Using Sound to Analyze Hardware Operation: A Progress Report

Robert Karns Henry
East Tennessee State University
robert.henry@acm.org

Abstract

Designers of massively parallel cluster computers help to ensure system reliability by using sensors to monitor a cluster's component systems. Current PCs ship with built-in data acquisition devices that monitor system parameters such as voltage, temperature, and hard drive status, in order to anticipate possible system failure. One modality for PC monitoring which is currently ignored is sound. Hard drives often produce distinct noises when they deteriorate, as do devices such as fans. The research described here seeks to develop a system that monitors the sounds produced by PCs during standard operation, identifies abnormal sounds using wavelets, and sends a message when an abnormal condition exists. The project is in progress and completion is anticipated in summer of 2005.

1. Introduction

In recent years, personal computers (PCs) have found use as nodes in massively parallel cluster computers. One problem with clustering PCs is an increased likelihood that symptoms of impending failures may go unnoticed, due to the reduced level of hardware supervision that individual PCs in clusters receive to make them easier to manage. This reduction in the level of hands-on PC monitoring is being compensated for, in part, by the equipping of modern PCs with data acquisition devices (DAQs) that monitor phenomena such as temperature, voltage, and fan speed. In addition, data reported by S.M.A.R.T. compliant hard drives can help to identify abnormal drive performance.

One potentially useful modality for detecting PC error that has not yet been exploited is sound. Sound has previously been used as a basis for detecting abnormal system operation in automotive and mechanical engineering. Modern automobile engines monitor vibrations caused by detonation using an accelerometer and the accelerometer signal used to control spark advance. [6] Another project has used signals from accelerometers to monitor vibration helicopter transmissions. [7] In that project, accelerometers installed near critical bearings and gears in the transmission transmit a signal that is analyzed for abnormal sounds.

Many hard drives make sounds when they begin to fail, as do fans and other components. Users often use abnormal sounds as a basis for reporting problems with their desktop and laptop PCs; computer service people routinely use abnormal sounds as a basis for replacing parts. The task of monitoring sound, unfortunately, becomes much more difficult in clusters, where the physical separation of users from backend systems and the number of systems makes it much less likely that users will hear abnormal noises.

Proceedings of MASPLAS'05
Mid-Atlantic Student Workshop on Programming Languages and Systems
University of Delaware, April 30, 2005

The research described here seeks to develop a system that will allow a computer to monitor the sounds that it is producing, determine if those sounds are associated with hardware failure, and send a message to the system administrator or cluster network manager when a hardware failure is imminent. The author hopes to conclude his study by August, 2005. The balance of this report describes the research's strategies for sound capture and analysis, and the initial experimental design.

2. Using Sensors to Capture Sounds from PCs

Attaching accelerometers to each component of a PC is impractical due to the expense and complexity. A cheaper, simpler alternative is to install a microphone inside a PC's cabinet that can monitor airborne sounds emitted from components. In this case, the microphone is connected to the microphone input on the computer's sound card. The sound card samples the input signal and converts it into a digital signal for analysis. Since the project is designed for a Linux-based cluster system, the sound card will be accessed using a standard Linux sound API.

Currently, there are two standard sound systems in use in Linux systems: the deprecated Open Sound System (OSS), and Advanced Linux Sound Architecture (ALSA), upon which the author's system will be based. The system will be designed to allow the user to specify sampling rate in a configuration file, allowing the system to be adapted in the field. Since higher sampling rates tax the processor and storage space, the lowest sampling rate that will adequately capture the desired signals is preferred. Nyquist's Law dictates that this rate must be at least twice the frequency of the signal's highest noise component.

