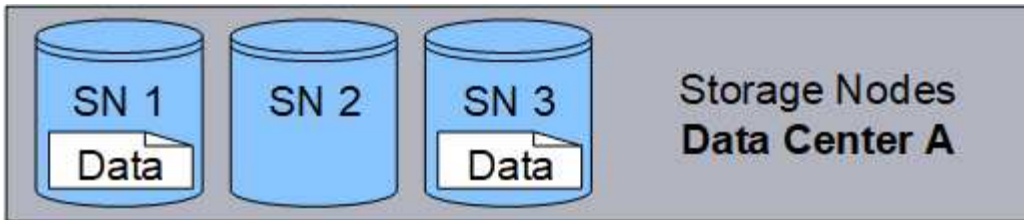
什么是纠删编码

纠删编码是 StorageGRID 存储对象数据的第二种方法。如果 StorageGRID 将对象与配置为创建纠删编码副本的 ILM 规则匹配，则会将对象数据分段为数据片段，计算额外的奇偶校验片段，并将每个片段存储在不同的存储节点上。访问某个对象时，系统会使用存储的片段重新组合该对象。如果数据或奇偶校验片段损坏或丢失，则纠删编码算法可以使用剩余数据和奇偶校验片段的子集重新创建该片段。

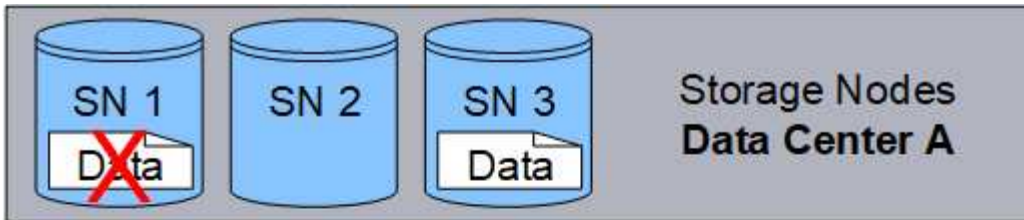
以下示例说明了如何对对象数据使用纠删编码算法。在此示例中，ILM 规则使用 4+2 纠删编码方案。每个对象都会被划分为四个相等的数据片段，并根据对象数据计算两个奇偶校验片段。六个片段中的每个片段都存储在三个数据中心站点的不同节点上，以便为节点故障或站点丢失提供数据保护。



4+2 纠删编码方案至少需要九个存储节点，三个不同站点中的每个站点各有三个存储节点。只要六个片段中的任意四个（数据或奇偶校验）仍然可用，就可以检索对象。最多可以丢失两个片段，而不会丢失对象数据。如果整个数据中心站点丢失，只要所有其他片段仍可访问，仍可检索或修复对象。



如果丢失两个以上的存储节点，则无法检索此对象。



相关信息

["什么是存储池"](#)

["什么是纠删编码方案"](#)

什么是纠删编码方案

为 ILM 规则配置纠删编码配置文件时，您可以根据计划使用的存储池中的存储节点和站点数量选择可用的纠删编码方案。纠删编码方案可控制为每个对象创建的数据片段数量和奇偶校验片段数量。

StorageGRID 系统使用 Reed-Solomon 纠删编码算法。该算法会将对象分段为 k 数据片段，并计算 m 奇偶校验片段。 $k + m = n$ 个片段分布在 n 个存储节点上，以提供数据保护。一个对象最多可承受丢失或损坏的碎片。检索或修复对象需要 K 个片段。

在配置擦除编码配置文件时，请对存储池遵循以下准则：

- 存储池必须包含三个或更多站点，或者只包含一个站点。



如果存储池包含两个站点，则无法配置擦除编码配置文件。

- [包含三个或更多站点的存储池的纠删编码方案](#)
- [单站点存储池的纠删编码方案](#)
- 请勿使用默认存储池，所有存储节点或包含默认站点的存储池所有站点。
- 存储池应至少包含 $k+m+1$ 个存储节点。

