LA-UR-17-28909 (Accepted Manuscript)

Speculation and replication in temperature accelerated dynamics

Zamora, Richard James
Perez, Danny
Voter, Arthur Ford

To be published in: Journal of Materials Research

DOI to publisher's version: 10.1557/jmr.2018.17

Permalink to record: http://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-17-28909

Speculation and Replication in Temperature Accelerated Dynamics

Richard J. Zamora, Danny Perez and Arthur F. Voter [a)]

Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545

Abstract

Accelerated Molecular Dynamics (AMD) is a class of MD-based algorithms for the long-timescale simulation of atomistic systems that are characterized by rare-event transitions. Temperature Accelerated Dynamics (TAD), a traditional AMD approach, hastens state-to-state transitions by performing MD at an elevated temperature. Recently, Speculatively Parallel TAD (SpecTAD) was introduced, allowing the TAD procedure to exploit parallel computing systems by concurrently executing in a dynamically generated list of speculative future states. Although speculation can be very powerful, it is not always the most efficient use of parallel resources. Here, we compare the performance of speculative parallelism with a replica-based technique, similar to the Parallel Replica Dynamics method. A hybrid SpecTAD approach is also presented, in which each speculation process is further accelerated by a local set of replicas. Overall, this work motivates the use of hybrid parallelism whenever possible, as some combination of speculation and replication is typically most efficient.

I. Introduction

Given the availability of parallel computing resources, it has become more valuable than ever to design atomistic simulation algorithms that are well-suited for distributed computing clusters. When it comes to Molecular Dynamics (MD) simulations, algorithm development has mostly focused on

[a)] Address all correspondence to this author.
e-mail: afv@lanl.gov

improving weak scaling through spatial decomposition. These efforts have been very effective, allowing current researchers to routinely simulate billions of atoms or more [1] [2]. Unfortunately, spatial decomposition has done very little to extend the maximum temporal reach of MD beyond a few microseconds. This is because MD requires the sequential integration of Newton's equations, and a femtosecond-scale time step is typically needed to ensure numerical stability.

The Accelerated Molecular Dynamics (AMD) approach to the MD timescale problem is to produce a state-wise trajectory at a faster rate than direct MD. It does so by leveraging the separation of time scales between the numerical integration time step and state-to-state transitions. This approach is particularly well suited for applications where this separation is large, like point-defect evolution in crystalline materials. The performance of an AMD algorithm is often expressed in terms of a computational boost factor, given by WCT_{MD}/WCT_{AMD}, where WCT_{MD} and WCT_{AMD} refer to the required wall clock time (WCT) for direct MD and AMD, respectively.

Since the late 1990's, three generalizations of the AMD approach have been established: Hyperdynamics [3], Parallel Replica Dynamics (PRD) [4] [5], and Temperature Accelerated Dynamics (TAD) [6], each containing a variety of sub-methods. In all cases, an MD-based approach is used to identify an accurate state-to-state trajectory at an accelerated rate. Of the three general approaches, PRD is the most accurate, and the only one relying completely on computational parallelism to obtain a boost. Although PRD offers both parallel scaling and accuracy, other approaches usually require less cpu time to achieve the same boost (but potentially longer wall-clock times). In contrast, when the dynamics of a system is dominated by activated transitions that are well-described by Harmonic Transition State Theory (HTST) [7], the TAD method can be used on a single processor to achieve an exponential speedup over MD. For the silver adatom system discussed below, PRD would require more than 10^3 processors to produce the same boost as serial TAD.

Although TAD is a powerful method for the study of HTST-abiding systems, the serial boost factor can be further improved by introducing computational parallelism. For example, the Speculatively Parallel TAD (SpecTAD) method improves upon the performance of TAD by concurrently executing high-temperature MD within a dynamically generated list of expected future states [8]. This speculation-based technique, which will be investigated further in this work, has been shown to improve the TAD boost by orders of magnitude, and has already been used to illuminate new details of defect clustering behavior in complex oxides [9]. With the obvious successes of parallel speculation in mind, the current work is intended to demonstrate that other means of parallelism will often lead to a more efficient use of computational resources.