3. Signal Analysis

After sound samples have been acquired, the system will analyze signals for their constituent frequencies in order to distinguish abnormal sounds. In 1807 Joseph Fourier showed that a signal could be approximated by a sum of Sines and Cosines as illustrated in equation 1. [5]

$$a_0 + \sum_{k=1}^{\infty} \left(a_k \cos kx + b_k \sin kx \right) \tag{1}$$

The Fast Fourier Transform (FFT), which is an efficient adaptation of Fourier's work, is regularly used to analyze sound waves. The FFT transforms a signal from the time-based domain into the frequency-based domain, thereby revealing its key components. The approximation works well when the input signal is continuous and consists of a few well-defined frequencies. An example of the use of FFT is presented in figure 1 and figure 2.

Figure 1 is a plot of an input signal created in Maple using frequencies of 440, 880, 1320, and 1760 Hz, each frequency using amplitude values of 1, 0.75, 0.4, and 0.1 respectively [4]. The FFT transform of the signal in Figure 2 demonstrates that FFT can identify the original signal's constituent frequencies, making Fourier Transform a powerful tool in signal analysis.

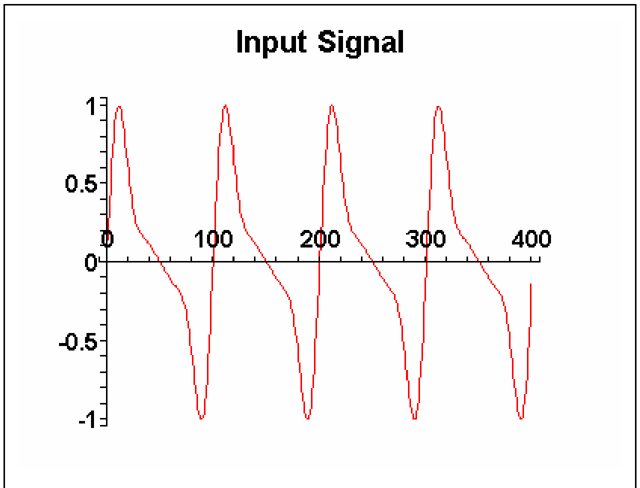

Figure 1

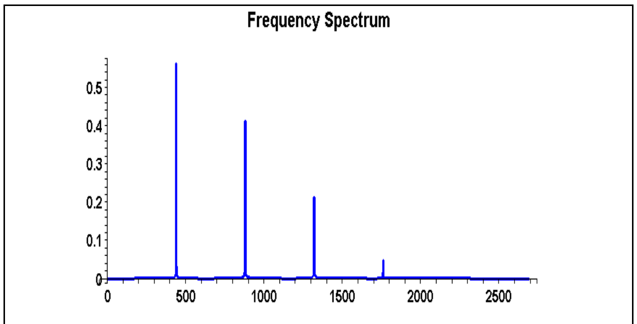

Figure 2

When an input signal contains discontinuous noise like intermittent pulses or spikes, the Fourier Transform cannot resolve the signal into a distinct set of frequencies. The Maple FFT transform presented in Figure 4 shows that FFT cannot resolve the spike shown in Figure 3 into a small set of discrete frequencies. Instead, FFT represents the signal as a sum of all

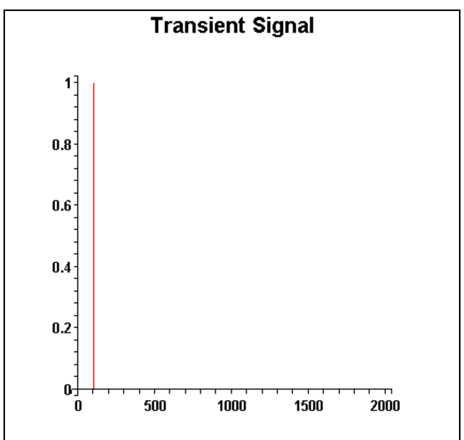

Figure 3

frequencies. Since many computer-produced sounds are intermittent, FFT's inability to characterize an intermittent signal is a serious problem.