所需的最小存储节点数为 $k+m$ 。但是，如果所需的存储节点暂时不可用，则至少添加一个存储节点有助于防止载入失败或 ILM 回退。

擦除编码方案的存储开销是通过将奇偶校验片段数 (m) 除以数据片段数 (k) 计算得出的。您可以使用存储开销计算每个擦除编码对象所需的磁盘空间量：

$$\text{disk space} = \text{object size} + (\text{object size} * \text{storage overhead})$$

例如，如果使用 4+2 方案存储一个 10 MB 的对象（存储开销为 50%），则该对象将占用 15 MB 的网格存储。如果使用 6+2 方案存储同一个 10 MB 对象（存储开销为 33%），则该对象将占用大约 13.3 MB 的空间。

选择总值最低的纠删编码方案 $k+m$ ，以满足您的需求。碎片数量较少的纠删编码方案在计算方面总体上效率更高，因为每个对象创建和分布（或检索）的碎片数量较少，由于碎片大小较大，性能可能会更高，并且在需要更多存储时，扩展中添加的节点可能会更少。（有关规划存储扩展的信息，请参见有关扩展 StorageGRID 的说明。）

包含三个或更多站点的存储池的纠删编码方案

下表介绍了 StorageGRID 当前支持的纠删编码方案，该方案适用于包含三个或更多站点的存储池。所有这些方案均可提供站点丢失保护。一个站点可能会丢失，但对象仍可访问。

对于提供站点丢失保护的纠删编码方案，存储池中的建议存储节点数超过 $k+m+1$ ，因为每个站点至少需要三个存储节点。

纠删编码方案 ($k+m$)	已部署站点的最小数量	每个站点的建议存储节点数	建议的存储节点总数	站点丢失保护?	存储开销
4+2	3.	3.	9	是的。	50%
6+2	4.	3.	12	是的。	33%
8+2	5.	3.	15	是的。	25%
6+3.	3.	4.	12	是的。	50%
9+3.	4.	4.	16.	是的。	33%
2+1	3.	3.	9	是的。	50%
4+1	5.	3.	15	是的。	25%
6+1	7.	3.	21	是的。	17%
7+5.	3.	5.	15	是的。	71%



StorageGRID 要求每个站点至少有三个存储节点。要使用 7+5 方案，每个站点至少需要四个存储节点。建议每个站点使用五个存储节点。

在选择提供站点保护的纠删编码方案时，请平衡以下因素的相对重要性：

- * 碎片数量 *：当碎片总数减少时，性能和扩展灵活性通常会提高。
- * 容错 *：容错可通过包含更多奇偶校验分段来提高（即，当 m 的值较高时）。
- * 网络流量 *：从故障中恢复时，使用包含更多片段的方案（即 $k+m$ 的总数更高）会创建更多网络流量。
- * 存储开销 *：开销较高的方案需要每个对象更多的存储空间。

例如，在选择 4+2 方案和 6+3 方案（两者都有 50% 的存储开销）时，如果需要额外的容错功能，请选择 6+3 方案。如果网络资源受限，请选择 4+2 方案。如果所有其他因素相等，请选择 4+2，因为其碎片总数较低。



如果您不确定要使用的方案，请选择 4+2 或 6+3，或者联系技术支持。

单站点存储池的纠删编码方案

单站点存储池支持为三个或更多站点定义的所有纠删编码方案，但前提是该站点具有足够的存储节点。

所需的最小存储节点数为 $k+m$ ，但建议使用具有 $k+m+1$ 存储节点的存储池。例如，2+1 纠删编码方案要求一个存储池至少包含三个存储节点，但建议使用四个存储节点。

纠删编码方案 (k+m)	存储节点的最小数量	建议的存储节点数	存储开销
4+2	6.	7.	50%
6+2	8.	9	33%
8+2	10	11.	25%
6+3.	9	10	50%
9+3.	12	13	33%
2+1	3.	4.	50%
4+1	5.	6.	25%
6+1	7.	8.	17%
7+5.	12	13	71%

相关信息

["扩展网格"](#)

