

Un Monitor de estado de servidores en un ambiente TCP/IP

Andrea Persico¹ - Sonia Flores - Marcela Printista

L.I.D.I.C.*

Departamento de Informática

Universidad Nacional de San Luis
Ejército de Los Andes 950
5700 – San Luis
Argentina

e-mails: {persicoa, sflores, [mprinti](mailto:mprinti@unsl.edu.ar)} @unsl.edu.ar
++54 – 2652 – 420822 Int. 253

RESUMEN

La performance de las redes de computadoras se presenta como un tópico atractivo, cuyo análisis es una tarea de gran interés en la actualidad. En este trabajo se presenta el diseño de un monitor de estado de servidores en una red de computadoras, trabajando en un ambiente TCP/IP con el objetivo de obtener información la cual describe la condición en que se encuentran los servidores a través de distintas métricas tales como memoria disponible, utilización de CPU, balance de carga, etc.

Palabras Claves: Monitor, Performance, Métrica, Threshold, TCP/IP, Socket, Cliente-Servidor.

¹ Becaria del LIDIC

* Grupo subvencionado por la UNSL y ANPCyT (Agencia Nacional para Promover la Ciencia y Tecnología).

Introducción

En la tecnología de computadoras el enfoque tradicional de las últimas décadas fue evolucionar hacia multiprocesadores o supercomputadoras[5,7]. Sin embargo, en los países desarrollados existe una tendencia contrapuesta al enfoque tradicional que plantea ejecutar las aplicaciones distribuyéndolas entre varios procesadores de diferente poder computacional. Las argumentaciones a favor de esta tendencia se basan en consideraciones económicas y de eficiencia.

La información sobre redes de computadoras se encuentra dispersa a lo largo de diversas publicaciones, todas ellas producidas en los países de avanzada tecnología.

Los aspectos metodológicos del presente trabajo contienen etapas de estudio intensivo, recurriendo a esta bibliografía especializada sobre tendencias actuales y problemas usuales en los temas de interés. Estas etapas de estudio sirvieron para la determinación, selección e implementación de las alternativas más convenientes en el diseño del monitor[1,2,3,6].

Desde el punto de vista de los usuarios, se ha difundido el uso de sistemas en red con el objetivo de compartir archivos, recursos, información (internet), el uso del correo electrónico, etc. Sin embargo las posibilidades que brinda la conexión en red de un conjunto de computadoras son mucho más amplias que la difundida entre usuarios corrientes. Aún resta generalizar cómo realizar las ejecuciones remotas de programas sobre una red, el acceso a bases de datos remotas, cómo generar ambientes computacionales distribuidos, cómo desarrollar computaciones paralelas y cómo realizar desarrollos sobre redes que involucren inteligencia computacional.

En la administración de una red existe una jerarquía de protocolos los cuales son conjuntos de reglas que describen una secuencia de acciones que inicializan y controlan la transmisión de datos a lo largo de los medios físicos. Una de las principales familias de protocolos es la de TCP/IP[1,2,3,6].

Para el desarrollo de este trabajo se ha seleccionado como Modelo de Programación, el “*Modelo Socket*”, a través del cual, se podrá acceder completamente a la funcionalidad de TCP, UDP y a la de otros protocolos[1,2,3,6].

El objetivo de este trabajo es definir un ambiente de control de servidores capaz de brindar suficiente información sobre el estado de los diferentes nodos que se quieren monitorear. La información describirá la performance de los nodos a través de distintas métricas tales como memoria disponible, utilización de CPU, balance de carga, etc.

Diseño y Arquitectura del Monitor

La arquitectura del monitor está basada en un modelo Cliente-Servidor, según la figura 1. Se puede observar la presencia de nodos (Y, Z, W) los cuales actúan como clientes, y un nodo (X) que se comporta como servidor. Las flechas que conectan los clientes con el servidor están representando el pasaje de mensajes, transmitiendo la información de estado requerida por el servidor.