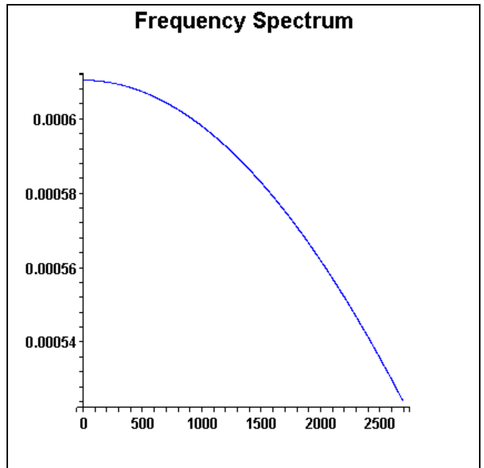

Figure 4

3.1. Wavelets

Wavelets, a related formalism for signal processing, provide much better support for analyzing intermittent signals. Unlike the sine and cosine functions used by the Fourier transform, most wavelet functions use basis functions with non-zero values over a finite range, a characteristic referred to as "compact support." Compact support enables combinations of scaled and translated wavelets to approximate signals that include transient events. A wavelet transform yields a set of coefficients that scale and translate a wavelet function to produce an approximation of an original signal. The process is reversible, allowing reconstruction of the transformed signal. Coefficients may be manipulated to filter or compress the signal. Since these coefficients characterize the signal, they can be used to detect equipment failures.

A family of wavelets proposed by Ingrid Daubechies exhibits compact support and orthogonality, and is useful as a filter and for time localization of signals. This wavelet is defined by a recursive function, $\phi(x) = \sum_{k=-\infty}^{\infty} p_k \phi(2x-k)$. Individual Daubechies wavelets differ in the number of non-zero terms. The Daubechies wavelet of interest here, D4, has four non-zero terms or vanishing moments. D4 is determined by the expansion

$$\phi_n(x) = P_0\phi_{n-1}(2x) + P_1\phi_{n-1}(2x-1) + P_2\phi_{n-1}(2x-2) + P_3\phi_{n-1}(2x-3)$$

where P_0 through P_3 are defined as

$$P_0 = \frac{1+\sqrt{3}}{4} \qquad P_1 = \frac{3+\sqrt{3}}{4}$$

$$P_2 = \frac{3-\sqrt{3}}{4} \qquad P_3 = \frac{1-\sqrt{3}}{4}$$

The initial condition is defined by

$$\phi_0(x) = \begin{Bmatrix} 1 \; if \quad 0 \le x < 1 \\ 0 \; otherwise \end{Bmatrix}$$

and the number of iterations n is equal to $\log_2(x)$, where x is the number of elements in the sample being analyzed. Evaluating the defining equation for D4 over an arbitrary number of iterations produces the scaling function $\phi(x)$ (cf. Figure 5). The mother wavelet function $\psi(x)$ can be computed from the scaling wavelet function $\phi(x)$ using $\psi_k(x) = \sum_{k=1}^{N} (-1)^k_{P_{1-k}} \phi(2x-k)$ [2] (cf. Figure 6).

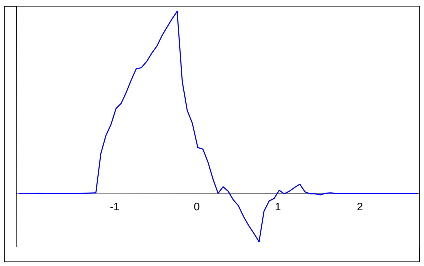

Figure 5 D4 Scaling Function ϕ

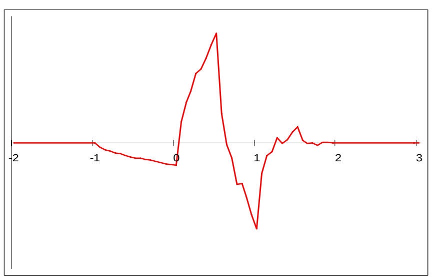

Figure 6 D4 Wavelet ψ

The wavelet function ψ can be used to analyze signals by finding coefficients $C_{n,k}$ satisfying equation 2. [1]

$$f(t) = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} C_{n,k} \psi(2^n t - k) \tag{2}$$

