Java 技术专区技术播客系列
Jon Gifford 日志、搜索和云计算的密切结合

GLOVER: 我是 Andy Glover，这里是 developerWorks 播客的 Java 技术系列。本期嘉宾是 Jon Gifford。他是 Loggly 的首席技术官和联合创始人。Jon，我觉得，从刚才询问您什么是 Loggly 时，我们的谈话就已经开始了。

GIFFORD: Loggly 是一个日志即服务系统。我们做的就是这个。
每个人都有日志。日志无处不在。它们存在于您所拥有的每台机器上。它们包含了一些有价值的数据，也包含了一些不太有价值的数据。并且管理它们实在是一种痛苦。

所以，基本上，我们的产品是一种让您更好更简单地管理日志的方式，它提供一些工具，让您可以使用它们完成无法通过其他方式完成的事情。

所以，基本上您可以将我们看作一个这样的地方，它可以放置所有系统和事件中的所有日志，甚至，您还可以直接通过我们跟踪浏览器事件以及诸如此类的事情。所以这真的是.....我们真的就是天空中的一个大日志倾卸场。

GLOVER: 在天空中，是的。让我再问一个问题。所以，日志和输出两种日志的应用程序已经遍布全球，只要有计算机的地方就有它们。

GIFFORD: 是的。

GLOVER: 但是，你们相对比较新。那么，是什么使你们突然决定，好吧，让我们组建一家公司来解决这个问题。为什么这个问题在之前没有解决呢？

GIFFORD: 我认为这个问题已经解决了。市场上有一大堆公司。在为您解决这个问题的公司中，Splunk 可能是最有名的。我们想要做的是尝试...尝试构建一个非常易于使用的系统，其中不会出现现有系统使用中的各种管理和让人头痛的问题。所以这就是我们在云中的原因。这就是为什么我们是一个服务，而不是一个下载的软件或设备.....

GLOVER: 好。

GIFFORD: ...这些也是解决这个问题的常见方法。但就我个人而言，真正促使我去做这件事的原因是，我专门从事搜索的工作已经有好长一段时间了；大约将近 15 年。并且处理过各种大型系统，高端多达约 900 台机器，而在低端，对于有效的产品而言，实际上 Loggly 可能是我在生产环境中使用过的最小的产品，它适用于 15 到 20 台机器。

所以当您在处理那么多机器时，无论是在云中还是在您自己的“colo”[或 “coloc”；co-location center...编者注。]中，您都需要知道发生了什么事情。所以我一直做的这件事最终似乎是为自己编写这个系统，以监视所有的这些箱子，并驱动仪表板和您要做的所有其他事情，从而确保您作为开发人员可以保持理智，或者如果您有操作人员的话，使他们保持理智。

所以，对我来说，这其实就像，我有这个心病，也在其他一些工作中思考了很多次，它只是好像，您知道，如果我这一次能做好，我就不必再做这件事了。

GLOVER: 就是这样。

GIFFORD: 所以，从某个角度来说，我想这是因为我比较懒。

GLOVER: 这不正是我们作为开发人员的定义吗？我们懒惰？ [大笑]

GIFFORD: 对。正是。所以，关于我们正在做的事情，单纯从技术角度来看，让我觉得有趣的另一件事情是，它是一个相当复杂的分布式系统，我喜欢从事这些系统的工作。我们使用我们自己的系统.....或者说，我们记录我们自己的系统，也记录它自己。 [大笑]

实际上，为检查我们的生产服务器的状态，我们需要去查看一个 Loggly 实例，所有生产实例的所有日志都来到这个实例中。实际上，我们将它作为一个单独的实例来运行，这在以前看来有点异常。但基本上，在我们刚开始时，代码并不是像以前那么稳定，我想这一点不用说您都知道。 [大笑]