纠删编码的优势，劣势和要求

在决定是使用复制还是纠删编码来保护对象数据不会丢失之前，您应了解纠删编码的优点，缺点和要求。

纠删编码的优势

与复制相比，纠删编码可提高可靠性，可用性和存储效率。

- *** 可靠性 ***：可靠性通过容错来衡量—即，在不丢失数据的情况下可以同时发生的故障数量。通过复制，多个相同的副本会存储在不同的节点上以及不同的站点上。通过纠删编码，对象会编码为数据和奇偶校验片段，并分布在多个节点和站点上。这种分散方式可同时提供站点和节点故障保护。与复制相比，纠删编码可提高可靠性，而存储成本相当。
- *** 可用性 ***：可用性可定义为在存储节点出现故障或无法访问时检索对象的功能。与复制相比，纠删编码可以以相当的存储成本提高可用性。
- *** 存储效率 ***：对于相似级别的可用性和可靠性，通过纠删编码保护的對象比通过复制保护的相同对象占用的磁盘空间更少。例如，复制到两个站点的 10 MB 对象会占用 20 MB 的磁盘空间（两个副本），而采用 6+3 纠删编码方案在三个站点之间进行纠删编码的对象只会占用 15 MB 的磁盘空间。



擦除编码对象的磁盘空间计算为对象大小加上存储开销。存储开销百分比是奇偶校验片段数除以数据片段数。

纠删编码的缺点

与复制相比，纠删编码具有以下缺点：

- 需要增加存储节点和站点的数量。例如，如果使用纠删编码方案 6+3，则必须在三个不同站点至少要有三个存储节点。相比之下，如果只复制对象数据，则每个副本只需要一个存储节点。
- 存储扩展的成本和复杂性增加。要扩展使用复制的部署，只需在创建对象副本的每个位置添加存储容量即可。要扩展使用纠删编码的部署，您必须同时考虑使用的纠删编码方案以及现有存储节点的容量。例如，如果您等待现有节点达到 100% 全满，则必须至少添加 $k+m$ 存储节点，但如果在现有节点达到 70% 全满时进行扩展，则可以为每个站点添加两个节点，同时仍可最大程度地提高可用存储容量。有关详细信息、请参见有关扩展 StorageGRID 的说明。
- 在分布在不同地理位置的站点之间使用纠删编码时，检索延迟会增加。与在本地复制并提供的对象（客户端连接的同一站点）相比，通过 WAN 连接检索经过纠删编码并分布在远程站点上的对象的对象片段所需时间更长。
- 在地理位置分散的站点之间使用纠删编码时，检索和修复的 WAN 网络流量使用率较高，尤其是频繁检索的对象或通过 WAN 网络连接进行对象修复。
- 当您在站点间使用纠删编码时，最大对象吞吐量会随着站点间网络延迟的增加而急剧下降。这一减少是由于 TCP 网络吞吐量相应减少，从而影响 StorageGRID 系统存储和检索对象片段的速度。
- 提高计算资源的利用率。

何时使用纠删编码

纠删编码最适合以下要求：

- 大于 1 MB 的对象。



由于管理与纠删编码副本关联的片段数量会产生开销、因此请勿对 200 KB 或更小的对象使用纠删编码。

- 长期或冷存储，用于存储不经常检索的内容。
- 高数据可用性和可靠性。
- 防止发生完整的站点和节点故障。
- 存储效率。
- 需要高效数据保护的单站点部署，只需一个纠删编码副本，而不是多个复制副本。
- 站点间延迟小于 100 毫秒的多站点部署。

相关信息

["扩展网格"](#)

如何确定对象保留

StorageGRID 为网格管理员和单个租户用户提供了指定对象存储时间的选项。通常，租户用户提供的任何保留指令优先于网格管理员提供的保留指令。

租户用户如何控制对象保留

租户用户可以通过三种主要方式控制其对象在 StorageGRID 中的存储时间：

- 如果为网格启用了全局 S3 对象锁定设置，则 S3 租户用户可以在启用了 S3 对象锁定的情况下创建分段，然后使用 S3 REST API 为添加到该分段的每个对象版本指定保留日期和合法保留设置。
 - 任何方法都不能删除处于合法保留状态的对象版本。
 - 在达到对象版本的保留截止日期之前，无法通过任何方法删除该版本。
 - 启用了 S3 对象锁定的存储分段中的对象由 ILM "Forever 保留。`但是，在达到保留截止日期后，可以通过客户端请求或存储分段生命周期到期来删除对象版本。

