MATHEMATICAL MODELLING AND ANALYSIS
VOLUME 12 NUMBER 4, 2007, PAGES 409–418
 ISSN 1392-6292 print, ISSN 1648-3510 online

PARALLELIZATION OF THE α-STABLE MODELLING ALGORITHMS

I. BELOVAS[1,2] and V. STARIKOVIČIUS[2]

[1] *Institute of Informatics and Mathematics,*
Akademijos 4, LT-08663, Vilnius, Lithuania
[2] *Vilnius Gediminas Technical University*
Saulėtekio al. 11, LT-10223, Vilnius, Lithuania
E-mail: igor_belov@takas.lt; vs@sc.vgtu.lt

Received August 8, 2007; revised September 23, 2007; published online December 15, 2007

Abstract. Stable distributions have a wide sphere of application: probability theory, physics, electronics, economics, sociology. Particularly important role they play in financial mathematics, since the classical models of financial market, which are based on the hypothesis of the normality, often become inadequate. However, the practical implementation of stable models is a nontrivial task, because the probability density functions of α-stable distributions have no analytical representations (with a few exceptions). In this work we exploit the parallel computing technologies for acceleration of numerical solution of stable modelling problems. Specifically, we are solving the stable law parameters estimation problem by the maximum likelihood method. If we need to deal with a big number of long financial series, only the means of parallel technologies can allow us to get results in a adequate time. We have distinguished and defined several hierarchical levels of parallelism. We show that coarse-grained Multi-Sets parallelization is very efficient on computer clusters. Fine-grained Maximum Likelihood level is very efficient on shared memory machines with Symmetric multiprocessing and Hyper-threading technologies. Hybrid application, which is utilizing both of those levels, has shown superior performance compared to single level (MS) parallel application on cluster of Pentium 4 HT nodes.

Key words: Parallel algorithms, stable modelling, financial mathematics

1. Introduction

Modelling of financial processes and their analysis is a very fast developing branch of applied mathematics. Originally processes in economics and finance were described by Gaussian models. However, at present normal models are taken with more criticism, because it has been noticed out that they often inadequately describe the behaviour of financial series. The reason is that the real data are usually characterized by skewness, kurtosis and heavy tails. Since the classical Gaussian models eventually have lost their positions, new models were proposed. Stable models attracted special attention [22].

Nowadays stable models have become an extremely powerful tool in mathematical modelling. Stable distributions are used in a wide sphere of application: probability theory, physics, electronics, insurance, economics, computer networking and sociology [4, 5, 23].

Particularly important role they play in financial mathematics. There are several essential reasons why the models with a stable paradigm are applied to model financial processes.

The first one is that stable random variables justify the generalized central limit theorem, which states that stable distributions are the only asymptotic distributions for adequately scaled and centered sums of independent identically distributed random variables [11].

The second one is that stable distributions are heavy-tailed. All but one of stable distributions have infinite variance, which implies that observations of large magnitude can be expected and may, in fact, dominate sums of these random variables. It is not valid to treat these observations as outliers since excluding them takes away much of the significance of the original data; indeed, it is precisely these observations that may be of greatest interest. This led Mandelbrot [14, 15] to suggest the stable laws as possible models for the distribution of income and speculative prices. Take for example the distribution of changes in stock market prices. Mandelbrot [15], Fama [10] and others have shown that the probability of very large deviations is so great, that many statistical techniques which depend for their validity on the asymptotic theory of finite variance distributions are inapplicable. The sum of a large number of these variables is often dominated by one of the summands - a theoretical property of infinite variance distributions. In such a case, a mathematical model assuming such a distribution for the observations is very useful.

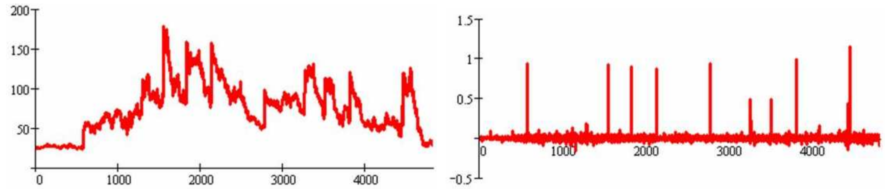
The third one is that stable distributions are asymmetric and leptokurtic. Since stable distributions can accommodate the heavy tails and asymmetry, they give a very good fit to empirical data. In particular, they are valuable models for data sets covering extreme events, like market crashes or natural catastrophes.

