基于网格监控系统的动态监控间隔的研究

An Dynamic Monitoring Interval for Grid Monitoring System

梁 鸿　郭晓菲　(中国石油大学计算机与通信工程学院 东营 257061)

摘要:网格环境是由很多元素构成的高度复杂的分布式计算环境。对于网格来说,监控其中每个计算节点的状态是至关重要的。然而频繁地监控资源的状态信息会增加系统的负担。本文提出一种动态调整监控间隔的方法,来降低监控服务所需的花费,并利用动态监控间隔来准确地获取监控事件,最后通过在网格环境下进行实验,对该方法做出客观的评价。

关键词:网格 监控 动态监控间隔

网格环境是一种分布式计算,它需要将地理上分布、系统异构的多种计算资源通过高速网络连接起来,共同解决大型应用问题[1]。所以对于网格来说,监控其中每个计算节点的状态是至关重要的。

在网格计算环境下,监控系统不仅是保持系统健壮性、可用性的手段,而且支持网格环境下的资源管理,是网格运行不可缺少的组成部分。它的资源监控系统与其他功能模块构成一个有机的整体,共同支持网格的运行[2]。

网格资源监控可以显示网格组件在某一时刻的状态。为了保证有效监控,监控必须是"端到端"的,也就是说,在应用端所有组件都必须监控,包括软件(如:应用程序、服务、中间件和操作系统)、终端主机的硬件(如:CPU、磁盘、内存和网络接口)和网络(如:路由器、交换机或端到端的路径)。

由于网格资源的动态性,即资源可以动态地加入或退出,故对网格资源进行监控也要考虑到动态性,而且监控必须实时,因为节点的信息是动态变化的,然而频繁地监控会增大系统负担,系统效率也会降低。例如:如果我们监控一个短期内没有较大改变的资源状态,就会增加不必要的监控负担。否则,监控系统就会得到不准确的信息。这是急需解决的一个问题。

1 网格监控结构

网格监控结构(Grid Monitoring Architecture,GMA)是全球网格论坛性能工作组提出的结构。GMA是基于目录服务的网格监控框架,这个框架由目录服务、生产者、消费者三部分组成,如图1所示。

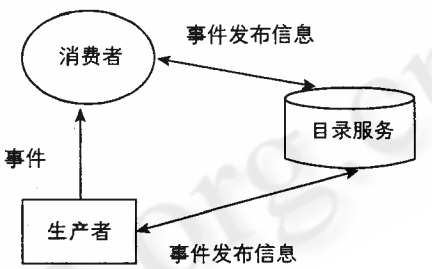

图 1　基于生产者/消费者的监控模型

在GMA中,监控数据的基本单位是一个事件。事件是一个经命名的、有时间戳的结构。此结构可以包含一个或多个数据条目。这些数据关联到一种或多种资源,如内存或网络的使用率,或应用特定的数据,如两个矩阵相乘用掉的时间。产生可用事件数据的组件叫生产者,请求或接受事件数据的叫消费者。目录服务用于发布什么事件数据是可得的以及是从哪个生产者获得的[2]。

在这个模型中,若想获得事件的数据,首先要通过目录服务找到该事件的生产者的静态信息,比如事件类型、主机地址等;然后再向该生产者发出请求获得相应数据。

由于GMA仍然处于成型阶段,因此没有最终确定

的架构、组件和协议。当前的一些网格监控系统没有解决基本的资源开销问题，，只是得到更多更准确的资源状态信息，针对这个问题，我们提出动态调整监控间隔的方法。

在网格环境中，资源监控系统要实时监控资源的状态，周期性地读取和发布各个节点的信息。然而在实际应用中，周期性地监控必然会增加系统的负担，监控信息地传递和数据的集成必然会占用系统和网格的资源，导致系统和效率降低。所以，本文提出一种动态调整监控间隔的方法，来解决网格监控系统中由于周期性监控所引起的系统开销问题。

