UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Parallel complex systems simulation

Schoneveld, A.

Publication date
1999

Link to publication

Citation for published version (APA):
Schoneveld, A. (1999). *Parallel complex systems simulation*. [Thesis, fully internal, vg logic/info]. ASCI.

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:16 Feb 2026

Chapter 6

Asynchronous Cellular Automata

"Who controls the past controls the future. Who controls the present controls the past."

–George Orwell

6.1 Introduction

A computer simulation must always be accompanied by some updating schedule, which is often not based on the natural phenomenon being simulated, but just assumed or imposed by the modeler.
The danger of simply imposing an update scheme has been perfectly put into word by Bernardo A. Huberman:

"The prisoner's dilemma has long been considered the paradigm for studying the emergence of cooperation among selfish individuals. Because of its importance, it has been studied through computer experiments as well as in the laboratory and by analytical means. However, there are important differences between the way a system composed of many interacting elements is simulated by a digital machine and the manner in which it behaves when studied in real experiments. In some instances, these disparities can be marked enough so as to cast doubt on the implications of cellular automata type simulations for the study of cooperation in social systems. In particular, if such a simulation imposes space-time granularity, then its ability to describe the real world may be compromised." [75]

Huberman used a spatial version of the prisoner's dilemma to show the remarkable differences between asynchronous and synchronous updating [75]. For synchronous dynamics, complex patterns emerge, but for asynchronous updating the number of cooperating players reaches a steady state for arbitrary initial conditions. In synchronous computations nothing happens for times shorter than integral values of unit time. The resulting global dynamics is mathematically described by a finite difference equation. However, in natural social systems, a global clock seldom exists. In for example biological organisms and

most social settings, players, agents, or organisms act at different and uncorrelated times on the basis of information that may be imperfect and delayed. The resulting global dynamics is usually expressed in the form of differential equations, whose solutions are not always the same as those of their finite difference counterparts. Bersini et al. have studied the difference in the dynamics between two CAs, the Game Of Life (GOL) and an Immune Network Model (IMN) [14]. The main difference between the two is that GOL uses synchronous and IMN uses asynchronous updating, yet their dynamical behavior is entirely different. IMN reaches a steady state, while GOL never stabilizes, for most initial conditions. Complementary to the previous result, it has been shown in a recent paper by Blok and Bergersen that, as the amount of synchronicity in GOL is increased, the dynamical behavior undergoes a critical phase transition [16].

In the previous chapters of this thesis only synchronous execution has been applied to all simulation models. The P-CAM framework has been defined as a topological framework for supporting parallel simulations of complex systems. It supports the spatial decomposition of a parallel computation and offers related functionalities like boundary cell administration and cell migration. Similar to space also time can be decomposed in a variety of ways (see Fig. 2.2). In this chapter we will discuss some methods for the so called temporal decomposition (execution mechanism) of cellular automata. The main focus is on the (speculative) critical behavior in the dynamics of a specific method for parallel asynchronous execution: *Parallel Discrete Event Simulation* (PDES) using *Time Warp*.

In this chapter we will shortly discuss various CA update schemes and the parallelization of this schemes. Implementation aspects of merging a specific PDES environment, APSIS (developed by Benno Overeinder [118]) with P-CAM are not considered in chapter. Among many other things, this task is left as a part of future work. But, it should be yet another step in the definition of a full blown generic parallel complex systems simulation environment.

The complex dynamics of Time Warp (TW) will be described. A highly speculative conjecture is made, that the TW dynamics can be characterized by *Self Organized Criticality* (SOC). SOC behavior is found in many complex systems, for example in sand-pile and earthquake dynamics. If this conjecture is true, yet another hint is given that also problems in the field of parallel computing display behavior as found in other complex systems.

6.2 Cellular Automata Update Schemes

An important characteristic of the CA paradigm, is the particular update scheme that applies the microscopic rules iteratively to the cells in the discretized universe. The sequential, synchronous and asynchronous update schemes impose different dynamical behavior. In the following sections five different update schemes are discussed.

The *synchronous (or parallel) update scheme* is the conventional update proce-

dure for cellular automaton models [166]. In a synchronous update scheme, every cell is updated at the exact same simulation time (concurrently).
The *sequential update scheme* is a special kind of asynchronous updating. In the sequential update scheme, a two dimensional lattice is traversed in for example a row-wise manner: starting with the top-left cell and ending with the right-bottom cell. The sequential update sequence may introduce a bias in the dynamical evolution lattice configurations.
Another special case of asynchronous updating is the *semi-random update scheme*. In one sweep through the lattice we order all cells into a random order and evaluate them in that order. Every cell is evaluated exactly once in one sweep. From sweep to sweep the order of evaluation is changed.
The *random update scheme* implies completely asynchronous dynamics. In the random update scheme the next cell to be evaluated is selected according to a uniform distribution.
The *discrete event update scheme* is a generalization of the random update scheme. With each update (now called an event) a simulation time is associated. The discrete event formulation can mimic the random update rules, but due to its expressiveness more complicated models, i.e., more complex microscopic rules, can be simulated. In particular, the explicit representation of simulation time enables a more meaningful interpretation to evolution time of the simulation models.

6.3 Parallel Update Schemes for Asynchronous CA Models

Complex CA simulations of large problem sizes, both in their spatial and temporal domain, are known to consume enormous amounts of time on sequential machines. One basic approach to reduce the required simulation time is the exploitation of parallelism. For simple synchronous CA, a simple lock-step model is sufficient to ensure a correct parallelization: update all cells simultaneously, followed by communicating boundary cells to neighboring processor domains.
By the locality (neighborhood) of the CA update rules, the parallelization of asynchronous CA models can also be realized by a domain decomposition. However, parallelization is complicated due to causality constraints between subdomains.
Although fully asynchronous CA models appear to be inherently sequential as a result of the random update scheme, they can be parallelized by adopting the *discrete event* formulation. However, a major drawback of the parallelization of a discrete event simulation is the inherent update complexity of this type of simulation, since the notion of global time does not easily map on a parallel computer. Sophisticated clock synchronization algorithms are required to ensure that cause-and-effect relationships are correctly reproduced by the asynchronous CA simulation.
The discrete event CA model can be described as follows: instead of randomly

choosing individuals for updating, a time stamp is assigned to each cell present in the current lattice configuration. This time stamp is taken from for example a Poisson stream with arrival rate λ. In addition, a *simulation time* is updated, while jumping from the current event to the next, based on the associated time stamp. The processing of an event is done by applying just the same update rules as with the random update scheme. The parallel discrete event simulation exploits the implicit spatial locality present in the update scheme, without altering the update history [94].

In a sequential discrete event simulation, the execution routine selects the smallest time stamped event from the event list as the one to be processed next. If the update execution departs from this rule, then it would allow for a simulation where the future can affect the past. Errors of this kind are called *causality errors*. A discrete event simulation consisting of processes that interact exclusively by exchanging time stamped messages (event messages), are locally causal *if and only if* each process executes events in a non-decreasing time stamp order. It is this local causality constraint that a parallel discrete event simulation strategy must guarantee.

Globally, Parallel Discrete Event Simulation strategies can be classified in two categories [57]: *conservative* and *optimistic*. Conservative approaches strictly avoid the possibility of any causality error ever occurring, and optimistic approaches use a detection and recovery approach.