The fourth one is that stable distributions are a more flexible tool compared to the normal distribution. The dependency of stable distributions on four parameters makes them flexible to adapt empirical data for calibration and model testing.

Following Rachev [6, 12] "the α-stable distribution offers a reasonable improvement if not the best choice among the alternative distributions that have been proposed in the literature over the past four decades".

Figure 1 presents diagrams of Microsoft company stock prices and Microsoft company stock daily returns. Empirical data, as we can see, features heavy tales and strong asymmetry: large observations are one sided. Chart behaviour corresponds with stable paradigm well.

However, the probability density functions of α-stable distributions have no analytical representations (with a few exceptions). This makes the prac-



a) stock prices b) returns

Figure 1. Empirical data: a) Microsoft company stock prices, b) Microsoft company stock daily returns.

tical implementation of α-stable distributions in various applications a non-trivial task and the whole α-stable modelling – computationally demanding. Especially, when we need to handle long sets of data, which are common, for example, in financial analysis, or a big number of such sets.

Parallel computing is successfully applied in many areas [8]. Our goal is to apply it for the numerical solution of α-stable modelling problems. The objective of this paper is parallelization of the solution of the stable law parameters estimation problem. We are defining several hierarchical levels of parallelism. Several coarse- and fine-grained data-parallel algorithms are proposed. Their performance is investigated on parallel computers with distributed and shared memory.

2. Stable Distributions

The stable distribution can be most conveniently described by its characteristic function

$$\log \varphi(t) = \begin{cases} -\sigma^\alpha |t|^\alpha \{1 - i\beta \operatorname{sign}(t) \tan \frac{\pi\alpha}{2}\} + i\mu t, & \alpha \neq 1, \\ -\sigma |t| \{1 + i\beta \operatorname{sign}(t) \frac{2}{\pi} \log |t|\} + i\mu t, & \alpha = 1, \end{cases} \tag{2.1}$$

where $\alpha \in (0, 2]$, $\beta \in [-1, 1]$, $\sigma > 0$, $\mu \in \mathbb{R}$. Here α is the characteristic exponent (the index of stability), β is the skewness, μ is the location parameter and σ is the scale parameter. A stable probability density function (PDF) is symmetrical if and only if $\beta = 0$. When $\sigma = 1$ and $\mu = 0$ the distribution is called standard stable. The general PDF of the stable distribution can be standardized such that

$$p(x, \alpha, \beta, \mu, \sigma) = \frac{1}{\sigma} p\Big(\frac{x - \mu}{\sigma}, \alpha, \beta, 0, 1\Big) = \frac{1}{\sigma} p\Big(\frac{x - \mu}{\sigma}, \alpha, \beta\Big).$$

The canonical representation (2.1) has one serious disadvantage. The functions $\varphi(t)$ have discontinuities at all points of the form $a = 1, \beta \neq 0$. Therefore for numerical purposes it is advisable to use Nolan's [18] parametrization

$$\log \varphi_0(t) = \begin{cases} -\sigma^\alpha |t|^\alpha (1 + i\beta \operatorname{sign}(t) \tan \frac{\pi\alpha}{2}((\sigma|t|)^{1-\alpha} - 1)) + i\mu_0 t, & \alpha \neq 1, \\ \sigma |t| (1 + i\beta \operatorname{sign}(t) \frac{2}{\pi} \log(\sigma|t|)) + i\mu_0 t, & \alpha = 1. \end{cases}$$

This parametrization is a variant of Zolotarev's (M) parametrization [23], with the characteristic function and hence the density and the distribution function jointly continuous in all four parameters. The location parameters of the two representations are related by

$$\mu_0 = \begin{cases} \mu + \beta\sigma \tan\frac{\pi\alpha}{2}, & \alpha \neq 1, \\ \mu + \beta\sigma\frac{2}{\pi}\ln\sigma, & \alpha = 1. \end{cases}$$