如果有东西在我们其中一个箱子中失败了，我们偶尔会在生产中剔除它。我们会将它移动到它自己的实例，然后一切都变得正常。因此，构建高度可扩展的系统是一种非常有趣的技术挑战。我想我必须说，我特别爱将搜索作为一种研究数据的方式。

GLOVER: 是。

GIFFORD: 我想，我之前构建的很多像这样的系统已经有点侧重于度量。这太好了。它告诉了我各项任务的执行情况。但在出现问题时，它并不一定能给我很大的帮助，实际上我想找出是什么引起问题的时候。

而且我认为搜索对于解决这一问题也是一种特别强大的方法。所以每次利用 grep 和 awk，以及所有那些神奇的... Perl 和 Python，还有 Ruby 和所有那些神奇的东西，您可以使用它们从文本中提取信息。

但我认为，在某些方面，我认为只是单纯的赤裸裸的搜索有很多好处，这些好处是您使用所有那些其他东西无法得到的。

GLOVER: 我想提个问题，其实我想到了两个问题。
希望我会记得住它们。第一个是，您提到过，它在云中是日志即服务。

GIFFORD: 是的。

GLOVER: 那么作为一个开发人员，这对于我来说意味着什么呢？我如何利用 Loggly？它如何工作？

GIFFORD: 好。基本上，我们有几种不同的方式使数据进入系统。如果您是一个开发人员，您希望给我们发送数据，那么您可以使用 Syslog，对于大多数人，至少大多数 UNIX 开发人员则是 Syslog Appender。它是非常简单的协议，例如，您可以在 Java 上执行它，您可以用一个 Syslog Appender 来执行它。

在其他语言中，还有直接记录到 Syslog 的其他方式。如果您准备这样做，您只需要在您想记录的箱子上设置您的 Syslog Appender，并将日志转发给我们。这相当简单，并且是解决该问题的一种非常非常轻量的方式。

GLOVER: 当您说 Syslog Appender 时，我是说，这就像是 log4j，对吗？我只需要配置 log4j？

GIFFORD: Log4j。是。

GLOVER: 没错。这是适用于 99% 的现有 Java 应用程序的通用日志库。 [大笑]

GIFFORD: 没错。

GLOVER: 所以我会配置它，并只需要直接附加到 Syslog，然后将它输送到 Loggly。

GIFFORD: 没错。没错。并且这是很棒的，因为 Syslog 已经出现了很长一段时间了。使用 syslog 真的没有任何惊奇。市场上有多种不同的 syslog 守护程序。有普通的 Syslog。还有在 G 中的 Syslog，但都可以对它们进行配置，以便执行正确的操作。所以这是第一种方法，只需使用 syslog。

您也可以使用 TCP 或 UDP。如果您希望使用 Secure Sockets，也没有问题。另一种方法是，您可以直接记录到一个 REST 接口。那么您就可以通过 HTTP 将事件直接将日志发送给我们。而且，同样，这种做法也很简单。还有像....我不能想象一种语言会使执行一个 REST 调用变得困难。 [大笑]

在这种情况下.....同样，只有您可以做到。您可以使它尽可能可靠或快速。使用 HTTP 需要权衡一下可靠性和速度，但不是....我想

只有在每秒记录数千个事件时才需要考虑这一点，而大多数人并不会遇到这种情况。

这就是您实际的操作方式。因此，我们用一个 REST API 将事件放进系统。其实，我们用一个 REST API 基本上完成了您希望通过创建输入而在系统中执行的一切工作....输入基本上是... 您可以将输入视为对您有特定意义的一些数据的逻辑组合。

例如，对我们来说，我们的输入实际上是我们的应用程序，这意味着我们可以很快就直接深入研究某个应用程序的日志，做我们想要做的任何调查或绘图，或任何我们想做的事情。

但基本上，您在我们的网站上可以做的事情，您都可以通过 API 完成。而事实上，我们的网站正是构建于我们的 API 之上。同样的，这也是其中一个我们尽可能多地使用我们自己资料来提供服务的地方，并且我们也向其他所有人提供这些服务。