6.3.1 Conservative Parallel Discrete Event Simulation

The basic problem that a conservative parallel discrete event simulation must address is to determine which event is allowed to proceed in time. By assigning each cell in the CA to a virtual processor, the algorithm as described in Fig. 6.1 can be applied

```
While t(c) < SimulationTime Do

    WaitUntil t(c) <=
       min{c' ∈ Neighbors | t(c')}

    State(c) ← NewState(c,
       State(Neighbors(c)), t(c))

    NextEventTime ←
       DetermineTime(c, t(c))
```

Figure 6.1: *An algorithm for conservative parallel discrete event simulation.*

Each cell waits until it has the lowest time stamp in its local neighborhood or *Region Of Influence* (ROI). For a CA the ROI is determined by the radius of the neighborhood. This scheme is deadlock free, i.e., the `WaitUntil` condition (see Fig. 6.1) is always true for at least one cell in the entire population, and therefore the waiting loop can never be indefinite for all cells. A slightly different ap-

proach is to assign a number of cells to each processor. This aggregate of cells has a single local time. All processors concurrently choose a random cell from this local aggregate and check the minimum time condition against all neighboring processors in their ROI. The cells is updated as soon as this constraint is met.

6.3.2 Optimistic Parallel Discrete Event Simulation

A serious disadvantage of conservative approaches as described above, is that the method is sometimes *too* conservative and unable to exploit the available parallelism in the simulation. Consider for instance the situation where a cell, residing somewhere far from the borders of a local aggregate on processor A, gets chosen for updating. It may occur that a processor B in the ROI of the cell's processor has a lower simulation time. This situation implies that processor A must wait for processor B to catch up in simulated time; even though it may be the case that the update does not need information from this processor B, due to the ROI of the cell which is generally a small subset of the sub-domain of B. Therefore, we may as well update this cell.

An optimistic parallel discrete event simulation mechanism that takes advantage of this situation is the so-called *Time Warp* scheme [78, 116, 119, 117]. In Time Warp a processor's clock may run ahead of the clocks of the neighboring processors. In contrast to conservative approaches, Time Warp does not need to determine whether it is safe to proceed and events are always updated.

As a consequence, the local simulation time of a processor may get ahead of simulation times of its neighbors. If an event occurs that takes place on the boundary of a neighboring sub-domain, a causality error can be triggered by this boundary event if it has a smaller time stamp. A boundary event is defined as all updates in a processor domain who's ROI expands to neighboring processors. All events on those neighboring processors that are ahead of time, have to undo all their premature updates down to the boundary event that triggered the causality error. The process of undoing all these changes is called a *rollback*, and requires the administration of an event history list.

Thrashing behavior in Time Warp occurs when correcting causality errors consumes more computation time than the forward simulation. This thrashing behavior is induced by overly optimistic behavior of the simulation protocol. The optimistic behavior is a combination of aggressiveness and risk: aggressiveness in the execution of events without the guarantee of freedom of errors, and risk is the property by which the results of aggressive processing are propagated to other LPs. An additional parameter to control the amount of aggressiveness has been added to Time Warp by Overeinder et. al [118]. To shorten the periods of thrashing, the aggressiveness of the protocol must be throttled, that is, the simulation execution mechanism should not execute events that lie in the remote future as it is likely that these events have to be rolled back eventually. In effect, the progress of the individual simulation process should be bound to a limited simulation time window, the so called *virtual time window*. In this way,

the processes are forced to synchronize with each other in a short time frame, after which the simulation can continue as before.
Notice that the radius of a neighbor set determines the maximal speedup that can be achieved by using any PDES formulation, i.e., both for conservative as well as optimistic methods. Increasing the radius, increases the area of overlapping updates. In addition, the communication volume increases as a result of larger communication strips that have to be exchanged.

6.4 Self Organized Criticality

Most of the time, equilibrium systems with short-range interactions, exhibit exponentially decaying correlations. Infinite correlations, i.e., scale invariance, can be achieved by fine tuning some parameters (temperature) to a critical value. An example of such a system is the Ising spin model of magnetization. In Chapter 4 of this thesis we have also encountered a phase transition in the statistical ensembles of problem instances stemming from parallel computing: allocation of parallel tasks.
In addition, a large class of non-equilibrium locally interacting nonlinear systems spontaneously develop scale invariance. These systems are subject to external driving: local "energies" are increased from the outside until some threshold condition holds. At the point where the critical value is reached the local state variables trigger a chain reaction by transporting its "stored energy" to neighbouring sites. At the steady state of the system, assured by open boundary conditions, chain reactions or avalanches of all sizes can occur. The distribution of "avalanche sizes" s obeys critical scaling:

$$P(s) \sim s^{-\tau} \tag{6.1}$$

where τ is a critical exponent and most other observables of the system have no intrinsic time or length scale. The size of an avalanche can be defined in different ways. It can be measured by the number of relaxation steps needed for the chain reaction to stop or the total number of sites involved in the avalanche. This phenomenon of spontaneously developing critical scaling has been called Self Organized Criticality (SOC) [9]. A lot of naturally occurring systems exhibit this kind of scaling- or self-similar behavior. The basic model in which SOC behavior is demonstrated, is with a special kind of Cellular Automaton, which simulates the sliding and growing of sand piles [9]. A discrete variable h is assigned to each site on a d-dimensional lattice, representing the height of a "slope". Units of sand are added subsequently to a random site until some critical threshold h_c is reached. If the state of a site has reached the critical value, a sliding event takes place: the sand of the site is distributed among its nearest neighbors. The rules for a d-dimensional version of such a model are:

$$h(\mathbf{r}, t+1) \rightarrow h(\mathbf{r}, t) - 2d \tag{6.2}$$

$$h(\mathbf{r} \pm \mathbf{e_j}, t+1) \rightarrow h(\mathbf{r} \pm \mathbf{e_j}, t) + 1 \tag{6.3}$$

where $\mathbf{r}$ is a lattice position, $\mathbf{e_j}$ is used to denote the j-th neighbour, and t is the iteration counter. The rules actually describe a non-linear diffusion equation. If the system has relaxed, another random perturbation is made. We have repeated the experiments with a continuous version of this model where randomly chosen energy quanta ε_h ($[0,1]$) are added to a randomly chosen site. If some site reaches the critical energy value 1.0, the system is relaxed until every site is below this critical value. At some point the system reaches a stationary state where the average energy has reached a steady value $< h >$. At this point, the addition of an amount of energy to some site can trigger avalanches of arbitrary sizes. From this point we measured the distribution of both the duration and size of the avalanches. Figs. 6.2 shows the distribution of the avalanche sizes. It is clear that the finite size distribution can be fitted by a function obeying a power law.