PDF of the two representations are related by

$$p(x,\alpha,\beta,\mu,\sigma) = \frac{1}{\sigma}p_0\Big(\frac{x-\mu_0}{\sigma},\alpha,\beta\Big),$$

here $p_0(x,\alpha,\beta)$ is Nolan's standard stable PDF

$$p_0(x,\alpha,\beta) = \frac{1}{\pi}\int_0^\infty \exp(-t^\alpha)\cos(h(x,t;\alpha,\beta))\,dt, \tag{2.2}$$

where

$$h(x,t;\alpha,\beta) = \begin{cases} xt + \beta\tan\frac{\pi\alpha}{2}(t-t^\alpha), & \alpha \neq 1, \\ \mu + \beta\sigma\frac{2}{\pi}\ln\sigma, & \alpha = 1. \end{cases}$$

3. PDF Calculation

Without analytical representation of PDF (with a few exceptions: Gaussian distribution, Cauchy distribution, Lévy distributions) the practical implementation of stable models is a nontrivial task. There is a number of numerical methods that have been found useful in practice (direct numerical integration methods [9, 13, 19], Fast Fourier Transform method [16], polynomial-based approximation method [7], method of two quadratures [2]). However, the most simple way to evaluate stable PDF (in the case of $\alpha > 1$, as it is usually assumed in financial mathematics) is to replace the improper integral in (2.2) with an approximation

$$p_0(x,\alpha,\beta) \approx \frac{1}{\pi}\int_0^\Delta \exp(-t^\alpha)\cos(xt + \beta(t-t^\alpha)\tan\frac{\pi\alpha}{2})\,dt. \tag{3.1}$$

Here $\Delta = -\ln\pi\varepsilon$, while the error of (3.1) is not greater than ε. Now we evaluate the integral in (3.1) via Gauss-Kronrod quadratures with an accuracy ε. The total accuracy of the evaluation does not exceed 2ε. We used this approach for its simplicity, however the method of two quadratures [2] is expected to be faster.

4. Estimation of Parameters of Stable Models

The most precise (and most time consuming) method of estimation of stable parameters is Maximum Likelihood (ML) method [3]. The parameters

$(\alpha, \beta, \mu, \sigma)$ can be estimated from the returns $x_1, x_2, \ldots, x_n$ by maximizing the log-likelihood function

$$L(\alpha, \beta, \mu, \sigma) = \sum_{k=1}^{n} \ln p(x_k, \alpha, \beta, \mu, \sigma). \tag{4.1}$$

Empirical studies show that the log-likelihood target function has uniextremal nature, often with very flat surface in the neighborhood of the extremum. Figure 2 shows 3D cuts of the target function, obtained by fixing pairs of parameters. The target function was calculated with the sample of returns of size 5000 generated with the following parameters:

$$\alpha = 1.5,\ \beta = 0.5,\ \mu = 0,\ \sigma = 0.1. \tag{4.2}$$

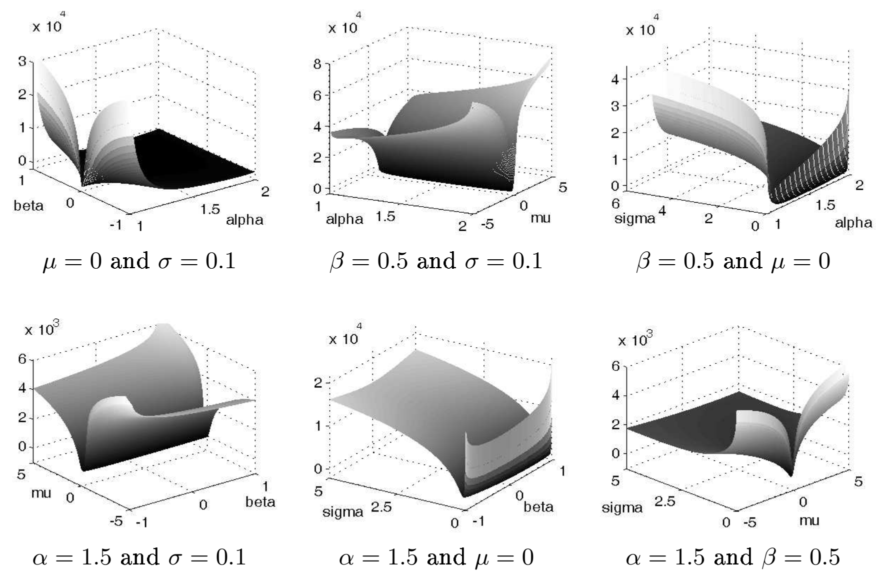


Figure 2. 3D cuts of ML target function (4.1).