2 监控间隔的研究

在网格环境中，监控系统不断地向资源状态信息发出请求，当新的信息产生的时候，监控系统中的状态信息就要更新。其实，资源状态信息和其它监控组件的改变都依赖于 CPU 的改变，这不仅在分布式计算中是这样，同样在网格环境中也是如此。因此，如果我们观察 CPU 的利用率，就可以大体上预测其他资源的改变。例如：假定 CPU 的利用率几乎没有改变，那其它的资源状态也不会有大的改变。如果 CPU 的利用率有很大的变化，那其它的资源状态也会有很多变化。换句话说，网格监控事件的改变依赖于 CPU 的改变。因此，当 CPU 的利用率有较大改变时，其他资源应该立即监控。而且，如果我们只观察 CPU 的利用率，在利用 CPU 的高性能系统中就会频繁地监控。在这种情况下，不必要的系统花费就会增加。为了防止这种情况的发生，下面的研究将会证明监控间隔随着 CPU 的利用率的改变而改变是非常有效的。

如果 CPU 的利用率动态改变较大，意味着资源的状态将会有一个较大的改变。这种情况下，我们应该调整监控的间隔，让资源监控服务获取更新后的资源状态信息。即，如果 CPU 的利用率现在非常高，但是突然回落到 0，这意味着资源的状态信息已经改变了。因此应该从这一点开始，改变监控间隔。CPU 利用率的改变值是通过 CPU 利用率的预测值和当前 CPU 的利用率的差值计算出来的。这时候，如果差值很大，我们会把它作为系统中一个大的改变，来控制监控间隔[3]。

简要来说，这种方法就是比较 CPU 利用率的预测值和当前 CPU 的利用率。当前 CPU 的利用率表示为一个百分数，$e(k)$ 是 100% 减去当前 CPU 的利用率所得的结果，即空闲率。在这个研究中，不仅要考虑当前的状态，而且还要考虑以前的状态。换句话说，用 PID 参数来作为动态的阈值[4]，预测 $u(k)$ 作为下一次 CPU 达到 100% 时的被请求的资源数。

$$u(k)=k_p\times e(k)+k_i\times\sum_{j=1}^{k}e(j)+k_d\times(e(k)-e(k-1)) \quad (1)$$

这里，每一个 PID 参数都设为 $k_p=-0.5$，$k_i=0.125$，$k_d=-0.125$。准确地说，在 PID 控制器中 $u(k)$ 通过 CPU 的空闲率 $e(k)$ 来表示。除此以外，$|u(k)-u(k-1)|$ 表示通过当前的 CPU 信息，CPU 的利用率会改变多少。

$$\begin{aligned}|u(k)-u(k-1)| &= |k_p\times e(k)+k_i\times\sum_{j=1}^{k}e(j)+k_d\times(e(k)-e(k-1))\\ &\quad -(k_p\times e(k-1)+k_i\times\sum_{j=1}^{k-1}e(j)+k_d\times(e(k-1)-e(k-2)))|\\ &= |k_p\times(e(k)-e(k-1))+k_i(\sum_{j=1}^{k}e(j)-\sum_{j=1}^{k-1}e(j))\\ &\quad +k_d\times(e(k)-2e(k-1)+e(k-2))|\\ &\leqslant k_i\times e(k) \quad (2)\end{aligned}$$

如果系统稳定，$e(k)$，$e(k-1)$，和 $e(k-2)$ 都是相等的。因此我们可以通过 $u(k)$ 得到 $|u(k)-u(k-1)|$。简而言之，我们可以简单地比较 $|u(k)-u(k-1)|$ 和 $ki\times e(k)$ 的值。当两个值相等或者 $|u(k)-u(k-1)|$ 小于 $ki\times e(k)$，它表示 CPU 的利用率和前一个时期相比没有较大的动态改变。然而，相反的情况，就意味着系统的状态信息有了明显的改变。因此，就应该调整监控的间隔。这就是我们动态控制监控间隔的方法。