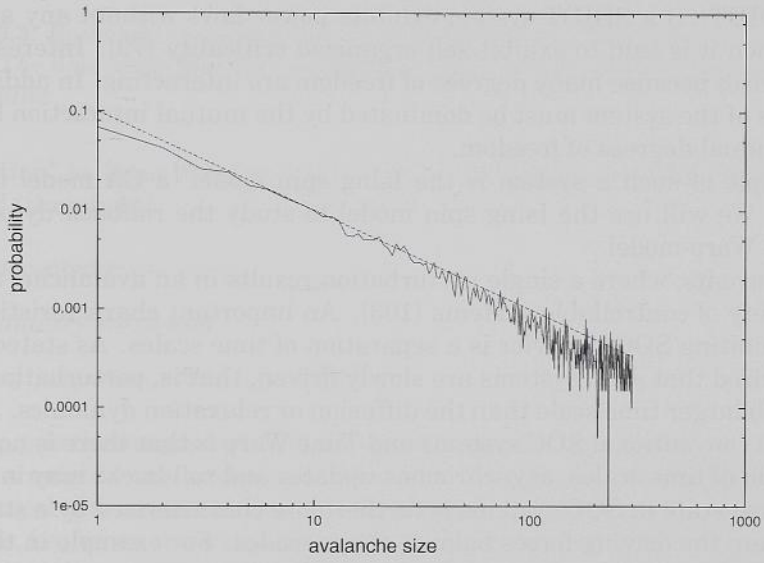

Figure 6.2: *Avalanche size distribution in a 100x100 lattice.*

The idea of Self Organized Criticality is assumed to be an explanation of the emergence of critical scaling behavior in many naturally observed systems. If this scaling behavior is spatial the signature of the system is often fractal. Many growth phenomena exhibit this self-similar behavior, e.g., DLA, Dielectric Breakdown (DB), Invasion Percolation, etc. Hence a logical step is to use SOC as a possible theory of the growth of fractal patterns. The idea that a dynamical system with spatial degrees of freedom evolves into a self organized critical state could very well be applied to these familiar models of growth. Some work in this area has been done on the DB model [4, 121, 122]. Exactly at the point where extinction of a branch balances branching itself, the growth process is stable with respect to fluctuations. A stationary state is reached when branching has broken down to a level where the flow only barely survives [4].

The distribution of the sizes of extinct branches describe the self organized critical state, corresponding to the avalanches in the sandpile. Another way of characterizing the SOC state is by keeping track of the electrostatic field and the number of iterations required to reach relaxation [121] after a growth step. The relaxation steps of a numerical procedure, e.g., Jacobi iteration, can be used to quantify this measure. Alternatively one could measure the range of the disturbance, defined as the total number of lattice sites in which the electrostatic potential changes above a given threshold. If the growth in the SOC regime, the range distribution can be described by a power law function.

6.5 SOC in PDES dynamics: experimental results

SOC behavior is found in *slowly driven*, *interaction-dominated threshold* systems (SDIDT), if a SDIDT system exhibits power laws without any apparent tuning then it is said to exhibit self organized criticality [79]. Interesting behavior arises because many degrees of freedom are interacting. In addition the dynamics of the system must be dominated by the mutual interaction between the individual degrees of freedom.

An example of such a system is the Ising spin model (a CA model for magnetism). We will use the Ising spin model to study the rollback dynamics in the Time Warp model.

Cascade events, where a single perturbation results in an avalanche, are seen in a variety of controllable systems [103]. An important characteristic of systems exhibiting SOC behavior is a separation of time scales. As stated before, it is required that such systems are slowly driven, that is, perturbations occur on a much larger time scale than the diffusion or relaxation dynamics. A difference with conventional SOC systems and Time Warp is that there is no explicit separation of time scales, asynchronous updates and roll-backs may intervene. The critical state in SOC systems is furthermore characterized by a stationary state where the driving forces balance the cascades. For example in the sandpile CA, adding sand causes on the one hand the pile to grow, on the other hand avalanches. The dynamically stationary state is obtained at the "critical point" where these two effects exactly balance.

A highly speculative analogy can be made with Time Warp: adding events causes the event rate to grow whereas on the other hand roll-backs may occur. In TW the event rate eventually reaches a kind of stationary state with superimposed rollback cascade effects (See Fig. 6.3). In this figure it can be observed that, after a short transient, the number of processes events eventually reaches a "steady" rate of approximately 19000. This steady state is distorted by periods of thrashing.

We define two time scales in Time Warp: *simulation time* and *protocol time*. The simulation time in Time Warp is updated by the rate at which the system is driven. This driving rate is determined by the dynamics of the simulation. The protocol time, needed to process a rollback, is determined by machine specific parameters.

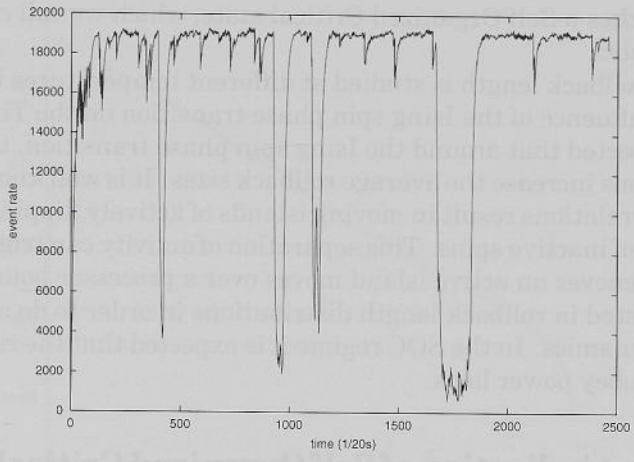

Figure 6.3: *Event rate vs. wall clock time for a PDES Ising simulation with* $P = 24$ *and* $T = 1.0$. *The time series has been smoothened using an exponentially weighted moving average filter.*

"Relaxation" in Time Warp occurs whenever the following three conditions are satisfied (threshold):

- if *accepted event* and
- *boundary event* and
- $\forall\, i \in \mathit{neighborhood}(\mathit{local}) : t_{\mathit{local}} < t_i$

where t_{local} is the simulation time on the local processor and t_i is the simulation time on a neighboring processor i. For Ising spin simulations, simulation time and protocol time separate at low temperatures, when there are not that many spin flips, i.e., when the acceptance ratio of the Metropolis algorithm is low. For high temperatures many spin flips are accepted and updates and roll-backs occur at comparable time scales.
We experiment with two different grid decompositions for the parallel simulation of the Ising dynamics on an $l * l$ lattice: a 1 dimensional "slice" decomposition and a 2D "box" decomposition. Both decompositions are composed to assure optimal load balance. For all parameter instances of the simulation experiments we measure "average rollback length" and "rollback length distributions."
It is very important to note that, in fact, we are confronted with two kinds of critical behavior. The critical behavior of the first kind is a result of the Ising spin phase transition at the critical temperature T_c. At the Ising spin phase transition, long range spin correlations occur, that probably influence the Time Warp dynamics. We call this critical behavior of the first kind the *physical critical behavior*. The critical behavior of the second kind is conjectured, and is caused by SOC. We assume that in the low temperature Ising regime the Time Warp

dynamics reaches a Self Organized Critical state, which we call *computational critical behavior*.
The average rollback length is studied at different temperatures in order to determine the influence of the Ising spin phase transition on the Time Warp protocol. It is expected that around the Ising spin phase transition, the long range spin correlations increase the average rollback sizes. It is well known that these long range correlations result in moving islands of actively flipping spins are located in a sea of inactive spins. This separation of activity can trigger very large roll-backs whenever an active island moves over a processor boundary.
We are interested in rollback length distributions in order to do a first check of SOC in TW dynamics. In the SOC regime it is expected that the rollback length distributions obey power laws.