GLOVER: 对。那么，我可以通过某个 RESTful API 输送数据进来，我也可以直接通过 syslog 输送数据。

GIFFORD: 对。

GLOVER: 这只是有点类似于方程式的一侧。我将我的所有数据放进去。现在，您在前面提到，传统上，我查找日志或类似的东西，或者，编写一些准备将东西投放于此的 awk 脚本来进行查找，并构建另一个文档，作为与其他东西连接的桥梁，使我能得到我所需要的数据。

GIFFORD: 对。对。

GLOVER: 这是什么.....现在我的数据在其他地方。我该怎么办？我如何能获得它呢？您提到 API 存储库网站。我当然已经使用过您的网站控制台，但请您描述一下，现在 Loggly 拥有我的数据，那么接下来会怎么样呢？

GIFFORD: 好的。那么我们做的第一件事其实是，将它归档到 S3。我们可以将它归档到由客户拥有的存储区，以便使我们所看到的来自您的所有数据位于一个永久的位置，您可以随时访问它，例如，您可以使用它执行映射生成任务。但这真的是第一步。

并且这很重要，因为这意味着，随着我们越来越深入研究搜索系统本身，如果在堆栈中更下层的某个地方出现故障，我们知道，我们还拥有您的数据的副本，我们可以重建任何丢失或损坏的数据。

但基本上，进入我们系统的一切都会被索引。我们的索引方式是，我们使用 Solr，如果您不熟悉 Solr，您可能会熟悉 Lucene。但基本上您可以将它想象

为一个非常强大且灵活的搜索系统。Lucene 更偏向于是一个搜索库。Solr 在此基础之上添加了一大堆管理工具。所以我们使用 Solr 作为索引引擎。

您发送给我们的每一个事件在大约 10 至 15 秒内被索引。因此...然后它进入我们服务器上的索引中。然后会发生的是，您可以进入我们的网站，只需键入搜索，然后搜索查询。您可以搜索一下，我的最爱，我总是这样做，我搜索 “exception”。 [大笑]

然后就会得到与“exception”相匹配的结果。现在，这是一个非常简单的示例，您可以使用 FRAP 对所有其他事项这么做，只不过来自您的所有机器的所有日志都将同时被搜索。

不仅如此，来自您所有应用程序的所有日志，即使使用不同的语言，也都将以不同的语言实现，然后将会被搜索。

在这种情况下，您也会获得来自 Ruby 和 Python 的异常。所以，真正的想法是，如果您能想象...我的意思是，我想，可视化这一点的方法之一是，如果您想象一下，您将所有机器记录到一个内部装载的驱动器，当您键入 grep 时，您就可以查找在该驱动器上的每一个文件。

GLOVER: 没错。

GIFFORD: 这是一种可视化的方法。现在，我们这样做的原因其实只是，它看起来像，将一切都一起放在一个地方，这个真正的需求至少在大型系统中很难解决，甚至对于中等规模的系统也不容易。

要使它工作得可靠真的很不容易。所以我们想做的是，基本上，我们说通信是相当可靠的。实际上网络流量非常顺畅。 [大笑]

网络流量是比其他东西更可靠。所以，基本上，我们决定，如果您能将自己的资料发送给我们，那么我们可以负责在收到资料之后构建一个非常可靠的系统。我想如果您曾经构建过复杂的多主机系统，您就会明白，有很多时间都花在试图保持系统稳定上。 [大笑]

所以我们花了很多时间来使这个系统稳定，我想它是相当不错的。很明显，您可以....总是有可以改善的地方。但它基本上消除了开发人员或管理员需要花费大量时间完成的工作。

我们的想法是，如果我们能够使它可靠，您只要将数据给我们就行了，然后您就可以把需要耗费您大量时间完成的那一大堆事抛在脑后。然后.....对不起，我前面提到了用户界面，您可以在那里访问该网站，进行搜索

