### 软件衰退的特点

与其他故障现象不同，软件衰退是成熟软件在运行时受到环境的变化和隐蔽的、间歇性的、暂时性的缺陷等因素威胁，长期累积的过程，它因这些因素的难以避免而必然存在，其导致的稳定性、性能下降或失效迟早会发生。上述这些威胁具有环境敏感性，能够激活它们的相同情景是随机的、很难重现，因此极难通过测试来再现和排除它们。由于软件自身的动态性及其对环境的依赖性和未知缺陷层出不穷，偶发性、不确定性和环境依赖性成为了软件衰退的三大特征：

1. **偶发性**：引发衰退的内外部威胁（fault等）具有隐蔽性、间歇性，这类bug的激活依赖于软件自身的内部环境和外部环境（运行环境、输入环境），因此衰退时隐时现。
2. **不确定性**[58]**：**软件运行在开放、动态和多变的环境中，其功能和结构日渐复杂化、多样化且在不断更新、升级、修改配置，未知的会引发衰退的缺陷也层出不穷，衰退的表现模式、趋势也随之动态多变且多样；从软件开始运行到触发导致衰退的因素、再到发生性能下降或失效的时间和间隔是随机的，因为触发衰退的条件及其强度、软件的使用强度、运行刨面、软件相应的进程执行的工作量和工作类型等因素具有随机性。
3. **环境依赖性（环境多样性）**：在不同的环境中，软件衰退的表现不同；换一个运行环境，软件衰退的影响可能就不一样。同样的，集群环境中一个节点遇到的衰退情景、其他节点也遇到的概率相当大。

### 未来软件抗衰的关键问题与发展趋势

从上一节可以看出，软件抗衰的第一层含义一定程度上已达到，第二层含义尚未实现，因为现有的抗衰方法一般基于特定的软件衰退问题（如内存泄露），而软件衰退的偶发性和不确定性使这些方法常常有漏洞。沿着软件抗衰方法发展的轨迹，本文认为未来的软件抗衰方法，应该解决如下几个关键问题：

1. **自适应评估衰退的程度、粒度**

分析运行时软件的衰退的表现行为对系统的影响，自适应找到适合刻画不同监测对象健康状态的性能参数，量化衰退（不健康）的程度和粒度，这是动态确定预测关键点、制定和实施抗衰策略的重要依据。目前的抗衰方法主要通过检测系统资源的耗费情况、响应时间和响应率等来验证软件衰退的发生，但不能进一步量化。

1. **动态确定衰退预测关键点**

考虑动态多变的环境的影响，需要动态检测软件性能异常，区分正常的负载改变、软件升级等行为引起的性能变化和真正的性能衰退，将软件性能异常的征兆作为触发性能预测的条件，解决基于时间模型和度量的混合抗衰方法中，预测关键点固定、不适合动态多样的软件衰退的问题。

1. **自适应检测未知软件衰退**

自适应检测未知的软件衰退模式，这些模式包括偶发性强的、衰退因素未知的、模式动态、多变的，它们的共同特点是衰退样本缺乏或难以获取。

1. **自适应预测未知软件衰退**

现有抗衰方法依赖于衰退样本，一般基于特定的软件衰退情况[42]（如内存泄露）及其衰退样本集进行预测，选取的模型过度拟合已有衰退样本。这些方法在预测偶发性强、动态、多样的软件衰退时，易发生因缺乏衰退样本而被逃避的问题，因此需要构建不依赖衰退样本即可对未知衰退模式的趋势进行预测的方法。

1. **群体抗衰策略**

当前的应用软件越来越多地以集群方式工作，因此，在网络应用软件系统中，每个节点的软件抗衰系统就不能各自为阵、不顾其他节点上软件的健康状态，而应该通力合作，以把握系统中软件衰退的整体趋势，并据此相互协助、取长补短，以提高整个软件系统抗衰的效率和软件系统的整体可用性和性能。

为解决上述问题，本文认为软件抗衰方法在未来的发展中，必须要具备以下三个特性：

1. **不依赖衰退样本**：日益复杂的应用软件的衰退具有较强的偶发性、难以人为再现，衰退样本难以获取，因此，未来的软件抗衰方法不能依赖于衰退样本就能应对未知的软件衰退模式。
2. **自适应**：这是最需要解决的问题。针对软件衰退的不确定性（表现形式多样、动态多变、随机），未来的软件抗衰方法应能在不同的环境下，通过学习机制自动地调节和改变自身的组件和配置，实现对未知软件和未知衰退的检测、预测。
3. **环境多样性**：越来越多的应用软件（尤其是网络应用软件）以集群方式工作，集群中每个节点的应用软件的环境几乎一样，因为软件衰退具有环境依赖性(敏感性）和必然性，集群中一个节点的软件遭遇的衰退，其他节点也可能在类似的环境下、遭遇类似衰退。因此，在（网络应用）集群系统中，每个节点的软件抗衰系统就不能各自为阵，而应该相互协助，以提高集群环境中软件抗衰的效率和应用系统的整体可靠性。

上述方法应具备的特性，也是本文的方法力图达到的目标。