Here we present two new parallel TAD methods, both intended to utilize replica-based parallelism to accelerate the execution of high-temperature MD within a TAD-based algorithm. First, we present the Parallel Replica TAD (ReplicaTAD) method, which is designed to utilize every available cpu resource within the current state of the official TAD trajectory. Since all concurrent MD work is executed in the same state, this parallelism approach can be understood as pure replication; similar to the combination of TAD and PRD. We also present the *hybrid* SpecTAD method, which is designed to combine the advantages of both speculation and replication into a single algorithm. Using both theoretical predictions and real simulations, we compare all three parallel approaches, and ultimately motivate further development of an adaptive hybrid approach.

For all algorithms described in this work, a stochastic Langevin thermostat is assumed to provide temperature control during MD execution [10]. All transition checks are to be performed by minimizing the potential energy of the system with respect to the configuration of atoms, X^{atoms}, using a force-based minimization procedure. This transition-checking procedure will be referred to here as *quenching*.

II. Temperature-Accelerated Dynamics (TAD)

Before discussing approaches to parallelism, we first review the basics of the serial TAD algorithm. This brief introduction is not intended to provide the reader with a comprehensive understanding of TAD, but to provide a necessary foundation for the parallel algorithms investigated in this work. For a more thorough discussion, we refer the reader to the original paper [6] and a recent review article [8].

Consider the 3N+1-dimensional potential energy surface of a material system: it contains a large number of basins that correspond to different conformations of the system. These basins are often also referred to as states. For any AMD method to perform well, these basins should be separated by barriers that are high compared to $k_B T$, where T is the temperature of interest, and k_B is the Boltzmann constant. TAD further relies on the HTST assumption that the transition rate from a given state, along some pathway j, is given by

$$k_j = \nu_{0j} e^{\left(\frac{-E_j}{k_B T}\right)}, \tag{1}$$

where ν_{0j} is the temperature-independent pre-exponential factor and E_j is the transition energy barrier. Note that the transition rate depends on the inverse exponential of the temperature. This allows for the construction of a hypothetical trajectory at a low temperature of interest, T_{Low}, by observing transitions at an exponentially accelerated rate at T_{High}.

As illustrated in **FIG. 1 (a)**, TAD can be expressed as a recursive function, where each call to TAD is tied to a specific official state visit. Here, we use the word *official* to refer to a state that is visited by the hypothetical low-temperature trajectory. Each of these visits will correspond to a minimum-energy configuration, X^{atoms}, and the time when the official low-temperature trajectory began the visit, t^{Low}. Beginning with these initial conditions, we first execute a block of MD to thermalize the

system at T_{High} (line 10 in **FIG. 1 (a)**). Although X^{atoms} is allowed to evolve during this MD block, the official high-temperature time, t^{High}, is left unchanged. This thermalization stage is repeated until no transition is observed during the prescribed thermalization time.

Once thermalization is complete, and a so-called hot point is prepared, we evolve the trajectory by executing a block of MD and advancing t^{High} (line 14 in **FIG. 1 (a)**). This stage is repeated until a transition is detected, or the high temperature stop time, t^{Stop}, is reached. When a transition is detected, the exact transition time, t^{High}, is drawn from a uniform distribution of times over the preceding MD block. This high-temperature transition time is then mapped onto a hypothetical low-temperature time using

$$t_j^{low} = t_j^{High} e^{-E_j\left(\frac{1}{k_B T_{Low}} - \frac{1}{k_B T_{High}}\right)}. \quad (2)$$

This calculation depends on an accurate definition of the barrier height for the j pathway: $E_j = E_j^{saddle} - E(X_{Min})$, which is determined using the nudged elastic band (NEB) method [11] [12] (line 18 in **FIG. 1 (a)**). Once a transition has been detected and its corresponding t^{Low} has been defined, the previously described stages must be repeated until t^{High} reaches t^{Stop}. At this point, it would be naive to accept the first transition observed at high-temperature, because it is possible that a later high-temperature transition will map onto an even earlier low-temperature time. This means that t^{Stop} would need to be equal to t^{Low} in order to accept an event with certainty. However, in order to obtain an AMD boost over MD, t^{Stop} is typically defined by

$$t^{Stop} = \frac{\ln(1/\delta)}{\nu_{Min}} \left(\frac{\nu_{Min} t^{Low}}{\ln(1/\delta)} \right)^{T_{Low}/T_{High}}, \quad (3)$$

so that the winning transition is accepted with a desired confidence level. Here, δ defines the acceptable level of uncertainty, and ν_{Min} is the assumed lower bound on the pre-exponential factor.