并查看结果。我们已经完成的是，我们试图使它看起来尽可能像一个规则终端。

GLOVER: 是。

GIFFORD: 如果您是一个使用 UNIX 的用户，会觉得它很熟悉。您可以手动执行帮助。它看起来就像一个规则的黑底白字的旧式日志，而底层有一点点神奇的东西在支持。它实际上是在您的浏览器中运行。它只是一个有趣的 JavaScript。

它在您的浏览器上运行，这意味着我们可以.....当显示结果时，我们可以做到在一个规则终端上无法真正做到的事情，并且我想您只使用规则的 UNIX 管道或者 HTML 也实现不了。 [大笑]

例如，我们可以直接在界面中绘图。您可以不进行搜索，如果您直接使用图形进行描述，您就会得到一个图形，这是在您选择的任何一段时间内针对您所搜索的任何内容的一个匹配计数。

这就是用户界面。有一个仪表板，显示我们正在接收事件，等等等等。但基本上，您可以把用户界面看作一个超级强大的 shell，它刚好在您的浏览器上运行，又恰好能够同时处理您的所有日志。

然后，为了完成这项操作，我们也有一个 API，让您可以做同样的事情。因此，我们在 API 中有搜索和图形，以及一些其他东西。您在 shell、浏览器中所看到的，同样刚好命中那些 API 。所以，您在我们的 shell 中可以看到的所有用法，您都可以通过我们的 API 来自己完成，它只是一个 RESTful API。

GLOVER: 那么让我回到搜索的问题。您曾提到 Solr。显然你们提供的价值是，嘿，看。现在，正如您所知，正如您所说的，所有的日志都在天空中的一个神奇的数据库里。

GIFFORD: 对。

GLOVER: 现在我可以在它上面进行搜索。怎么样，比如说，不同？
我要问一个愚蠢的问题，正是如此，我们希望对高科技的搜索和 Solr Lucene 的复杂性了解更多一点。但这比起您或我将我的所有日志放进我的 SQL 或 Oracle 或某个东西里有什么不同呢？好的，因为 DB2 据说也提供搜索，对吧？有什么区别？让我们谈得更深入一点，再说一下文本搜索的复杂性。

GIFFORD: 当然可以。作为一个...对于在搜索这一行当干了很长时间的人，这些东西会有点难以表达。我已经将它内在化了，但我会尝试一下。 [大笑]

您知道吗？我想从整体上讲是这样的。对于那些您并不真正确定您要查找什么的非结构化数据来说，您不能直接将它扔进数据库。举一个 Apache 日志的示例。每个人都有 Apache 日志。它们有少量较标准化的格式，您可以执行数据提取并设计数据库流转化器，它将提供状态代码和[听不清]，还有用户代理，等等。

您可以很容易就做到这一点，但在那种请求中还是有一些不容易理解的东西，在某种意义上来说，是在同一层面的东西。例如，请求本身有一个路径，但路径可能在系统上的任意位置，它可能是一个 CGI，也可能是一个静态文件，等等。

那么您最终得到的是，无论如何努力尝试，您最终得到的是数据库中的一个或多个字段，并且无法再进一步分解。但其实您可以。在关系数据库中，您可以尝试与原子字段匹配。您可以执行一个“like” 查询。

但这些，语言的表达能力非常有限。因此，我看待它的方式是，搜索实际上是...在某种程度上，搜索是非结构化数据的金弹。对于非结构化数据和关系数据库以及类似的东西，到处都能看到很多工作在进行。

我认为，从根本上讲，搜索是能够在这些数据存储中查找资料的最佳方式。因此，我认为它需要....作为用户，看待这个问题要有一个思维转变，不再是我需要包含这个值的一个字段。