"使用S3对象锁定管理对象"

- S3 租户用户可以将生命周期配置添加到其分段中，以指定到期操作。如果存在分段生命周期，则 StorageGRID 会存储一个对象，直到满足到期操作中指定的日期或天数为止，除非客户端先删除该对象。
- S3 或 Swift 客户端可以对删除对象请求进行问题描述 处理。在确定是删除还是保留对象时，StorageGRID 始终会优先处理客户端删除请求，而不是 S3 存储分段生命周期或 ILM 。

网格管理员如何控制对象保留

网格管理员可以使用 ILM 放置说明来控制对象的存储时间。如果对象与 ILM 规则匹配，则 StorageGRID 会存储这些对象，直到 ILM 规则中的最后一个时间段结束为止。如果为放置说明指定了 "forever"，则对象将无限期保留。

无论谁控制对象的保留时间，ILM 设置都可控制存储的对象副本类型（复制或擦除编码）以及副本所在位置（存储节点，云存储池或归档节点）。

S3 存储分段生命周期和 ILM 如何交互

S3 存储分段生命周期中的到期操作始终会覆盖 ILM 设置。因此，即使有关放置对象的任何 ILM 指令已失效，该对象也可能会保留在网格中。

对象保留示例

要更好地了解 S3 对象锁定，存储分段生命周期设置，客户端删除请求和 ILM 之间的交互，请考虑以下示例。

示例 1：S3 存储分段生命周期将对象保留的时间超过 ILM

ILM

将两个副本存储 1 年（365 天）

分段生命周期

对象在 2 年（730 天）后过期

结果

StorageGRID 会将对象存储 730 天。StorageGRID 使用存储分段生命周期设置来确定是删除还是保留对象。



如果存储分段生命周期指定对象的保留时间应超过 ILM 指定的时间，则 StorageGRID 在确定要存储的副本数量和类型时会继续使用 ILM 放置说明。在此示例中，从第 366 天到第 730 天，此对象的两个副本将继续存储在 StorageGRID 中。

示例 2：S3 存储分段生命周期将对象在 ILM 之前过期

ILM

将两个副本存储 2 年（730 天）

分段生命周期

对象在 1 年（365 天）后过期

结果

StorageGRID 将在 365 天后删除此对象的两个副本。

示例 3：客户端删除将覆盖存储分段生命周期和 ILM

ILM

在存储节点 "Forever" 上存储两个副本

分段生命周期

对象在 2 年（730 天）后过期

客户端删除请求

发布日期：第 400 天

结果

StorageGRID 会在第 400 天删除此对象的两个副本，以响应客户端删除请求。

示例 4：S3 对象锁定会覆盖客户端删除请求

S3 对象锁定

对象版本的保留截止日期为 2026-03-31。合法保留无效。

符合 ILM 规则

在存储节点 "Forever" 上存储两个副本。`

客户端删除请求

发布日期：2024-03-31。

结果

StorageGRID 不会删除此对象版本，因为保留截止日期仍在 2 年后。

相关信息

["使用S3对象锁定管理对象"](#)

["使用 S3"](#)

如何删除对象

StorageGRID 可以直接响应客户端请求删除对象，也可以因 S3 存储分段生命周期到期或 ILM 策略要求而自动删除对象。了解可删除对象的不同方式以及 StorageGRID 如何处理删除请求有助于您更有效地管理对象。

StorageGRID 可以使用以下两种方法之一删除对象：

- 同步删除：当 StorageGRID 收到客户端删除请求时，将立即删除所有对象副本。删除副本后，系统会通知客户端删除操作成功。
- 对象将排队等待删除：当 StorageGRID 收到删除请求时，该对象将排队等待删除，并且系统会立即通知客户端删除已成功。对象副本稍后将通过后台 ILM 处理进行删除。