3 实验结果及其评价

在网格环境中，我们通过监控系统记录下某段时间内的 CPU 利用率，通过公式(1)和(2)来分析验证我们所提出的动态调整监控间隔的方法。实验数据如表 1 所示。

表 1　系统的变化率

CPU 利用率	e(k)	u(k)	u(k-1)	|u(k)-u(k)|	$k_i \times e(k)$
53	47	117.75	128.75	11	5.875
57	43	96.875	102.125	5.25	5.375
56	44	101.825	96.875	4.95	5.5
54	46	111.875	101.625	10.25	5.75
53	47	117.375	111.875	5.5	5.875
52	48	122.875	117.375	5.5	6
*51	49	128.5	122.875	5.625	6.125
*61	39	78.125	128.875	50.375	4.875
60	40	82.375	78.125	4.25	5
59	41	87	82.375	4.625	5.125
*58	42	91.75	87	4.75	5.25
*53	47	118.875	91.75	27.125	5.875
54	46	112.25	116.875	4.625	5.75
55	45	107	112.25	5.25	5.625
56	44	101.875	107	5.125	5
*55	45	106.75	101.875	4.875	4.875
*60	40	83.125	106.75	23.625	5
61	39	78.125	83.125	5	4.875
60	40	82.375	78.125	4.25	5
61	39	78.125	82.375	4.25	5.625
60	40	82.375	78.125	4.25	5.75
55	45	106.25	82.375	23.875	5.625
54	46	112	106.25	5.75	5.75
55	45	107	112	5	5.625
56	44	101.875	107	5.125	5.5
55	45	106.75	101.875	4.875	5.625

根据以上数据所得的图表如图 2 所示。

由表 1 和图 2，我们可以看到在这段时间内，系统的变化比较平稳，即使资源状态改变了一点，它也仍然意味着已经改变了。因此可以利用以上提到的方法来调整监控间隔。

当 CPU 的利用率改变较大的时候，就如表 1 中用“*”所标注的间隔那样，$|u(k)-u(k-1)|$ 大于 $k_i e(k)$，系统应该控制它的间隔。例如：当资源信息的可用性从 46% 增加到 60%，$|u(k)-u(k-1)|$ 的值会大于 $k_i \times e(k)$ 的值。当 $|u(k)-u(k-1)|$ 的值非常大的时候，如果有较大的改变，如图 2 所示，边缘会非常的高。这时，$|u(k)-u(k-1)|$ 大于 $k_i \times e(k)$，我们会控制监控间隔变小点。当 $|u(k)-u(k-1)|$ 的值小于 $k_i \times e(k)$ 的值时，资源状态的改变很小，因此应该重新设置监控间隔，使其变大。

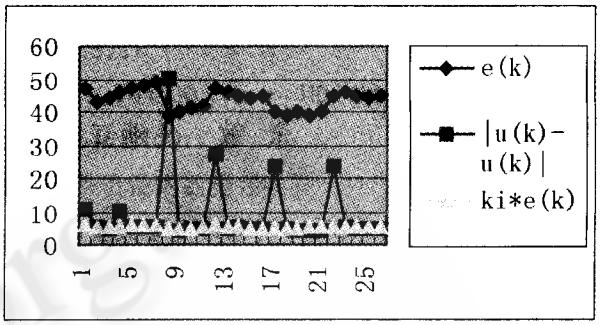

图 2　系统参数的变化

通过实验，系统监控本身在每一秒都精确显示当时的资源状态。因此，这种方法对于控制监控间隔是很和谐的，可以运用到真实的网格环境中。

4　结论

本文提出一种动态调整监控间隔的方法，来降低监控服务所需的花费，并利用动态监控间隔来准确地获取监控事件，通过在真实的网格环境下进行实验，我们发现不必要的系统开销明显降低，事件的准确率也得到满足。

参考文献

1　都志辉、陈渝、刘鹏，网格计算，北京 清华大学出版社，2002。

2　徐志伟、冯百明、李伟，网格计算技术，北京 电子工业出版社，2004。

3　Monitoring event，“Discovery and Monitoring Event Description (DAMED - WG)”，http://www - didc. lbl. gov/damed/

4　PID Tutorial，http://www. engin. umich. edu/group/ctm/PID/PID. html.