再仔细地看一下，我需要尝试并构造一些随机表达式，为我在某种随机的文本 blob 中找到我想要的东西，但筛选掉我不想要的东西。我想，搜索用得越多，就越会习惯作出这类决定。我认为，Google 已经培养人们将搜索视作一个非常简单的事情。

但是..... Web 上有很多问题，人们只是浏览并希望找到一个 Web 页面，这是千真万确的。如果您本来是用它来进行数据挖掘，Google 和 Yahoo 等其他人教您的东西，基本上是做一个非常简单的搜索，然后只是希望第一个在顶部弹出的就是正确的，这实际上是一种错误的方法。 [大笑]

所以，如果您带着搜索能为您做些什么的心态进入搜索，那么您很可能会对开头几次的搜索尝试有点失望，因为您会得到很多垃圾。对不对？

所以您必须习惯于这种想法，搜索.....可用的搜索语言，表达能力非常强，即使范围搜索语言可以帮助您深入，

确定您要查找的内容到底是什么。以下是一个示例。

那么我们，回到我喜欢的异常搜索，如果我只是在我们的生产日志上搜索异常，我会得到了很多基本上属于噪音的异常。为了向您举一个例子，我们将碎片、索引碎片从一个箱子复制到另一个箱子。

这些副本会周期性地发生故障。现在，这听起来像是一件坏事，但事实上，这根本不是什么大事。碎片在原始箱中仍然可用，并且它刚好没有在目标箱中做出任何变更。几分钟后，我们将再次尝试。

GLOVER: 好。

GIFFORD: 当我去搜索异常时，我得到很多那种异常。要把这些筛选掉，有一个真正简单的方法，就是执行同样的搜索，搜索一个异常，然后删除带有这个特定问题的网络超时异常日志的类名称。那么，我会看到其他...它会删除干扰的信息，留下一些真正有用的异常信息。。

这时，我可以进一步细化我的搜索，我可以添加更多 `not` 子句，删除更多类，假如说，我注意到一个特别有趣的模块，或者说似乎是一个有很多问题的包，我甚至还可以扩展它。

我可以将它扩展为，包括来自那个包的任何东西，那些东西，您真的.....您没必要知道自己到底会找到什么。

搜索变得非常强大，因为它这么快就能为您找到 `1000` 个结果。您真的可以就只是坐在那里，利用它来玩一下，然后添加或删除条件，以您的方式围绕内容进行导航，对于传统的面向数据库系统而言，这些内容似乎只是一个.....乱七八糟的大泥坑。

GLOVER: 对。

GIFFORD: 所以我觉得，就像我说的，我已经做了这么久，我无时无刻不在自己的脑袋里构造搜索。[大笑]

因此，有点，您知道，有点难以表达，但它是那种做得越多，就越能了解如何做得更好的事情。

GLOVER: 我这么问吧。回到.....我是一个 `Java` 开发人员。我好奇。因为我们都....嗯，是，我相信在感觉没问题之前，我们所有人都会有这个问题，我需要在芝麻里挑绿豆。

有些东西出错了，所以不管它是什么，我都 FRAP，我每次检查一个日志。这样那样。所以我很喜欢。对吗？这是有意义的。我必须.....我该如何开始呢？我要下载一些库吗？我该如何前进呢？

GIFFORD: 好吧，最简单的开始方法是使用 Syslog。然后，完全不需要下载。它只是 Syslog Appender。您必须访问网站。注册。我们会向您提供关于如何设置 Syslog 的一些信息。但仅此而已。

如果您使用 log4j，我假设是这样，那么它就只是一个在内部使用 Syslog Appender 的情况.....就像将它放进 log4j 配置中，基本上就大功告成了。然后您就可以回来，并在它们所在的任何箱子上搜索来自您的所有 Java 应用的所有日志。