删除对象时，StorageGRID 会使用方法来优化删除性能，最大限度地减少潜在的删除积压并以最快的速度释放空间。

下表总结了 StorageGRID 何时使用每种方法。

执行删除的方法	使用时
对象已排队等待删除	当满足以下条件中的 * 任意 * 时： <ul style="list-style-type: none">• 以下事件之一已触发自动对象删除：<ul style="list-style-type: none">◦ 已达到 S3 存储分段的生命周期配置中的到期日期或天数。◦ ILM 规则中指定的最后一个时间段已过。• 注意：* 如果已启用 S3 对象锁定的存储分段中的对象处于合法保留状态，或者已指定保留日期但尚未满足，则无法删除这些对象。• S3 或 Swift 客户端请求删除，但其中一个或多个条件为 true：<ul style="list-style-type: none">◦ 无法在 30 秒内删除副本，因为对象位置暂时不可用。◦ 后台删除队列处于空闲状态。
立即删除对象（同步删除）	当 S3 或 Swift 客户端发出删除请求且满足以下 * 所有 * 条件时： <ul style="list-style-type: none">• 可以在 30 秒内删除所有副本。• 后台删除队列包含要处理的对象。

当 S3 或 Swift 客户端发出删除请求时，StorageGRID 首先会向删除队列添加多个对象。然后，它会切换到执行同步删除。确保后台删除队列包含要处理的对象，这样 StorageGRID 可以更高效地处理删除，尤其是对于低并发性客户端，同时有助于防止客户端删除积压。

了解 StorageGRID 删除对象的方式的影响

StorageGRID 删除对象的方式可能会影响系统的执行方式：

- 当 StorageGRID 执行同步删除时，StorageGRID 可能需要长达 30 秒才能将结果返回给客户端。这意味着删除的速度可能会更慢，即使副本的实际删除速度比 StorageGRID 将对象排队等待删除时要快。
- 如果您在批量删除期间密切监控删除性能，则可能会注意到删除了一定数量的对象后，删除率似乎会变慢。当 StorageGRID 从对要删除的对象进行排队转移到执行同步删除时，会发生此更改。删除率明显降低并不意味着删除对象副本的速度较慢。相反，它表明平均而言，空间释放速度更快。

如果要删除大量对象，并且您的优先级是快速释放空间，请考虑使用客户端请求删除对象，而不是使用 ILM 或其他方法删除这些对象。通常，当客户端执行删除时，空间释放速度会更快，因为 StorageGRID 可以使用同步删除。

请注意，删除对象后释放空间所需的时间取决于多种因素：

- 对象副本是同步删除还是稍后排队等待删除（对于客户端删除请求）。
- 其他因素包括网格中的对象数或对象副本排队等待删除时网格资源的可用性（对于客户端删除和其他方法）。

如何删除受版本控制的 S3 对象

如果为 S3 存储分段启用了版本控制，则无论这些请求来自 S3 客户端，S3 存储分段生命周期到期还是 ILM 策略要求，StorageGRID 都会在响应删除请求时遵循 Amazon S3 的行为。

对对象进行版本控制时，对象删除请求不会删除对象的当前版本，也不会释放空间。相反，对象删除请求只会创建一个删除标记作为对象的当前版本，从而使先前版本的对象 "noncurrent." 成为非最新版本。

即使尚未删除此对象，StorageGRID 的行为仍会使当前版本的对象不再可用。对该对象的请求将返回 404 NotFound。但是，由于未删除非当前对象数据，因此指定非当前对象版本的请求可能会成功。

要在删除受版本控制的对象时释放空间，必须执行以下操作之一：

- * S3客户端请求*：在S3删除对象请求中指定对象版本号 (DELETE /object?versionId=ID)。请注意，此请求仅删除指定版本的对象副本（其他版本仍占用空间）。
- 分段生命周期：使用 NoncurrentVersionExpiration 存储分段生命周期配置中的操作。满足指定的非当前磁盘数后，StorageGRID 将永久删除非当前对象版本的所有副本。无法恢复这些对象版本。
- * ILM：向 ILM 策略添加两个 ILM 规则。在第一个规则中使用 * 非当前时间 * 作为参考时间，以匹配对象的非当前版本。在第二个规则中使用 * 载入时间 * 以匹配当前版本。必须在 * 载入时间 * 规则上方的策略中显示 * 非当前时间 * 规则。

相关信息

["使用 S3"](#)

["示例 4：S3 版本对象的 ILM 规则和策略"](#)

版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。