Available online at www.sciencedirect.com
ScienceDirect
Future Generation Computer Systems 24 (2008) 763–773
FGCS
FUTURE GENERATION COMPUTER SYSTEMS
www.elsevier.com/locate/fgcs

Application of Grid computing for designing a class of optimal periodic nonuniform sampling sequences

Andrzej Tarczynski[a], Tamas Kiss[b,*], Gabor Tersztyanszki[b], Thierry Delaitre[b], Dongdong Qu[a], Stephen Winter[b]

[a] *Centre for System Analysis, University of Westminster, 115 New Cavendish Street, London, W1W 6UW, United Kingdom*
[b] *Centre for Parallel Computing, University of Westminster, 115 New Cavendish Street, London, W1W 6UW, United Kingdom*

Received 1 March 2007; received in revised form 18 December 2007; accepted 7 February 2008
Available online 26 February 2008

Abstract

Designing periodic nonuniform sampling sequences for digital alias free signal processing is a computationally extensive problem where sequential single computer based solutions could easily run for days or even weeks. In order to reduce computation time, the sequential algorithm needed to be parallelized making it possible to execute parts of the calculations on different nodes of a computational Grid at the same time, reducing the overall runtime of the application. This paper presents and compares two different Grid based implementations representing the two main Grid approaches at the moment. The first solution utilizes a production Grid environment based on GEMLCA and the P-Grade Grid portal, and the second represents a BOINC desktop Grid-based solution.

1. Introduction

In traditional digital signal processing (DSP) analysis, the processed signals are sampled at uniformly distributed time instants. While use of such sampling schemes has obvious advantages like cyclostationarity of signal processing systems or facilitating use of efficient processing algorithms e.g. Fast Fourier Transform (FFT), uniform sampling suffers from a well-known limitation that prevents it from use in frequency ranges wider than half of the sampling rate. This limitation, known as aliasing, does not allow telling apart signal's sinusoidal components if the sum or difference of their frequencies is an integer multiple of the sampling frequency. This fact is illustrated in Fig. 1. We show there a sequence of noisy samples of a sinusoidal signal. The samples are taken at 200 MSps rate (sample time is 5 ns). In turn, we attempt to fit sinusoids of frequencies 40 MHz and 160 MHz into the acquired samples. In both cases we use the least squares method to get the best fit. Both sinusoids fit the data equally well. Therefore, it is impossible to tell which of them, if any, has been sampled.

The classical way of preventing aliasing is to establish which frequency ranges could be present in the processed signal. We refer to these ranges as the signal spectral support. The support ought to be conservative, i.e. no frequency present in the signal should be neglected. Once the spectral support is established the sampling rate is chosen in such a way that no two frequencies present in the support could become aliases of each other. If the spectral support is chosen very conservatively, then the resultant sampling rate is high in comparison with the Landau rate [1] — the theoretically lowest sampling rate that allows perfect signal reconstruction. Landau rate equals the double-sided bandwidth of the signal. In some cases, in order to preserve signal reconstructability, sampling at the Landau rate may require taking the samples at nonuniformly distributed time instants. Therefore, a more common approach is to use uniform sampling at the rate that exceeds twice the highest frequency present in the signal (Nyquist rate).

Use of excessive sampling rates (higher than Landau or even the Nyquist rate) is often acceptable, particularly when dealing with low-frequency signals. However, in some cases, e.g. processing of gigahertz signals, it could lead to solutions that are either economically or technically not viable. Digital

* Corresponding author.
E-mail addresses: tarczya@westminster.ac.uk (A. Tarczynski), kisst@wmin.ac.uk (T. Kiss).

0167-739X/$ - see front matter
doi:10.1016/j.future.2008.02.005

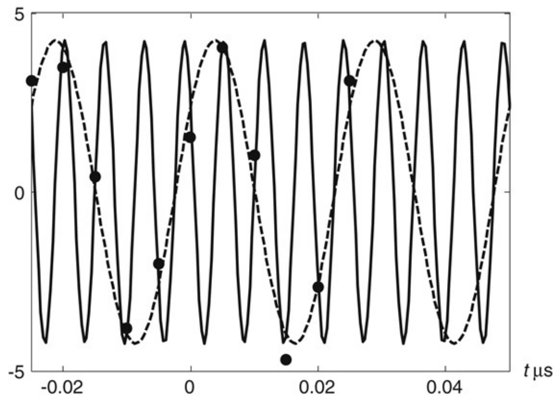

Fig. 1. 160 MHz and 40 MHz sinusoids fit equally well to signal measurements taken at 200 MSps rate.