6.5.1 A first indication of Self Organized Criticality in Time Warp

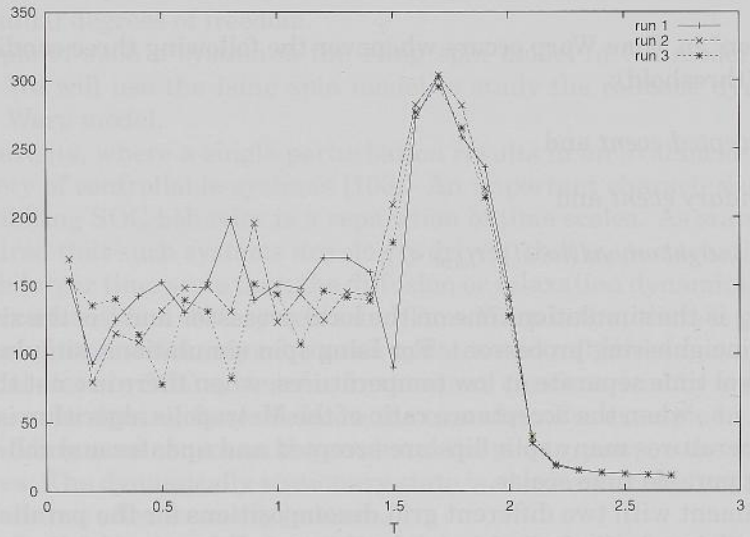

Figure 6.4: *Average rollback length for different temperatures. For each temperature, the results of 3 experiments are shown. Using the simulation parameters* $l = 220$, $P = 12$ *and* $VT = 3000$ *and a 1D decomposition.*

For the first series of experiments in this section we have fixed the lattice size to $l = 220$, the number of processors to $P = 12$ and the virtual time window (VT) to 3000, using a "sliced" 1D decomposition. In Fig. 6.4 the average rollback lengths are shown for the temperature range [0.1-2.7]. The rollback lengths are averaged over time for all processors. The results of three different runs are depicted in the figure. Close to the Ising phase transition ($T_c \approx 2.2$ for infinite lattices), the expected peak in the average rollback length can be observed. From this

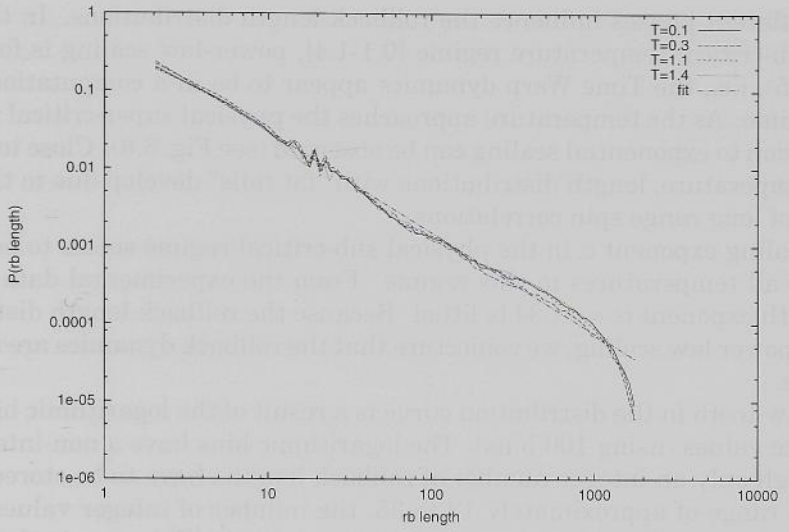

Figure 6.5: *Rollback distribution for temperatures in the range 0.1-1.4, fitted exponent has value -1.34 (± 0.05). Using the parameters $l = 220$, $P = 12$ and $VT = 3000$ and a 1D decomposition.*

figure, three different regimes can be identified: the *physical sub-critical phase* ($< T_c$), the *physical critical phase* ($\approx T_c$) and the *physical super-critical phase* ($> T_c$).

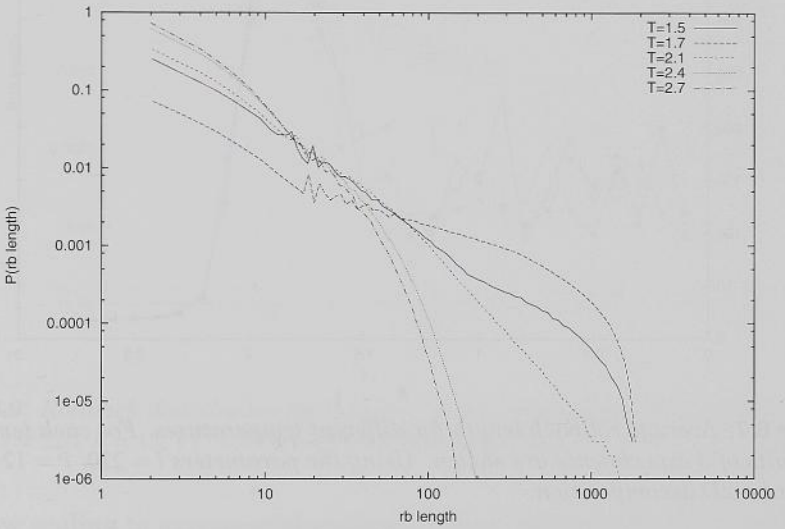

Figure 6.6: *Rollback distribution for temperatures in the range 1.5-2.7. Using the parameters $l = 220$, $P = 12$ and $VT = 3000$ and a 1D decomposition.*

The different phases influence the rollback length distributions. In the physical sub-critical temperature regime [0.1-1.4], power-law scaling is found (see Fig. 6.5), i.e., the Time Warp dynamics appear to be in a computational critical regime. As the temperature approaches the physical super-critical regime a transition to exponential scaling can be observed (see Fig. 6.6). Close to the critical temperature, length distributions with "fat tails" develop due to the emergence of long range spin correlations.
The scaling exponent α in the physical sub-critical regime seems to be universal for all temperatures in this regime. From the experimental data a power law with exponent $\alpha = -1.34$ is fitted. Because the rollback length distribution obeys power law scaling, we conjecture that the rollback dynamics are in a SOC regime.
The saw-tooth in the distribution curve is a result of the logarithmic binning of discrete values (using 100 bins). The logarithmic bins have a non-integer size, although only an integer number of rollback lengths have to be stored. In the length range of approximately 12 to 35, the number of integer values that fit in the corresponding bins alternate between 1 and 2. However, in the binning procedure the bin values are normalized by the floating point sizes of the corresponding bins. The differences in these sizes is very small, while the number of integers that can actually be stored can vary by a factor of 2. The saw-tooth is therefore a discreteness artifact in this length region.

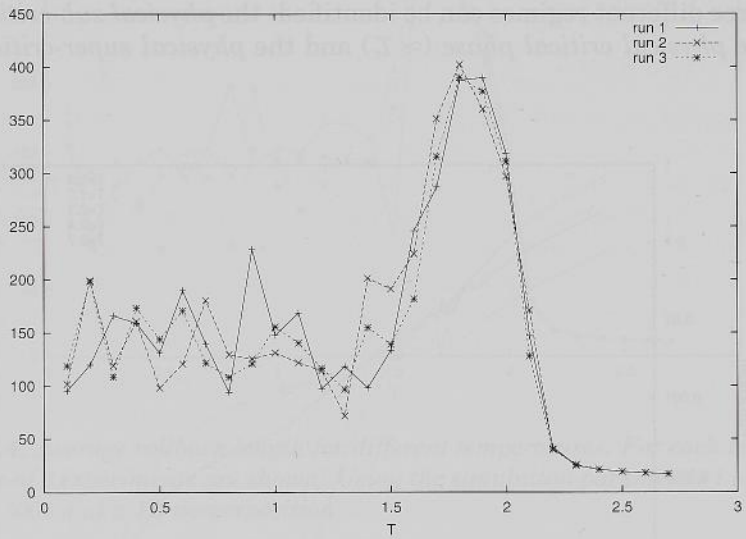

Figure 6.7: *Average rollback length for different temperatures. For each temperature, the results of 3 experiments are shown. Using the parameters* $l = 220$, $P = 12$ *and* $VT = 3000$ *and a 2D decomposition.*