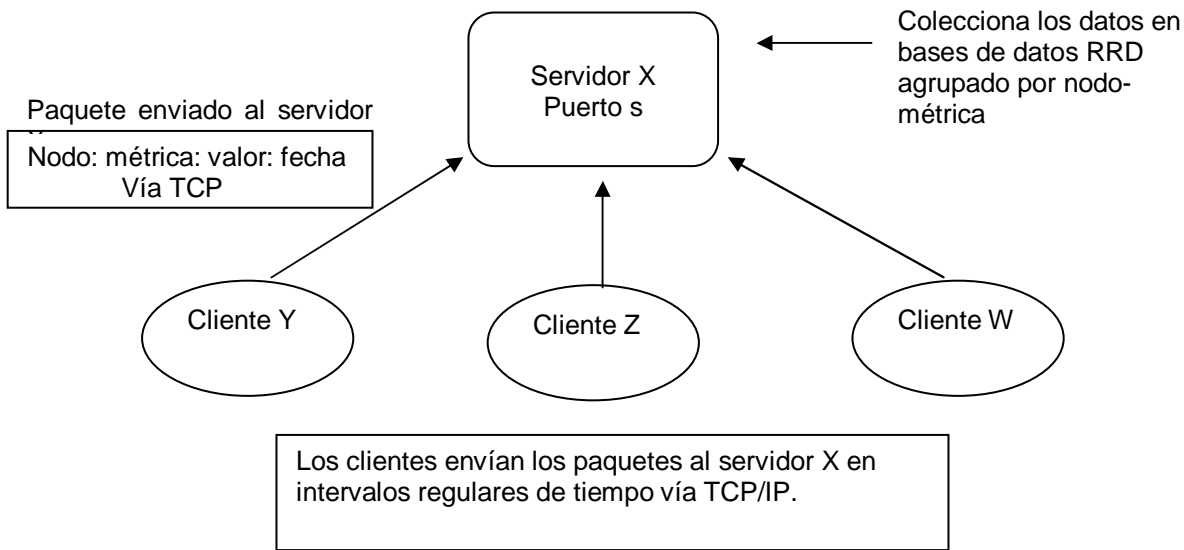


Figura 1. Arquitectura del Monitor

La información requerida por el servidor, es transmitida en paquetes. Cada paquete contiene en su espacio dedicado a los datos, tal como lo muestra la figura 2. El paquete contiene un identificador del nodo que envía la información de estado (IP), la métrica que se está monitoreando, el valor obtenido para la misma, y el tiempo de la muestra (Timestamp).

| Encabezado de paquete | Nodo (IP) | Nbre. Métrica | Valor de Métrica | Tiempo de Muestra |
|-----------------------|-----------|---------------|------------------|-------------------|
| | | | | |

Figura 2. Formato de paquete

El encabezado del paquete, en el protocolo que se establece para esta comunicación, contiene la información necesaria para realizar la comunicación cliente-servidor.

En la siguiente sección se presentan detalles del funcionamiento del Servidor y el funcionamiento del Cliente.

Funcionamiento del Servidor

El servidor es el proceso que monitorea un conjunto de nodos. El servidor espera por la comunicación de los clientes, en un puerto *s* conocido por éstos. Los paquetes que recibe concurrentemente de los nodos, los guarda en bases de datos round robin para su posterior procesamiento.

Es necesario determinar un umbral (threshold), para cada una de las métricas, de modo que de acuerdo a este umbral se establecerá la precisión del monitor para evaluar la performance de un nodo.

Estas facilidades alertarán al administrador del sistema sobre posibles caídas en la performance de los nodos, con el objeto de prevenir, por ejemplo, situaciones no deseadas.

Funcionamiento del Cliente

El cliente es un proceso que se encuentra recolectando información del estado del nodo en que se encuentra. El tipo de métrica, cuyos valores el cliente se encuentra recopilando, lo establece el servidor bajo demanda. Por ejemplo, en un intervalo de tiempo recolectará memoria disponible, en otro utilización de la CPU, etc.

Esta información se envía, en forma de paquetes, al servidor que está esperando en el puerto *s*.

Tipo de comunicación

El modelo de comunicación elegido para este monitor es socket con conexión usando TCP/IP, consecuentemente es una comunicación sincrónica.

Dentro de este esquema, el servidor da a conocer un puerto por el cual los clientes solicitan la comunicación. Una vez establecida esta comunicación, el cliente comienza a transmitir la información solicitada, y seguirá transmitiendo hasta que el servidor requiera otro tipo de métrica.

El proceso cliente duerme durante un período de tiempo y luego vuelve a realizar el mismo procedimiento.

Conclusiones

Con la realización de este trabajo se espera poder monitorear los nodos, teniendo una interfase gráfica (Web), la cual muestre el estado de los nodos de la red, proveyendo alertas al administrador sobre bajas en la performance del sistema.

Tal como se presenta el trabajo se puede observar que no existe replicación de los datos obtenidos por el servidor, y ante posibles caídas del demonio que monitorea los nodos en el sistema se puede presentar una ambigüedad respecto del estado del monitor.

Si bien el sistema introduce posibles sobrecargas, debido a las conexiones TCP/IP, puede mejorar notablemente la administración de un gran sistema de red. En base a esta información se podrá visualizar tendencias anómalas en el comportamiento del sistema en un tiempo previo al momento en que todo el sistema experimente una baja no aceptable en su performance o en el peor de los casos colapse.

Bibliografía

1. Comer D. E. "Internetworking with TCP/IP Vol I Principles, Protocols, and Architecture" - Second Edition - Prentice Hall - 1991.
2. Comer D. E. "Computer Networks and Internet" - Second Edition - Prentice Hall - 1999.
3. Comer D. E. and Stevens D.L. "Internetworking with TCP/IP" - Prentice Hall.
4. Stallings W. "Data and Computer Communications"- 6a Edición - Prentice Hall.
5. Tanenbaum A. S. "Modern Operating Systems" - Prentice Hall.

6. Tanenbaum A. S. "Computer Networks" - Third Edition - Prentice Hall – 1996.
7. Wilkinson B. And Allen M. "Parallel Programming Techniques and Applications using Networked Workstations and Parallel Computers" – Prentice Hall -1999.