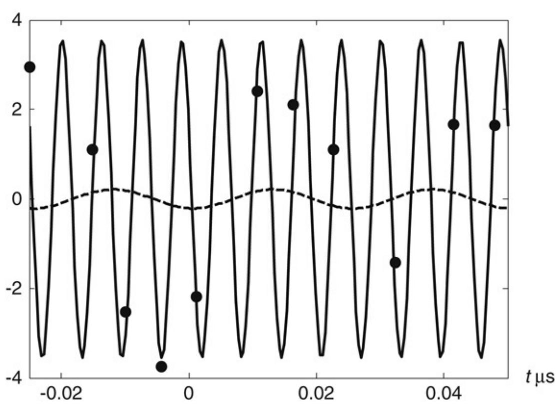

Fig. 2. 160 MHz sinusoid fits nonuniformly sampled data much better than 40 MHz sinusoid.

alias-free signal processing (DASP) is an approach that offers effective solutions to processing signals with conservatively estimated spectral support. It uses carefully designed low-rate nonuniform sampling schemes and appropriate corresponding processing algorithms. To illustrate how nonuniform sampling could be used for suppressing aliasing we repeat the experiment shown in Fig. 1. This time however, we use nonuniform sampling to collect data. The distances between consecutive sampling instants have been selected randomly between 5 ns and 10 ns. The results are shown in Fig. 2. Now it is obvious that the samples have been taken from the 160 MHz sinusoid as it fits the data much better than the 40 MHz sinusoid.

DASP-type solutions reported in research literature rely on either random sampling [2–4] or on periodic nonuniform sampling (PNS) [5–7]. In this paper we revisit the problem of designing PNS sequences for DASP applications that has been originally presented in [7]. This time however, we concentrate on computational aspects of the problem. We show that selection of optimal sampling sequence is a computationally expensive process and solving it with use of a single computer may require long waiting times before the results are produced. In order to make search for optimal solution more practical we took a number of steps to reduce the total computational workload and then implemented the optimization procedure in a parallel environment to speed up the calculations.

The aims of this paper are twofold. First, it illustrates how Grid computing can be applied to speed up computations in the area of digital signal processing. Parallelization and the utilization of distributed computing resources can reduce very lengthy execution times. This allows scientists to run simulations and carry out experiments that are not feasible by single computer implementations. Our solution aims to stand as a reference implementation that can not only solve the particular problem presented in this paper faster, but also inspires scientists in related fields. Second, we introduce the application of some specific Grid computing tools and high level environments like the P-GRADE/GEMLCA portal or the Distributed Computing API. These tools provide the means to port applications onto the Grid with minimal programming effort and without the need to understand low level details of the underlying computing architecture. The aim of the described application porting is not necessarily to achieve the best possible efficiency of the execution, rather to provide a useful and user friendly solution easily and quickly. We do not state that our solutions are optimal concerning execution times. However, the solution still provides significant speed-up and can be easily generated by application programmers without spending months or years studying Grid computing principles and architectures. This easy applicability is one point often overlooked in current Grid solutions.

The rest of this paper is organized as follows. In the next section we give brief description of the design of the PNS method that was originally introduced in [7]. Then we look at the numerical aspects of designing the sequence and show some simulation results. In Section 4 we describe how the optimization procedure has been implemented in a grid environment. The results of testing the new implementation and analysis of how much timing is improved are presented in Section 5. Related solutions are described and analysed in Section 6. Finally, we conclude the paper in Section 7 and discuss briefly further improvements that can be introduced to Grid implementations.

2. PNS for DASP

2.1. General analysis of PNS

PNS is a sampling scheme where the positions of the sampling instants τ_m are distributed periodically. Let T be the period of such a sequence and $\{\tau_1, \ldots, \tau_N\}$ the positions of the sampling instants inside the first period. The remaining sampling instants are: $\tau_m = \tau_{kN+n} = \tau_n + kT$ where k takes whole values and $1 \leq n \leq N$. Note that when $N = 1$ PNS becomes a classical uniform sampling sequence. Let $x(t)$ be a continuous-time signal. Our goal is to design a low-rate PNS sampling sequence such that the spectrum $X_d(f)$ of a resultant discrete-time signal $x_d[m] = x(t_m)$ is as similar as possible to the spectrum $X(f)$ of $x(t)$. It has been shown in [7] that $X_d(f)$ and $X(f)$ are related to each other in the following manner:

$$X_d(f) = \sum_{k=-\infty}^{\infty} c_k X\left(f - \frac{k}{T}\right) \quad (1)$$