For the 2D decomposition we repeat the same set of experiments as for the 1D case, again with $l = 220$ and $P = 12$. As for the 1D case, we observe a peak in the average rollback lengths around T_c (see Fig. 6.7). Again a transition from

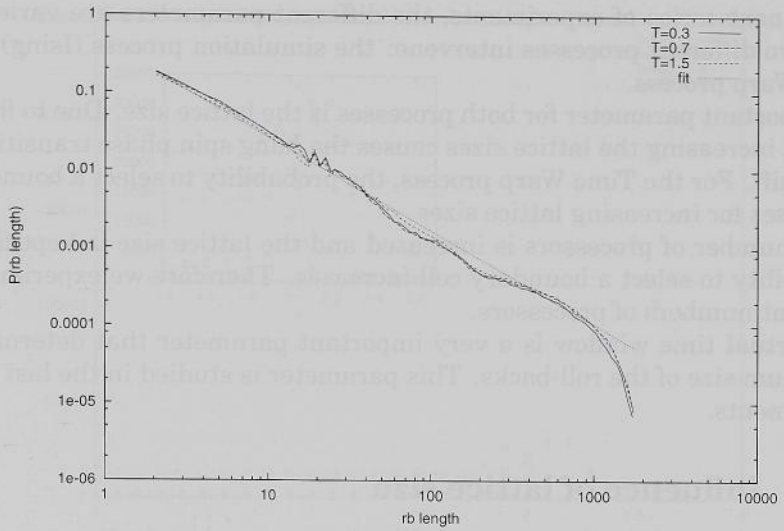

Figure 6.8: *Rollback distribution for temperatures in the range 0.3-1.5, fitted exponent has value -1.26 (± 0.05). Using the parameters $l = 220$, $P = 12$, $VT = 3000$ and a 2D decomposition.*

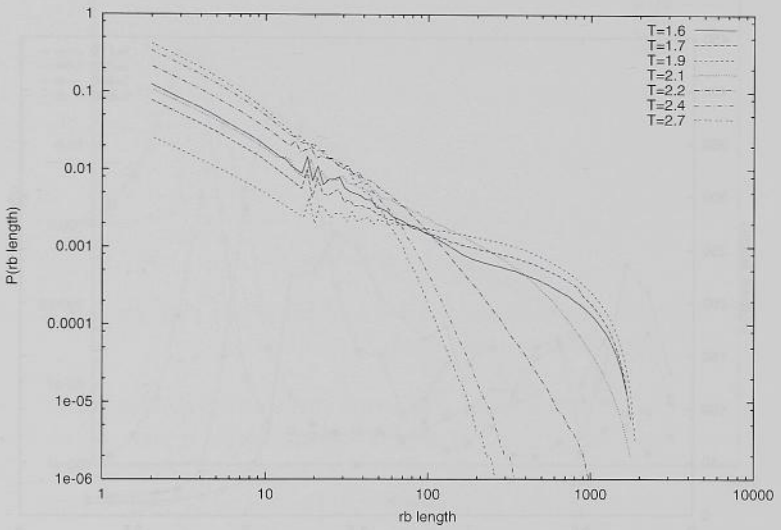

Figure 6.9: *Rollback distribution for temperatures in the range 1.6-2.7. Using the parameters $l = 220$, $P = 12$, $VT = 3000$ and a 2D decomposition.*

power law scaling to exponential scaling is observed (see Figs. 6.8 and 6.9). In the physical sub-critical regime we find $\alpha = -1.26$, slightly lower as in the 1D case. This could be the result slight of shorter distances between processor partitions, enabling a faster propagation of cascading roll-backs.

In the next series of experiments, the different parameters are varied. Note that two different processes intervene: the simulation process (Ising) and the Time Warp process.
An important parameter for both processes is the lattice size. Due to finite size effects, increasing the lattice sizes causes the Ising spin phase transition point T_c to shift. For the Time Warp process, the probability to select a boundary cell decreases for increasing lattice sizes.
If the number of processors is increased and the lattice size is kept fixed, the probability to select a boundary cell increases. Therefore we experiment with different numbers of processors.
The virtual time window is a very important parameter that determines the maximum size of the roll-backs. This parameter is studied in the last series of experiments.

6.5.2 Influence of lattice size

Because the peak in the average rollback appears close to T_c, we expect that this peak is related to the long range correlations of the Ising phase transition. To support this hypothesis we have conducted a number of experiments with increasing lattice sizes in order to study the presence of finite size effects. For finite lattices, the critical temperature T_c shifts with increasing lattice sizes.

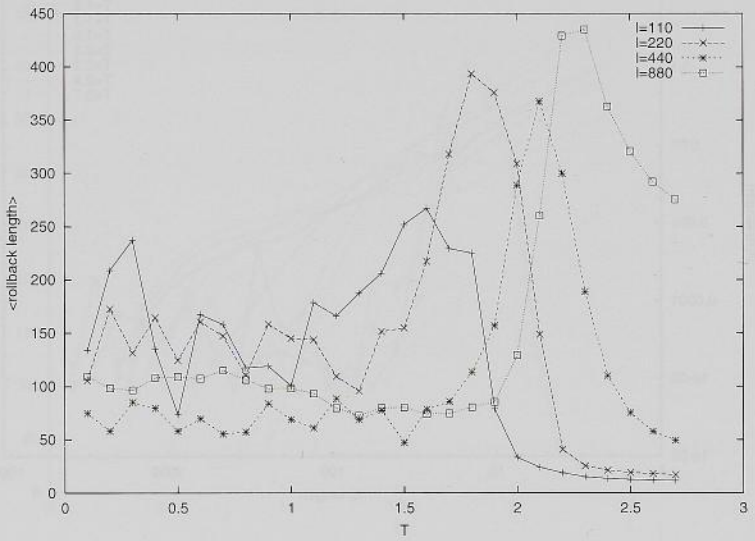

Figure 6.10: *Average rollback for* $l = \{110, 220, 440, 880\}$ *using the parameters* $P = 12$, $VT = 3000$ *and a 2D decomposition.*

Due to limited computer and time resources we did not extract any critical exponents from the generated data, a more detailed study of this phenomenon is therefore necessary. To simulate an Ising spin system on one specific $T \approx T_c$, more than 2 days (see Fig. 6.11) of computing time on a 12 node Pentium II (at

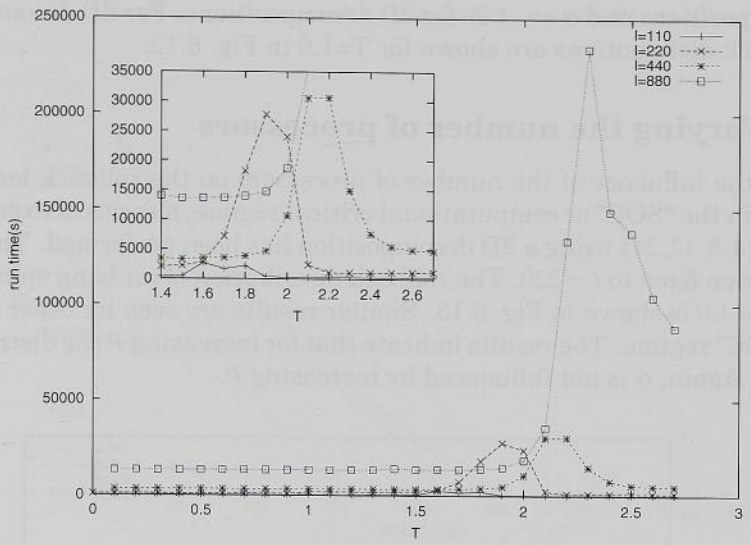

Figure 6.11: *Run times for l={110,220,440,880} using the parameters $P = 12$, $VT = 3000$ and a 2D decomposition.*