这很简单。我不想....您必须在网站上处理一大堆资料，授权设备，等等，等等。但是，这只是我们产品中的产品。

GLOVER: 管理，对吗？

GIFFORD: 是的，基本上从您的角度看，它就是设置 Syslog 就可以了。或者说实际上，还有.....我知道至少有一个，很可能不只一个 Syslog.....对不起，是 Log4j，并且它会使您能够访问更多 API，所以举例来说，您实际上可以编写一个 Java 应用程序来命中 API。

GLOVER: 说到开发人员，注册了管理和诸如此类的东西，您就会有一个免费的开发人员帐户，对不对？

GIFFORD: 我们是这样做的。他们有一点限制，但他们.....我试着记起来他们是什么。我想他们是一个星期每天 400 或 500 MB 的数据。差不多是这样。所以，您可以保持一个合理的数据量。您可以在那里保持您所需足够的数据，并决定它对您是否有意义。我还应该说一些东西。对于将自己的资料发射到云中这样的想法，有些人并不喜欢。

这可能是一个破坏因素。我们已经听很多人说过，我爱这个。我喜欢它可以做的事情，但我想在内部使用它。而这并不是我们采用的路径。不幸的是，我们不得不说，我们不这样做。我们可能永远也不会这样做。但我们肯定现在不会这样做。所以我觉得人们....市场上有其他具有不同特点的产品。我们在地球上并不是惟一的。

GLOVER: 好的。我们在哪里可以找到有关 Loggly 的更多信息呢？

GIFFORD: 就是 www.loggly.com。

GLOVER: 而且你们有一个博客。

GIFFORD: 我们有一个博客。是的。

GLOVER: 我还必须补充一句，我最喜欢 Loggly 的东西之一是视频中的吉祥物。 [大笑]

GIFFORD: 海狸 Hoover。

GLOVER: 是。海狸 Hoover 是一只有趣的动物。 [大笑]

我们将让....我强烈推荐正在收听的各位观众去看看视频。对 Loggly 的那种解释。我认为你们也会喜欢。非常棒。

Jon，我知道，当我说这很有趣时，我说出了大家的心声。我觉得你们所做的是.....就我个人而言，我可以说我正在使用 Loggly，而且我发现，您提到的有关能够将所有数据放在同一个地方的所有事情都是非常有价值的。

我觉得...我将要说，在我心目中，Loggly 是日志的 Google，但我想我们要小心，因为在说到搜索是多么强大时，您的确提及 Google 如何有可能将您引导到错误的方向。 [大笑]

所以我很想看到我所描述的，Loggly 是日志的 Google。

GIFFORD: 其实我想还有一件事要说。

GLOVER: 是。

GIFFORD: 我尝试将我们视为为日志的某种东西，我思考它的方式是，我们实际上有一点点，如果我们和其他任何人完全一样，我们是有点像 Salesforce。我们想要的是类似于，Salesforce 是 CRM 即服务产品。

GLOVER: 是。

GIFFORD: 人们....它有一个 API 且人们在它上面构建应用程序。这是我们针对日志想实现的目标。所以我还没有提到它。我本应该早一点提到它，但我们也可以让[听不清]。所以，您知道吗？

在理论上，我们可能是天空中的一个神奇的数据库，使用 API 您可以基于非结构化数据构建应用程序。而我们真正想要做的是，还要鼓励开发人员快速参与并开始使用我们....您知道吗？我觉得我们是天空中的时间序列数据库。

GLOVER: 非常有趣。是的。我肯定会推荐现在的开发人员在他们的维基中查看 Loggly 博客。在引导人们了解更多的信息库、API 等方面，你们已经做得很好。了解

Loggly 然后认识你们并利用您们的服务，这一定很有意思。我喜欢天空中的神奇数据库。 [大笑]

GIFFORD: 是的。致力于这项工作其实会更有乐趣。

GLOVER: 我敢打赌是这样的。很好，我的嘉宾一直是 Jon Gifford。再提醒大家一次，他是 Loggly 的 CTO 和共同创始人。我是 Andy Glover。这里是 developerWorks 播客的 Java 技术系列。感谢收听。