To optimize the log-likelihood function we use the Nelder-Mead simplex method. Although this method is not the fastest, it does not require any derivative (gradient, Hessian) calculation.

In Table 1 we show the results of ML optimization for the data sets of increasing size. Each data set was generated with the same set of parameters (4.2), which is a reference solution in our numerical experiments. The PDF integral (3.1) is computed with the accuracy 10^{-11}. In the simplex method, we take $\varepsilon = 10^{-5}$. Further increase of the accuracy does not affect the results. Computations and time measurements were performed on Intel Pentium 4 processor with 3.2 GHz.

Table 1. Results of ML optimization for the data sets of increasing size.

Set size	Time [s]	Iter.	$\hat{\alpha}$	Δ_α	$\hat{\beta}$	Δ_β	$\hat{\mu}$	Δ_μ	$\hat{\sigma}$	Δ_σ
$5 \cdot 10^2$	14.74	158	1.6039	0.1039	0.8415	0.3415	0.0120	0.0120	0.1082	0.0082
10^3	25.12	139	1.5241	0.0241	0.5950	0.0950	0.0082	0.0082	0.1044	0.0044
$5 \cdot 10^3$	156.37	167	1.4667	0.0333	0.5375	0.0375	0.0094	0.0094	0.1013	0.0013
10^4	254.18	143	1.4961	0.0039	0.5171	0.0171	0.0021	0.0021	0.1012	0.0012
$5 \cdot 10^4$	1384.2	149	1.4900	0.0100	0.5200	0.0200	0.0026	0.0026	0.1000	0.0000
10^5	2614.0	148	1.4975	0.0025	0.5204	0.0204	0.0017	0.0017	0.1000	0.0000

5. Parallelization

Numerical computations with computationally demanding ML-method combined with nontrivial numerical calculation of stable PDF are very time consuming. Especially, when we need to handle long sets of data, which are common in financial analysis. Thus only parallel technologies allow us to get results in an adequate time.

For this problem we distinguish the following levels of possible parallelization (in hierarchical order):

- **Multi-Sets (MS) level**. Concurrent solution of big number of independent ML optimization tasks for multiple sets of returns.
- **Optimization level**. Parallelization of optimization method.
- **Maximum Likelihood (ML) level**. Parallel computation of ML target function (4.1) for long data sets.
- **PDF level**. Parallel computation of PDF integral (3.1).

In the current work we deal with MS and ML levels of parallelization.

In MS parallelization, first we uniformly distribute the data (sets of returns) among available processors. Next, each processor consequently solves the independent ML optimization tasks for each set of returns it has received. And finally, we collect and proceed the obtained results.

In Table 2 we present results on the performance and scalability of MS parallelization, which was done with MPI [17], for the numerical test with 100 sets of 1000 returns. Each set was generated using a unique seed with parameters (4.2). The computations (as well as all others in this paper) were performed on VGTU PC cluster http://vilkas.vgtu.lt. As can be expected, the results show almost perfect parallelization. Practical problems, which can be solved with this algorithm, deal with portfolios of stocks. We note, that MS algorithm can be easily implemented using various parallel templates, for example, Master-slaves templates [1]. It also suits well for computational grids.

ML level represents fine-grained parallelism in our problem. For each call of ML target function (4.1) summands can be distributed among processors and computed in parallel. Table 3 shows the performance results of OpenMP [20]

Table 2. Performance and scalability of Multi-Sets (MS) parallelization.

	1 proc	2 proc	4 proc	5 proc	10 proc
Time, T_p [s]	2547.9	1292.5	650.13	527.04	267.13
Speedup, S_p	1	1.97	3.92	4.83	9.54
Efficiency, E_p	100%	98.5%	98.0%	96.7%	95.4%

implementation. We have conducted the same numerical experiments as in Section 4 with the generated data sets of increasing size. Calculations were performed on SMP (Symmetric multiprocessing) node with two Pentium III Tualatin 1.4 GHz processors and Pentium 4 node (Prescott 3.2 GHz) with enabled Hyper-threading (1 physical processor is seen by operating system as 2 logical processors).