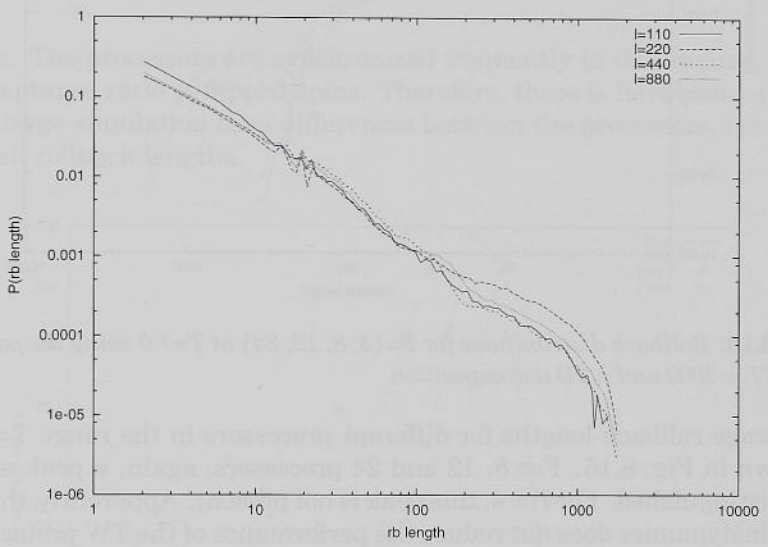

Figure 6.12: *Rollback distributions for l={110, 220, 440, 880} at T=1.0 using the parameters $P = 12$, $VT = 3000$ and a 2D decomposition.*

300 MHz) is needed, this gives an indication on the total amount of computer time required to run these experiments. Fig. 6.10 shows the results for varying lattice size experiments (using $l = \{110, 220, 440, 880\}$ and $P = 12$). A shift towards T_c is observed for increasing lattice sizes.
Furthermore we find that for all lattice sizes in the SOC regime $\alpha \approx -1.34$ for

1D decompositions and $\alpha \approx -1.26$ for 2D decompositions. For 2D decomposition the rollback distributions are shown for T=1.0 in Fig. 6.12.

6.5.3 Varying the number of processors

To study the influence of the number of processors on the rollback length distribution in the "SOC" or computational critical regime, a series of experiments with $P = \{4, 8, 12, 24\}$ using a 2D decomposition has been performed. The lattice size has been fixed to $l = 220$. The rollback distribution of an Ising spin simulation at $T = 1.0$ is shown in Fig. 6.13. Similar results are seen for other T values in the "SOC" regime. The results indicate that for increasing P the distributions converge. Again, α is not influenced by increasing P.

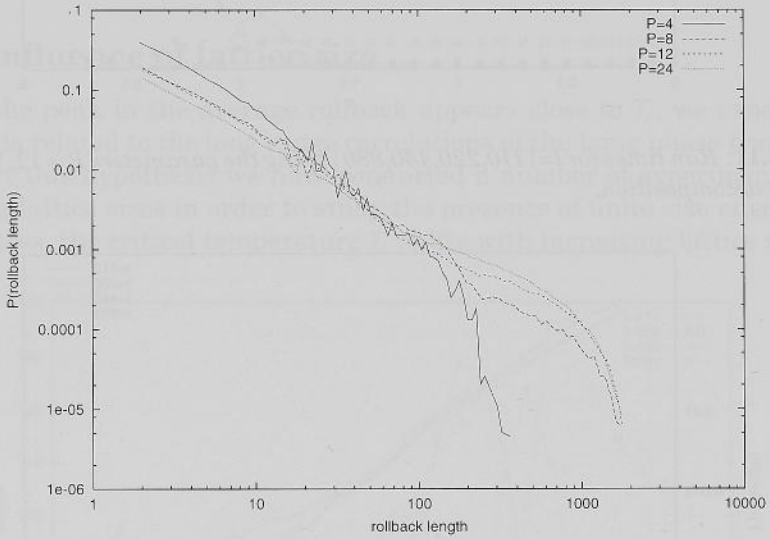

Figure 6.13: *Rollback distributions for P={4, 8, 12, 24} at T=1.0 using the parameters* $l = 220$, $VT = 3000$ *and a 2D decomposition.*

The average rollback lengths for different processors in the range T=[0.1-2.7] are shown in Fig. 6.15. For 8, 12 and 24 processors, again, a peak around T_c can be distinguished. For $P = 4$, this peak is not present. Apparently, the critical Ising spin dynamics does not reduce the performance of the TW protocol if only 4 processors are used.

In the low temperature regime the average rollback length increases with the number of processors. From Fig. 6.13 we observe that the rollback length cutoff size increases with P. Therefore it is expected that the average rollback lengths must increase with P in the low T regime (see Fig. 6.15).

This is not valid anymore in the high T regime, where the rollback length distributions approximately collapse (see Fig. 6.14) to the same exponentially decreasing distribution. As a result, so do the average rollback lengths (see Fig. 6.15). In the high T regime the rollback lengths are not influenced by P as in the low

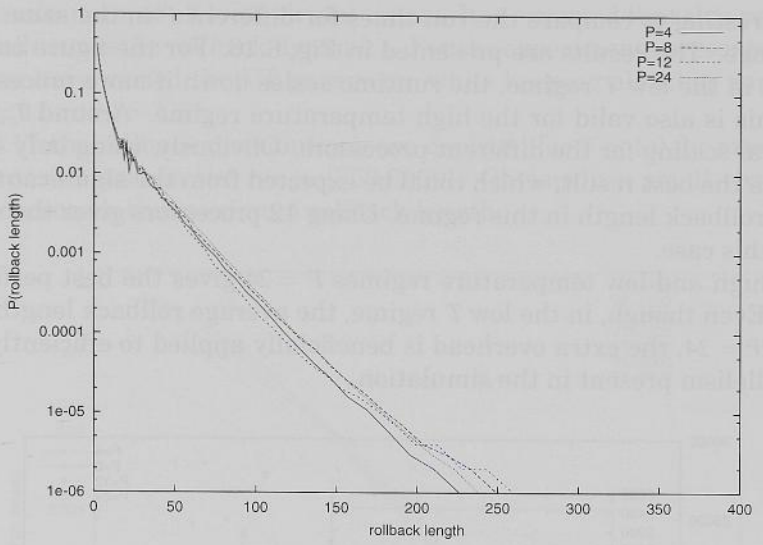

Figure 6.14: *Rollback distributions for P={4, 8, 12, 24} at T=2.7 using the parameters $l = 220$, $VT = 3000$ and a 2D decomposition. Note that the horizontal axis has a linear scale.*

T regime. The processors are synchronized frequently in this regime, due to a high acceptance ratio of flipped spins. Therefore, there is hardly any real time to build large simulation time differences between the processors, resulting in only small rollback lengths.