Before moving on to parallelism, it is important to recognize that the interatomic force calculations (the force calls) account for most of the execution time of any AMD method. Even for a classical interatomic potential, the total run time of a serial TAD simulation is approximately equal to the product of the total number of force calls and the mean execution time of a single force call. As illustrated in the cartoon diagram in **FIG. 2 (a)**, this means that both quenching and NEB calculations can account for a significant fraction of the total execution time when many high-temperature transitions are detected before one is accepted.

III. Parallel Algorithms

Starting with the serial TAD algorithm described in the previous section, several opportunities for parallel execution are relatively obvious. For applications in which the force calls are computationally expensive, for example, a typical parallel decomposition routine is likely to provide a significant speedup. This approach does not require any modification to the serial TAD algorithm, and is highly recommended whenever it is efficient to do so. In fact, the code used for this work allows for spatial decomposition through a library binding to the popular LAMMPS engine [13]. Although this type of parallelism was not advantageous for the small physical test system used in this work, one should assume that any of the presented algorithms can (and should) be implemented with parallel force call capabilities.

In addition to spatial decomposition, there is also a clear opportunity to accelerate the saddle-point calculations. Since the NEB is performed by minimizing the total energy of an entire chain of configuration images, it is relatively straightforward to parallelize this calculation by decomposing the chain into groups of images. Although this approach should improve performance, the remaining serial parts of the TAD algorithm are likely to prevent the overall speedup from scaling very far. In contrast to the spatial decomposition of force calls, the code used for this work does not include parallel NEB

calculations, as the implementation has been left for future work.

Although there are a variety of ways to use parallelism to improve the performance of TAD, we will focus on speculation and replication algorithms for the remainder of this paper.

A. Speculative Parallelism

The basic concept of Speculatively Parallel TAD (SpecTAD) is that, for many physical systems, it is typical for a winning transition to be observed by a TAD process long before its corresponding t^{Stop} is actually reached. When there are multiple processors available, an idle core can (and should) be used to immediately begin executing a separate TAD process in what is expected to be the next official state (state B, for example). Recall that an *official* state is one that is visited by the hypothetical low-temperature trajectory. This early execution in state B is speculative in nature, because there is still a possibility that the parent TAD process will replace B as the next official state.

An ideal pattern of SpecTAD parallelism is illustrated in **FIG. 2 (b)** for the simulation of a three-transition trajectory with three speculative worker processes. This shows a special case, because the first transition observed at high temperature always corresponds to the transition that is ultimately accepted. With this simplification in mind, the diagram still accurately depicts the intended overlap of work between sequential states. This overlap implies that pure speculation can use approximately $\overline{w}_{\text{Stop}}/\overline{w}_{\text{winner}}$ cores efficiently, where the $\overline{w}$ quantities represent average WCT values and $\overline{w}_{\text{winner}}$ is the average WCT needed to observe the winning transition in each state. **FIG. 2 (b)** makes it clear that the number of cores used at any given time in a SpecTAD simulation cannot exceed the total number of official states accepted during the entire simulation. For the simple three-transition example in **FIG. 2 (b)**, this means that the parallel speedup must be less than three. When the number of accepted events (N_{accept}) is much larger than the number of available speculation processes (N_{spec}), the total run time can be approximated by

$$WCT_{Total} = max\left(\frac{\overline{w}_{winner}}{\overline{w}_{stop}}, \frac{1}{N_{spec}}\right)\overline{w}_{stop}N_{accept}. \quad (4)$$

The SpecTAD implementation used in this work requires two types of processes: a single master process, and a pre-defined number of speculation processes. The purpose of the master process is to build the official low-temperature trajectory and to manage the mapping of speculated work onto the various speculation processes. In order to understand the SpecTAD algorithm, it is most important to understand the behavior of a single speculation process, whose primary responsibilities are to wait for messages, and to execute the SpecTAD function detailed in **FIG. 1 (b)**.