Table 3. Performance of ML parallelization with OpenMP.

Set	Pentium III SMP			Pentium 4 HT		
size	1 proc	2 proc	Efficiency	1 proc	2 proc	Efficiency
10^3	63.44	31.09	102.0 %	25.12	16.61	75.62 %
10^4	647.05	324.65	99.7 %	254.18	168.9	75.25 %
10^5	6533.0	3253.8	100.4 %	2614.0	1735.5	75.31 %

The results show that ML parallelization is very efficient on shared memory machines. They also indicate that almost all computational work is done in the calculation of the target function – all other operations in the optimization method are relatively very cheap. Especially impressive are the results on Pentium 4 node. We get 1.5 speedup on a single physical processor. This proves that the use of Hyper-threading technology can bring real benefits to some applications. We note that there is no degradation of the results when the size of the set is changing in the reasonable range.

Next, we decided to utilize at once both levels of parallelism, namely, Multi-Sets (MS) and Maximum Likelihood (ML). To obtain the parallel application, we used hybrid programming [21] with MS parallelization implemented with MPI and ML with OpenMP, accordingly. The code was tested on two different SMP clusters (hybrid architecture). The first cluster is made from the Pentium III SMP nodes, the second cluster uses Pentium 4 nodes with Hyper-Threading technology enabled. For the details look at http://vilkas.vgtu.lt. In both cases at each single node we can run two threads or processes which share the node's memory.

Table 4 presents the performance results of our hybrid application. We are solving already described test problem with 100 sets of 1000 returns. Reference times with 1 processor are 6400 seconds on Pentium III node and 2548 seconds on Pentium 4 node. In the same table we present the results of application with MS parallelization only, solving the same problem on the same physical resources, running 2 MPI processes per node.

Table 4. Performance of hybrid parallelization (MS+ML) vs MS parallelization.

	cluster of Pentium III SMP nodes						cluster of Pentium 4 HT nodes					
Procs	MS+ML			MS			MS+ML			MS		
grid	T_p	S_p	E_p	T_p	S_p	E_p	T_p	S_p	E_p	T_p	S_p	E_p
1x2	3212	1.99	1.00	3250	1.97	0.99	1694	1.50	0.75	2286	1.11	0.56
2x2	1628	3.93	0.98	1633	3.92	0.98	859	2.97	0.74	1158	2.20	0.55
4x2	818	7.83	0.98	863	7.42	0.93	432	5.90	0.74	604	4.22	0.53
5x2	664	9.65	0.97	668	9.59	0.96	351	7.27	0.73	475	5.37	0.54
10x2	334	19.15	0.96	338	18.96	0.95	177	14.42	0.72	251	10.14	0.51

Comparison of those two applications on our clusters gives us interesting observations. While on cluster with Pentium III SMP nodes the hybrid application brings no substantial benefits in speedup, on cluster with Pentium 4 HT nodes we see real speedup gains using the hybrid approach. Obviously, the use of additional ML parallelization level allows the scheduler to utilize the resources of Pentium 4 processor better than simple spawning of additional MPI process. The characteristics of Pentium III SMP nodes allow to organize efficiently the data flow to processing units for both applications.

Note that the hybrid application distributes the workload in a more flexible way. See the case 4x2 in Tab. 4, when 100 is not divisible by 8.

6. Conclusions

Stable modelling (and particularly stable optimization) is very complex and computationally demanding problem (see Fig. 2 and Tab. 1), which can greatly benefit from parallelization.

We have distinguished and defined several hierarchical levels of parallelism. Tab. 2 shows that coarse-grained Multi-Sets parallelization is very efficient on computer clusters. This level of parallelization should suit very well for computational grids also.

Fine-grained ML level is very efficient on shared memory machines with Symmetric multiprocessing and Hyper-threading technologies (Tab. 3). We expect good performance on multicore processors also.

Finally, we have implemented hybrid application, which is utilizing both of those levels of parallelism (MS+ML). On cluster of Pentium 4 HT nodes, this application has shown superior performance compared to single level (MS) parallel application (Tab. 4). It will be interesting to see, how effective hybrid application will be on the clusters of nodes with multicore processors.

