In this issue

Surviving feast and famine

For growth and reproduction, all living organisms need to get nutrients from their surrounding environment. The enormous diversity of life, ranging from bacteria to plants and animals, is a reflection of the various strategies and mechanisms adopted by the living systems to obtain enough nutrients for the continuity of their race. While animals, and to a limited extent plants, have some means of actively finding and gaining access to richer sources of nutrients, most of the bacteria can only be passively transported to locations where the nutrients are in plenty. Once there, the bacterial species feast and proliferate very actively, resulting in a logarithmic growth which, in turn, results in a depletion of the local nutrient supply. The bacteria now starve and wait till the nutrient supply improves. Thus the life cycle of most bacterial species alternates between growth phase ('feast') and stationary or starvation phase ('famine'). These changes obviously involve dramatic alterations in metabolism and physiology of the organism. While some bacterial species tide over the 'famine' or other harsh conditions by changing into resistant forms like spores, others initiate a genetic programme that alters their physiology to withstand starvation and other stresses.

T. K. Mukherjee *et al* (**page 684**) review one of the strategies that many bacterial species employ to regulate their metabolic activities under starvation. This genetic programme, commonly termed as the 'stringent response' involves synthesis and accumulation of guanosine 3′-5′-biphosphate (ppGpp) which serves as an emergency 'brake' on the protein synthesizing machinery when the substrates for protein synthesis are in short supply. It is interesting that ppGpp is at the centre stage in modulating responses to a variety of stresses like oxidative, temperature, light, pH, etc. In eukaryotes also, ppGpp molecules have roles in regulating metabolic activity. Furthermore, like many of the stress proteins, ppGpp has important functions under 'normal' physiological conditions as well. This is typical of the biological system where the same molecular processes are efficiently utilized under very different conditions and with very different consequences in a context-dependent manner.

S. C. Lakhotia

Engineering crops for abiotic stress

Plants have always offered unique opportunities to understand the mechanisms of cellular behaviour related to adaptation. Based on the genetic make-up, and interaction with the environment, each species has found for itself an ecological niche to grow. There are a number of species which can survive under extreme climatic and soil conditions. However, crop plants, which have been domesticated for centuries, when faced with extreme stress environment are unable to survive. It has been an effort of plant biologists to understand why some species or even varieties can tolerate biotic and abiotic stresses, whereas others succumb to the same conditions. Once this is understood it is the dream of biotechnologists to transfer such traits to sensitive cultivars. During the last decade a number of proteins and the genes coding for stress associated characters have been identified. Attempts are now being made to transfer these genes to the plants of interest. On **page 689**, Grover *et al.* discuss the basic issues related to stress biology and the potential and limitations of engineering crops for abiotic stresses. It is well conceived that tolerance to stress is an end result of the regulation of a number of genes. It is therefore a formidable task to pyramid these genes in stress-sensitive crops for developing resistance, and without putting a penalty on other characters and yield potential. An alternate pathway, as was discussed in the Research News section of *Current Science*, 1998, **75**, 178–179 by the same group, would be to develop transgenic crops overexpressing transcription factors that regulate the expression of a battery of stress-related genes. Certainly this area of plant stress biology has a lot of excitement for fundamental researchers and potential for agro-biotechnologists.

S. K. Sopory

A small step in parallelizing – and a giant stride in performance: Another *tour de force* from the Flowsolver team