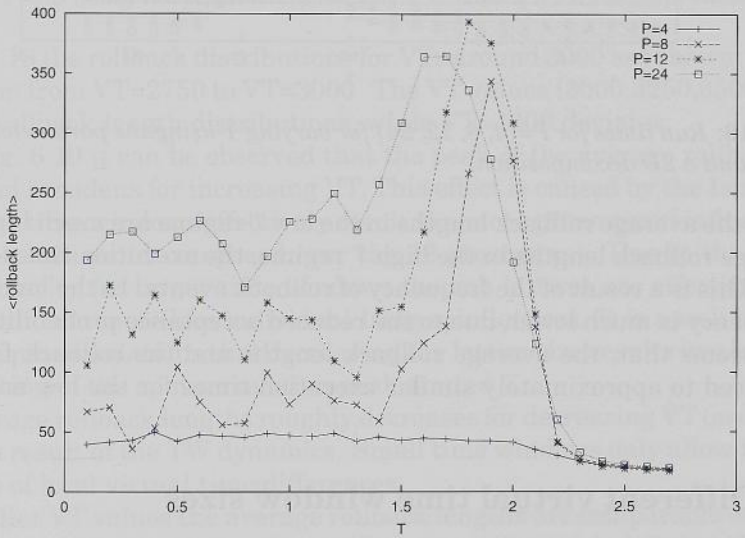

Figure 6.15: *Average rollback lengths for P={4, 8, 12, 24} for varying T using the parameters $l = 220$, $VT = 3000$ and a 2D decomposition.*

It is interesting to compare the run times for different P in the same temperature range. The results are presented in Fig. 6.16. For the figure on can derive that in the low T regime, the runtime scales down if more processors are used. This is also valid for the high temperature regime. Around T_c, we find non-trivial scaling for the different processors. Obviously, using only 4 processors gives the best result, which could be expected from the significantly lower average rollback length in this regime. Using 12 processors gives the worst results in this case.
For the high and low temperature regimes $P = 24$ gives the best performance results. Even though, in the low T regime, the average rollback length is maximal for $P = 24$, the extra overhead is beneficially applied to efficiently exploit the parallelism present in the simulation.

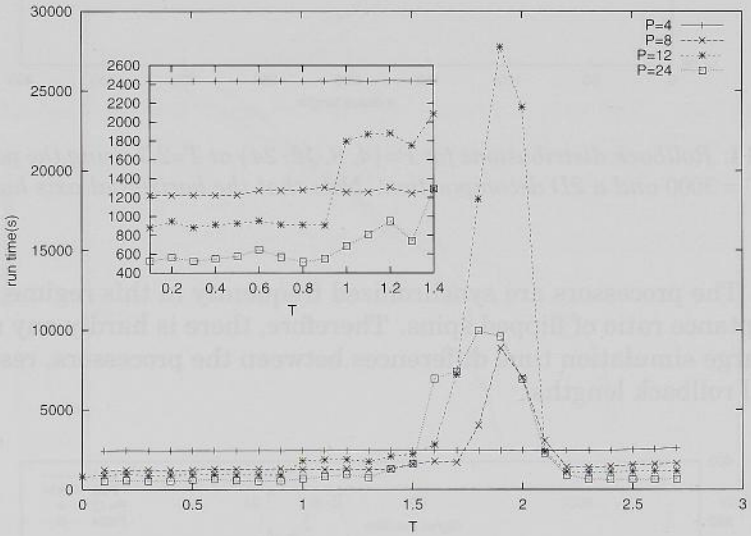

Figure 6.16: *Run times for P={4, 8, 12, 24} for varying T using the parameters* $l = 220$, $VT = 3000$ *and a 2D decomposition.*

Although the average rollback lengths in the low T regime are much larger than the average rollback lengths in the high T regime, the execution times are comparable. This is a result of the frequency of rollback events. In the low T regime this frequency is much lower, due to the reduced acceptance probability of spin flips. It seems that, the average rollback lengths and the rollback frequency are balanced to approximately similar execution times for the low and high T regimes.

6.5.4 Different virtual time window sizes

An important parameter of the TW protocol is the so called *virtual time window* (VT). This parameter control the asynchronicity of the simulation. It specifies the maximum difference between the local virtual time and the global virtual

time (the minimum of all local virtual times). It is expected that this parameter greatly influences the rollback dynamics. For the experiments presented in this section we have varied the VT parameter while keeping all other parameters fixed ($P = 12$ and $l = 220$).
In Fig. 6.17 the rollback distributions are depicted for $T = 1.0$ for experiments with VT parameters in the range [750,6000]. Obviously, a small virtual time window decreases the maximum rollback length.

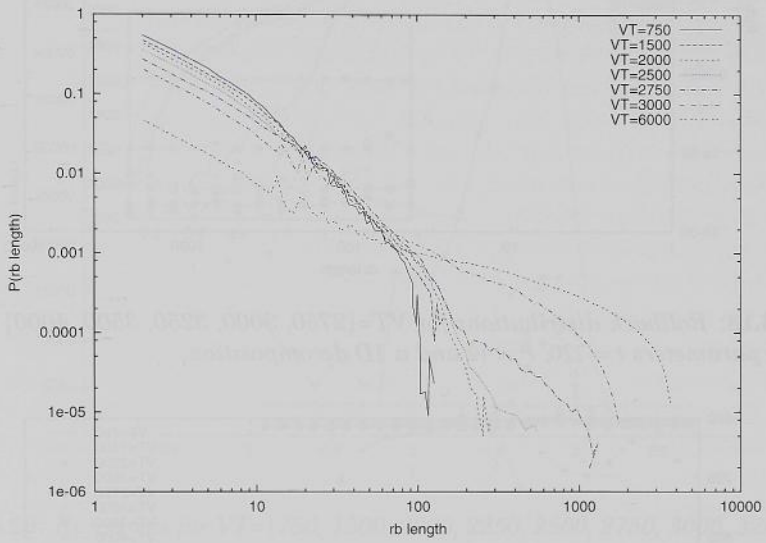

Figure 6.17: *Rollback distributions for VT={750, 1500, 2000, 2250, 2500, 2750, 3000, 6000} at T=1.0 using the parameters $l = 220$, $P = 12$ and a 1D decomposition.*

In Fig. 6.18 the rollback distributions for VTs around 3000 are shown. There is a transition from VT=2750 to VT=3000. The VT values [3000,3250,3500] produce similar rollback length distributions, while VT=4000 deviates.
From Fig. 6.19 it can be observed that the peak of the average rollback length shifts and broadens for increasing VT. This effect is caused by the Ising dynamics. Large time windows effectively result in a more pronounced influence of the finite sub lattices (decomposed over the 12 processors). Due to the increased asynchronicity for larger time windows the sub lattices are effectively loosely coupled and act more like individual Ising spin lattices. It is a well known fact in Ising spin simulations that decreasing the lattice size results in a broadening and shifting of the spin correlation peak around T_c.
The average rollback lengths roughly decreases for decreasing VT (see Fig. 6.19). This is a result of the TW dynamics. Small time windows only allow for a small build up of local virtual time differences.
For smaller VT values the average rollback lengths are comparable over the entire temperature range. For these values it can be concluded that the TW dynamics are not constrained to the details of the Ising dynamics. The increased synchronization frequency disables the build up of large time differences.

