

Perl在用于系统管理的cfengine中应用 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/167/2021_2022_Perl_E5_9C_A8_E7_94_A8_c103_167234.htm cfengine（配置引擎）是一种 UNIX 管理工具，其目的是使简单的管理的任务自动化，使困难的任务变得较容易。它的目标是使系统从任何状态收敛到一种理想状态。依照它的作者 Mark Burgess 所说，cfengine 总是使您的系统更接近于您所定义的配置；它决不会使系统变得更糟。在本文中，Ted 通过演示一些简单的 cfengine 使用，使您熟悉这一独特的系统管理工具。初学者和中级系统管理员将从本文获益最多；解说和示例假设您可以运用大部分初学者级别的系统管理概念。要使用本文中的示例，您的系统应该是一个最近的（2000 或更新）主流 UNIX（Linux、Solaris、BSD）安装。示例可以与早期版本的 Perl 和 UNIX 以及其它操作系统一起使用，如果它们不能工作，对您来说，可以把它当成练习来解决。cfengine 正处于开发中。版本 1.6.3 是稳定的，它是本文的基础。当前正在进行阿尔法测试的版本 2.0 的计划包括重写体系结构、添加许多很棒的新功能部件以及使语法基本上保持不变。要了解 cfengine 的所有功能以及版本 2.0 中的增强功能，请访问 cfengine 主页（请参阅本文后面的参考资料）。cfengine 的要点 cfengine 将改变您的系统管理方法。您将运行一个命令并观察系统将收敛到一种稳定状态。我保证这看起来象在变魔术。在您喝茶的时候，cfengine 将编辑文件、运行命令并创建符号链接。然而，cfengine 并不能代替您思考。在将配置文件放入产品之前，您仍需要编写它并对它进行测试。而另一方面，cfengine 所

做的事情几乎很少会造成损害。 cfengine 使系统收敛成为可能。但是，为什么需要收敛呢？我认为系统管理员首先需要保持内心宁静，其余的事情都是次要的。稳定性、可靠性和可预测性是达到内心宁静的方法。这看似过分简单，但询问任何一名好的系统管理员，每次您都会得到这个答案。收敛可以帮助管理员达到稳定性、可靠性和可预测性。虽然它不是达到那些目标的唯一方法，但在我的经历中它所带来的痛苦最少。稳定性可定义为防止无意的更改。当 cfengine 实现收敛时，它通过规则集做到这一点。良好设计的规则（请放心，cfengine 使良好的设计变得容易）只是使系统达到一种理想状态（如在 cfengine 规则中定义的那样理想）。例如，每次运行 cfengine 时，都可以重新创建关键的系统符号链接。或者，只要修改了本地副本，都可以从可信资源库复制 init.d 启动脚本。如果 cfengine 没有发觉进程在运行，那么可以重新启动它们。可靠性是使机器幸免于问题的能力。网络出故障或磁盘出故障是主要的可靠性考验。您的系统能幸免于那些问题吗？通过收敛到一种理想状态，您可以期望系统处于或接近那一理想状态。虽然可靠性不单由 cfengine 完成，但收敛可使之更加容易。因为稳定的系统不太可能受问题影响，所以稳定性也是可靠性的主要资产。最后，cfengine 的收敛使获得“空白”系统并使它达到期望的状态成为可能，并能够对关键系统进行快速临时替换。这里，有必要多讲讲期望和理想的状态。目前为止，我们假设仅有一种理想状态，它就是那个期望的状态。事实上，对于所有机器来说并没有理想状态。机器按任务、位置、连通性、用户、操作系统类型和版本等分类。每个系统管理员都用某些方法来对他的机

器分类，通常根据上述各项以及更多项（我们稍后将详细讨论类）。虽然对于所有机器来说没有一种理想状态，但对于给定的机器类有一个要达到的理想状态。这是 cfengine 的设计目标。cfengine 使寻找理想状态变得可行，并且易于收敛到该状态。可预测性是机器按照期望的那样来运转的能力。收敛通过使系统变得稳定且可靠来达到可预测性。而且，一旦新机器收敛到一种理想状态，就可以期望它象它所替换的旧机器那样工作。当机器将很快收敛到一个已知状态时，就可以很容易地估计添加系统或替换它们的调度成本。最后，为您系统编写的软件可以期望系统处于一种接近理想的状态。然后，系统资源处于一种可预测的状态，这样，软件就可以更多地集中在功能上，而更少地将每个系统作为怀有敌意的未知领域对待。

cfengine 的概述 Cfengine 由几个程序组成。在版本 1.6.3 中，主程序叫作 cfengine。cfengine 程序解释文件中的规则集，并执行那些规则请求的操作。严格来说，cfengine 程序只是 cfengine 语言的解释器，并且任何 cfengine 程序都只是那个解释器的脚本。在版本 1.6.3 中，还有一个名为 cfd 的守护程序及其同伴 cfrun。cfd 将在版本 2.0 得到加强，而在 1.6.3 版本中有许多未完善的地方。幸运的是，不用 cfd，就可以完成我所需要的任务（用信号通知运行 cfengine 并远程复制文件）。我宁愿在 ssh 上通过显式脚本启动 cfengine。它只比 cfd 少许慢一点，但更易于监控。当通过 cfd 启动时，cfengine 发出的错误和远程定义的类没有可靠地显示出来。对于远程文件复制，我发现 cfd 是不可靠的而且可能会危害安全性，所以我使用 rsync。cfengine 的作者 Mark Burgess 声明 cfd 在版本 2.0 中将有很大改进，而 rsync 将不再是必需的，但

在版本 2.0 推出之前，我建议避免 cfd。在开始使用 cfengine 之前，应该编译和安装它。对于可以使用它们的系统，RPM 都有可用的版本，而且还有一个可用的 Solaris 包（请参阅参考资料）。如果要存储文件的永久校验和（类似于 Tripwire 所做的事情），则应该带有 Berkeley DB 支持进行编译。然后，应该开始创建配置文件。主要的配置文件是 `/etc/cfengine/cfengine.conf`，它是在不带文件名调用 cfengine 时运行的（当在 1.6.3 中编译时，可以指定一个不同的缺省配置目录，但在 2.0 及更高版本中，`/etc/cfengine` 将是检查的唯一位置，所以您应该严格遵守这一点）。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com