3.2. Multiresolution Analysis

An efficient way of finding wavelet coefficients is Multiresolution Analysis, sometimes referred to as the Fast Wavelet Transform (FWT). A signal sample, which can be referred to as a signal space V_0, can be approximated by a subspace V_j such that $f \in V_j \Leftrightarrow f(2^j) \in V_0$ [3]. V_j is a scaled subspace of the original space V_0, occupying half as many elements but with the requirement that V_j must be contained within the space V_0. The signal V_0 has been downsampled to represent the same signal with fewer samples. The diagram in Figure 7 illustrates one iteration of the process. The signal V_0 is filtered with a low pass filter, then downsampled, reducing the number of data elements by half. This results in a sample that approximates the original signal but with half the number of elements. The signal is also filtered with a high-pass filter and downsampled, yielding a subsample W_1 that represents the high-frequency details that were removed from the original signal by the low-pass filtering. The process may be repeated on the vector V_1 through V_n. At the point of V_n, there remains only a single element representing the average of the original signal sample. The subsample W_1 through W_n represents the coefficients in equation 2.

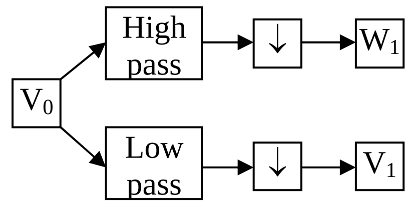

Figure 7 Multiresolution Analysis

Existing code implementing the Daubechies D4 wavelet turned out to be hard to find and a wavelet transform was developed from the Daubechies formulas using C++. The code performs the D4 transform using the same storage that holds the input data; no additional storage is needed. A variation on Multiresolution Analysis allows reuse of storage space using “lifting,” but that technique turned out to be unnecessary [3]. An advantage of Multiresolution Analysis is that it completes in time O(n), a significant improvement over FFT which requires O($n\log_2(n)$).

Figure 8 shows a 3D plot of the D4 wavelet transform, applied to the spike signal in Figure 3. This plot was obtained from a Maple worksheet linked to an external DLL incorporating the D4 C++ code. The axis “Level” indicates the iterations of the Multiresolution analysis where level zero is the first pass and level 10 is the end of the transform. The axis “X” refers to the elements in each iteration. There are 1024 elements in the original signal and only 100 elements are depicted in this plot, with the rest

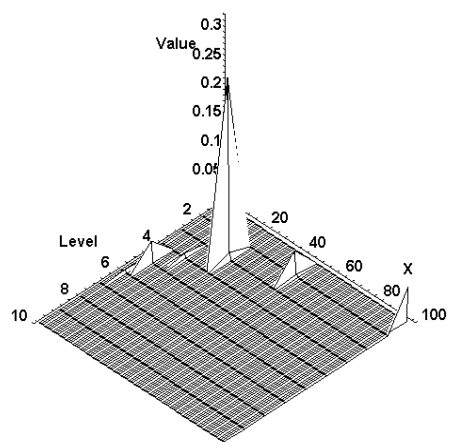

Figure 8

truncated for clarity. The line of spikes extending diagonally across the plot is referred to as a "ridge." Irregular spikes composing the ridge, such as are produced by signals of this type are referred to as "ribs."

4. Analyzing Wavelet Results

When the wavelet transform has produced a set of coefficients, the system must compare these results to a set of known results to determine if the sound signal indicates that a fault exists. Other wavelet analysis systems have used neural nets to analyze the map of wavelet coefficients. [7]

A simpler approach under consideration is to compare the wavelet to a set of known wavelet coefficients using the chi-square test for statistical significance. A chi-square test value exceeding a specified value would indicate a failure condition for the type of failure under test. A series of such tests would be performed for each known type of failure. If none of the comparisons test true then the system may be considered to be operating normally. This approach will require a set of sample data that would be compiled into test sets for comparison.