Few activities can be as exhilarating as making things go faster. For a street-child rolling a wheel along the road or for the pilot of the latest F-16, for Venkatesh Prasad sending a bouncer or for Goran Ivansenivick serving an ace – faster is not only better, but more enjoyable too. Computer scientists are unlikely to be exceptions, and developing faster computers (or more efficient programs) is taken up more for the innate joy of it. Of course, the cover story that all this is being done so as to predict the weather, which will help the hard-working farmer to produce a good harvest, and thereby guarantee food for all by the time we reach the next millennium, is very convincing too. Predicting weather ahead of time in a scientific manner is probably one of the most challenging computational tasks – meteorologists, unlike the economists, are always forced to take a global view. To find out if it would rain here a few days from now, one needs to know the pressure, temperature, wind velocities and moisture all over the world, and that too at many different levels in the atmosphere. Add to it effects due to the rotation of the earth, the heating from the sun, the turbulence of the air and the movement of the clouds, and the laws of physics will help one to compute what will happen during the next instant of time. The highly complex collection of computer programs which do this are called Global Climatic Models – compute away over many time steps to predict what will happen at any time in the future. The only problem is that one needs to carry out a lot of calculations. Not so long ago, for predicting weather one day ahead of time, even the fastest computers in the world used to take several weeks of computer time. The speed of a single processor was nowhere near what was needed for making a worthwhile weather forecast.

This is where parallel computation works at its best. The main program distributes the calculations to be done across to many processors (for example, each of the twelve processors may compute the calculations for one twelfth of the globe), who complete the tasks and return the results, which are then put together. Thus, a substantial part of the computation is done simultaneously (in a parallel manner), and the speed is correspondingly higher. The Global Climatic Models are very well suited for such parallelized operation, and there are many successful implementations. In fact, the Department of Science and Technology of the Government of India, during 1993–96, had launched an initiative for developing such a parallelized model, and several groups, including the FLOWSOLVER, developed by U. N. Sinha and colleagues (see **page 709**), had successfully completed the project.

How fast can a paralleized program run, if it distributes the tasks to many processors? Can ten processors make it run ten times as fast? Even an idealized answer is not simple. Not everything in the program can be parallelized, and a small fraction of it has to be run on a single processor. It is this fraction which has a critical role in deciding the speedup. A program with 10% of such code can be made to run at most ten times faster (and that too, if there are infinite processors); one with 5% can theoretically achieve a maximum twenty-fold speedup.

Even more importantly (as explained by T. N. Venkatesh *et al.* **page 709**), 'serial fraction' also controls the 'scalability', that is, the extent to which every additional processor contribute to the speedup. A serial fraction of 10% implies a speedup of about 6.4 with 16 processors, which increases to 9.1 for a serial fraction of 5% and up to 13.9 when the serial fraction reduces to 1%. An apparently small reduction in serial fraction of the code from 4% to 1% thus leads to a substantial speed-up, by making the program more scalable.

The earlier efforts of the flowsolver team had parallelized to the extent of 95.26%; that is, the serial fraction was only 4.74%. The article by T. N. Venkatesh *et al.* gives an insightful description of why it is critical to reduce the serial fraction even more. The very important contribution of this group has been to reduce the serial fraction to a very low value of 0.34%. With a sixty-four-processor machine, while the earlier code could theoretically achieve a maximum of sixteen-fold speedup, their new, optimized code could be close to a fifty-fold speedup! The authors have been quick to point, however, that this extreme parallelization does impose some form of performance penalty – the communication overhead. For a 'coalition' of processors to 'function', communication amongst the 'constituents' is very important – and for extreme parallelization, more and more communication is needed. In fact, there comes a stage when more time is spent on the communication between the processors than on computations themselves. The authors have very ingeniously cut down the communication time by making use of the recently developed 'shared memory' architecture.

If language is the means of concealing thoughts, this article certainly qualifies as means for concealing accomplishments. A bland one liner 'the program was modified . . .' gives little indication to the uninitiated of the great deal of careful intellectual effort that has gone into this exercise (but then again, the paper is not exactly targeted to the uninitiated either). For obtaining the crucial insight that parallelization of the linear part of the Global Climatic Model code is essential, for successfully developing a modified code, and for making it work efficiently on a multiprocessor machine requires formidable expertise in meteorology and atmospheric physics, numerical analysis, software, hardware and systems programming. The team of four authors of the article accounts for at least seven experts between them, and I can do no better than wholeheartedly agree with their exuberant conclusion – 'the pursuit of this approach could well trigger a new chapter in parallel computing for meteorological applications in India'. Amen.

N. V. Joshi