Acknowledgment

This work was supported by the Lithuanian State Science and Studies Foundation within the project on B-03/2007 "Global optimization of complex systems using high performance computing and GRID technologies".

References

[1] M. Baravykaitė and R. Šablinskas. The template programming of parallel algorithms. *Mathematical modelling and analysis*, **7**(1), 11–20, 2002.

[2] I. Belov. On the computation of the probability density function of a-stable distributions. In: *Mathematical modelling and analysis. Proceedings of the 10th International Conference MMA 2005*. Technika, 333–341, 2005.

[3] I. Belovas, A. Kabašinskas and L. Sakalauskas. Study of stable models of equity markets. In: *Informational technologies 2005. Proceedings of the conference*, 2, Technologija, Kaunas, 439–462, 2005. (in Lithuanian)

[4] I. Belovas, A. Kabašinskas and L. Sakalauskas. On covariance and codifference in optimal portfolio construction. *Information sciences*, 42–43, 182–188, 2007. (in Lithuanian)

[5] I. Belovas, A. Kabašinskas and L. Sakalauskas. On the modelling of stagnation intervals in emerging stock markets. In: *Computer Data Analysis and Modeling: Complex Stochastic Data and Systems: Proceedings of the Eighth International Conference, Minsk, Sept. 11–15, 2007*, 2, Publ. center BSU, Minsk, 52–55, 2007.

[6] M. Bertocchi, R. Giacometti, S. Ortobelli and S. Rachev. The impact of different distributional hypothesis on returns in asset allocation. *Finance Letters*, **3**(1), 17–27, 2005.

[7] T. Doganoglu and S. Mittnik. An approximation procedure for asymmetric stable paretian densities. *Computational Statistics*, **13**(4), 463–475, 1998.

[8] J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon and A. White(Eds.). *The Sourcebook of Parallel Computing*. The Morgan Kaufmann Series in Computer Architecture and Design. Elsevier, November 2002.

[9] W.H. DuMouchel. *Stable distributions in statistical inference*. Unpublished Ph.D. Thesis, Department of statistics, Yale University, 1971.

[10] E. Fama. The behavior of stock market prices. *The Journal of Business of the University of Chicago*, **38**, 34–105, 1965.

[11] B.V. Gnedenko and A. N. Kolmogorov. *Limit Distributions for Sums of Independent Random Variables*. Addison-Wesley Publishing Co., Reading, Mass., 1954, revised 1968.

[12] M. Hoechstoetter, S. Rachev and F.J. Fabozzi. Distributional analysis of the stocks comprising the DAX 30. *Probability and Mathematical Statistics*, **25**(2), 363–383, 2005.

[13] D.R. Holt and E.L. Crow. Tables and graphs of the stable probability density functions. *Journal of research of the National Bureau of Standards. B. Mathematical Sciences*, **77B**(3–4), 143–198, July-December, 1973.

[14] B. Mandelbrot. The Pareto-Lévy law and the distribution of income. *International Economic Revue*, **1**, 79–106, 1960.

[15] B. Mandelbrot. The variation of certain speculative prices. *The Journal of Business of the University of Chicago*, **36**, 394–419, 1963.

[16] S. Mittnik, T. Doganoglu and D. Chenyao. Computing the probability density function of the stable paretian distribution. *Mathematical and Computer Modelling*, **29**, 235–240, 1999.

[17] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard.* www.mpi-forum.org, Version 1.1, 1995.

[18] J.P. Nolan. Numerical calculation of stable densities and distribution functions. *Communications is Statistics-Stochastic Models*, **13**, 759–774, 1997.

[19] J.P. Nolan. An algorithm for evaluating stable densities in Zolotarev's (M) parametrization. *Mathematical and Computer Modelling*, **29**, 229–233, 1999.

[20] OpenMP Group. www.openmp.org.

[21] Rolf Rabenseifner. Hybrid parallel programming: Performance problems and chances. Web page: citeseer.ist.psu.edu/rabenseifner03hybrid.html

[22] S. Rachev and S. Mittnik. *Stable Paretian Models in Finance.* John Wiley and Sons, New York, 2002.

[23] V.M. Zolotarev. *One-Dimensional Stable Distributions.* Amer. Mathematical Society, 1986.