Upon detecting a positive indication for a failure, the system would enter the event into a log file and send a message to the cluster supervisor or to the system administrator's monitor. The message will allow action to be taken which might consist of removing the failed node from work assignments or dispatching of maintenance personnel.

5. Experimental Design

East Tennessee State University has a program to refurbish used university computers and provide them to area public school systems. Failed components, particularly hard drives, were made available to this project for testing. Sounds from these hard drives are recorded digitally and readings from the hard drives' S.M.A.R.T. system are collected. The S.M.A.R.T. data are correlated with the recordings to verify that the hard drive has failed and to categorize the type of failure.

A tool will be developed to analyze recorded sounds from the hard drives using the D4 wavelet and create a set of wavelet coefficients for analysis. These wavelet coefficients will be compiled into a database that will be used by a Linux daemon that will be developed to monitor a Linux cluster systems. The daemon will compare the wavelet transforms of sounds from the system it is monitoring to the database of transforms.

A remaining problem for the project is choosing a method of classifying the wavelet transforms of the input sounds. A method using chi-square analysis between entries in the database and the sample transform has the advantage of being simple to implement. It will require analysis of test samples for relevant coefficients and compiling them into comparison sets. Coefficients corresponding to frequencies determined to be irrelevant to the type of failure being considered may be ignored to improve performance of the comparison.

An alternate approach using neural networks to analyze the wavelet output has the putative advantage of being able to learn and add to the database through experience. A problem with a neural net in this application is that a failure event triggering a response from the system might be the last act that the system is able to complete. Hence, there will be little advantage to a neural net's ability to learn. Also, the neural net must be trained in order to be put into operation, requiring a large number of training samples.

An additional problem is accounting for individual variations among devices of the same manufacture and type, and variation between similar devices from different manufacturers. It cannot be assumed that all failing hard drives sound the same. A statistical analysis of results must be made to assure validity of the results. A difficulty for this project is that the number of failed components available for test is small and statistical validity may be impossible to attain. Though the objective of the project is to be able to detect a wide range of hardware failures, the focus is primarily on hard drives since there are larger numbers of hard drives available for test and S.M.A.R.T. equipped hard drives provide a simple means of verifying faults. It is hoped that other types of hardware may be tested as well.

Conclusion

This project is intended to develop a system that will use sounds or vibrations coming from computer components to detect failures in computers which normally operate unattended. The use of wavelets allows these sounds to be analyzed for both frequency and transient events, either of which may have significance in detecting abnormal operation. A remaining obstacle is to develop a strategy for classifying the wavelet coefficients. Once completed, the project will provide cluster computer operators with another tool to monitor the condition of the hardware. The project is anticipated to be completed by summer of 2005.

References

[1] Aboufadel, E., *Daubechies Wavelet*¸ Maple worksheet, Waterloo Maple, Inc., 2001

[2] Daubechies, I, *Ten Lectures on Wavelets,* Society for Industrial and Applied Mathematics, 1992

[3] Daubechies, I. and W. Swelden. *Factoring wavelet transforms into lifting steps.* Technical report, Bell Laboratories, 1996.

[4] Goossens, P., *Spectral Analysis using Maple 6,* Maple worksheet, Waterloo Maple Inc., 2000

[5] Graps, A., *An Introduction to Wavelets,* Institute of Electrical and Electronics Engineers, Inc., 1995

[6] Mangeruca, L., Ferrari, A., Sangiovanni-Vincentelli, A., Pierantoni, and A., Pennese, M., *System Level Design of Embedded Controllers: Knock Detection, a Case Study in the Automotive Domain,* 2003

[7] Yen G., and Lin, K., *Wavelet Packet Feature Extraction for Vibration Monitoring,* 2000, IEEE Transactions on Industrial Electronics, Vol, 47, No. 3, 650-667