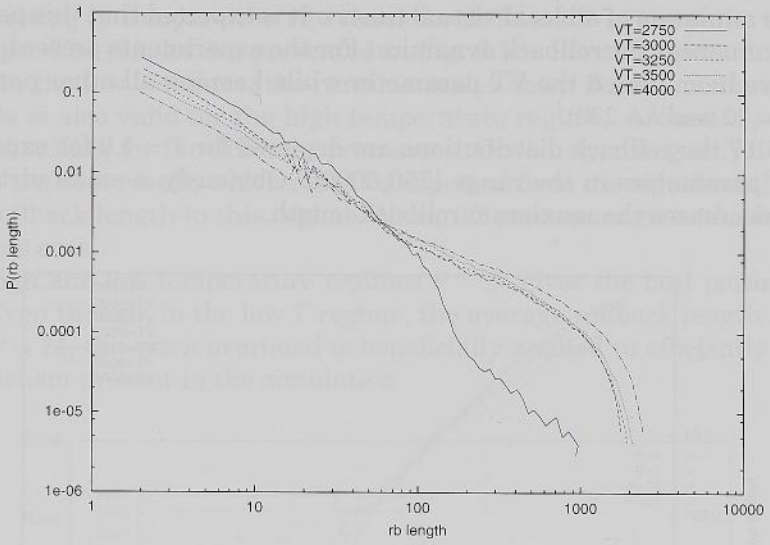

Figure 6.18: *Rollback distributions for VT={2750, 3000, 3250, 3500, 4000} at T=1.0 using the parameters $l = 220$, $P = 12$ and a 1D decomposition.*

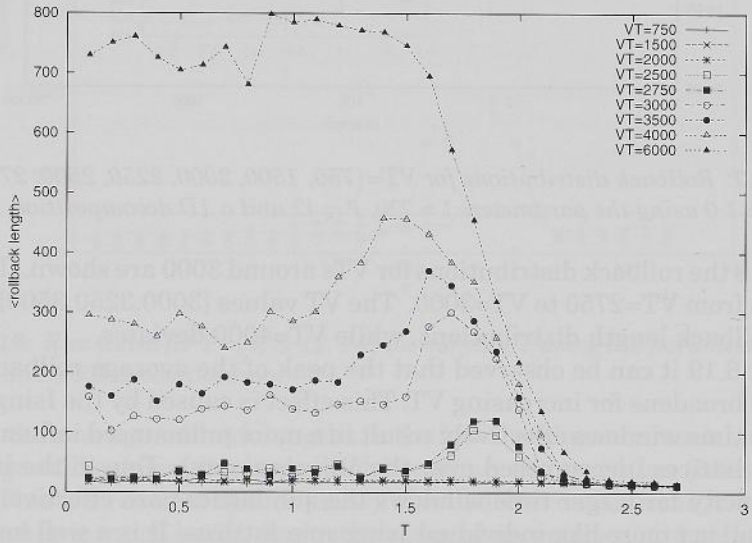

Figure 6.19: *Average rollback length for VT={750, 1500, 2000, 2250, 2500, 2750, 3000, 3500, 4000, 6000} for varying T using the parameters $l = 220$, $P = 12$ and a 1D decomposition.*

Somewhere there is a crossover point where increasing the maximum rollback lengths (by increasing VT) does not improve the progress in simulation time due to the increased protocol overhead. For this specific simulation instance it seems that VT=2750 is optimal for the low and high temperature regime (see

Fig. 6.20). For the regime around T_c it is almost optimal. Note that VT=3000 is comparable to VT=2750 in the low and high T regimes, while around T_c, VT=3000 produces significantly higher execution times. Hence the run times are highly susceptible for VT around T_c, as a consequence of the critical Ising dynamics.

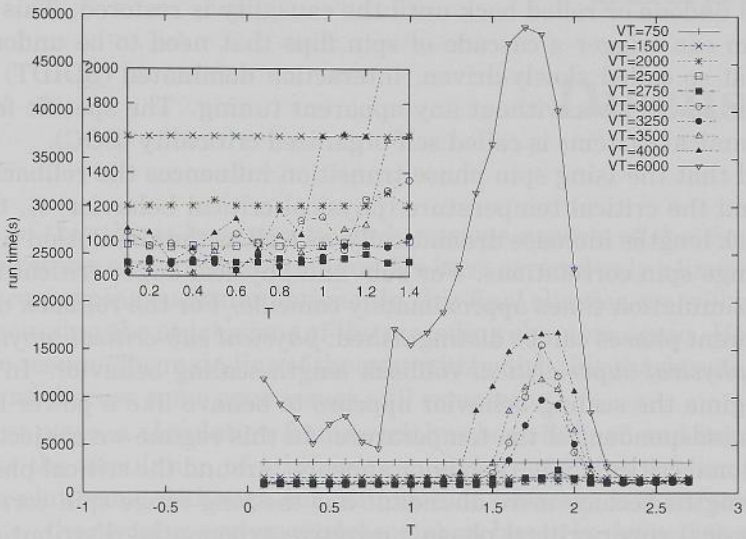

Figure 6.20: *Run times for VT={750, 1500, 2000, 2250, 2500, 2750, 3000, 3250, 3500, 4000, 6000} for varying T using the parameters* $l = 220$, $P = 12$ *and a 1D decomposition.*

In contrast to the disappearance of the average rollback length peak in Fig. 6.19 for increasing VT, a peak remains in the run time curves. This can be explained from reduced rollback frequencies in lower temperature regimes. The increase of the run times around T_c with increasing VT can be explained from the increased average rollback lengths (see Fig. 6.19) and the fat tail in the rollback size distribution around T_c for large virtual time windows (data not shown).

6.6 Conclusions

In this chapter we have intensively studied the dynamical behavior of the Time Warp protocol for parallel discrete event simulations. As a simulation case we considered the Ising spin model, which is basically a cellular automata model. The most common update dynamics for the Ising spin model is random selection of spins. The random spin selection mechanism can be cast into a parallel discrete event (PDES) update scheme due to the inherent locality of the Ising spin Hamiltonian. A single spin flip only influences its 4 direct spin neighbors in a 2 dimensional grid, therefore time locality can be exploited. The PDES scheme either uses a conservative or an optimistic protocol. Time Warp is an example of

an optimistic protocol, which exploits the maximal inherent parallelism when properly tuned.

A property of the Time Warp protocol is the appearance of so called roll-backs whenever a (time) causality error occurs. A causality error occurs when a spin flip event needs information from spins that are ahead in time. As a consequence the flips of the neighboring spins, that have a higher simulation time, need to be undone or rolled back until the causality is restored. This rollback mechanism can trigger a cascade of spin flips that need to be undone. It is known that so called slowly-driven, interaction dominated (SDIDT) systems can exhibit power laws without any apparent tuning. The specific feature of these dynamical systems is called self-organized criticality (SOC).

It is found that the Ising spin phase transition influences the rollback behavior. Around the critical temperature (physical critical behavior) T_c, the average rollback lengths increase dramatically, as well as the simulation times, due to long range spin correlations. For sub- and physical super-critical temperatures the simulation times approximately coincide. For the rollback dynamics three different phases can be distinguished: *physical sub-critical*, *physical critical*, and *physical super-critical* rollback length scaling behavior. In the sub-critical regime the scaling behavior appears to behave like a power law, with exponents independent of the temperature. In this regime we conjecture that computational critical (SOC) behavior appears. Around the critical phase large rollback lengths become more abundant due the long range spin correlations. In the physical super-critical phase a negative exponential distribution of the rollback lengths is observed.

Obviously a lot of work remains to be done in the study of physical- and computational critical behavior in Time Warp. The results presented in this chapter are, to our knowledge, the first series of experiments that have ever been conducted to study the influence and the appearance of critical behavior in Time Warp.