A typical SpecTAD simulation begins with the master process sending the initial geometry, X^{atoms}, to an idle speculation process. The speculation will then follow a procedure that is very similar to the serial TAD algorithm. The most important difference is the call to a *Speculate* function (line 24 of **FIG. 1 (b)**), during which the process must alert the master that it has replaced its best guess for the next official state. Since the SpecTAD function is not recursive, and the speculation process is externally managed, it also becomes necessary for the speculation to alert the master when it has officially accepted the winning transition (line 27). Since it is often the case the first transition observed at high-temperature is not accepted, the master must also be able to interrupt speculation processes, and any dependent child processes, to reassign work. That is, when a speculation process is executing the SpecTAD function in a state that is no longer in the list of expected future states, the master must be able to inform the speculation process to wait for a different state to explore. This implementation detail was accomplished for this work by including message probes that are excluded from **FIG. 1 (b)** for simplicity.

B. Replica Parallelism

The goal of Parallel Replica TAD (ReplicaTAD) is to use all available resources to accumulate high-temperature MD time as quickly as possible within the TAD procedure. Like SpecTAD, the

ReplicaTAD implementation used in this work uses two types of processes: a single master process, and a pre-defined number of replica processes. Using this structure, the master process is effectively responsible for executing the serial TAD procedure (**FIG. 1 (a)**), with the modification that all MD-based work (steps 8 through 15) is off-loaded to replica processes. The procedure followed by the master can be framed as the recursive ReplicaTAD function defined in **FIG. 3**.

While the master process is responsible for executing most of the TAD procedure, the replica processes are used to discover transitions. The idea is very similar to the conventional PRD method. However, in the case of ReplicaTAD, a slightly different procedure is used to leverage the fact that the high-temperature MD is constrained to a single energy basin until the total sum of MD time (across all replicas) has reached t^{Stop}. In contrast to PRD, this means that a replica process does not need to stop after each transition is detected. Instead, the replica processes can follow the more efficient procedure detailed in **FIG. 3**. Note that, in order for the master process to deduce an accurate t^{High} for each transition, the replica processes must record both the WCT and the local MD time whenever an MD block is either initiated or finished. These pairs of values are stored in a history so that, given the WCT of a reference transition on a different process, the replica has a good estimate of the local MD time when that transition occurred.

A simplified diagram of the ReplicaTAD procedure is illustrated in **FIG. 2 (c)** for a single-transition trajectory with three replica processes. The diagram shows that the master process is responsible for all NEB calculations, meaning that the execution time for an ideal ReplicaTAD simulation with infinite replicas would be approximately equal to the time required to thermalize the system in each state and to calculate all the necessary saddle-point energies. This also implies that a ReplicaTAD simulation using only one replica process (and one master) will achieve better performance than Serial TAD, because the NEB and MD work will be performed concurrently. The

expected parallel speedup of the MD and transition checks for N_{rep} replica processes is given by

$$s_{rep} = \frac{1 + n_{quench} + n_{therm}}{(1 + n_{quench})/N_{rep} + n_{therm}}. \quad (5)$$

Here, n_{therm} represents the expected fractional number of thermalization force calls needed for every MD step, and n_{quench} represents the expected fractional number of transition-check force calls needed for every MD step. Since the master process will always be responsible for NEB calculations, the total ReplicaTAD run time can be approximated by

$$WCT_{total} = \left[max\left(\overline{w}_{NEB} N_{NEB}, \frac{\overline{w}_{stop} - \overline{w}_{NEB} N_{NEB}}{s_{rep}} \right) + \overline{w}_{therm} \right] N_{accept}. \quad (6)$$

Although ReplicaTAD modifies the traditional PRD procedure to avoid unnecessary synchronization costs, it is important to note that this asynchronous behavior requires special attention to implement. For example, there is no guarantee that the master process will receive transition messages from replica processes in the order that the transitions actually occurred. Therefore, it is possible for the master to reduce a value of t^{High}, and then subsequently receive a transition message corresponding to an even earlier high-temperature time. This can be a problem if the master decides that $t^{High} > t^{Stop}$, because it may still be possible for an out-of-order message to replace the winning transition. For this reason, it is necessary for a replica process to inform the master of any outstanding messages whenever it is asked to participate in a reduction operation.

Another important characteristic of ReplicaTAD is that the error can be very sensitive to the number of MD steps between sequential transition checks (the MD block size). When a replica process detects a transition, it defines the WCT of the transition by drawing a random number from a uniform distribution over the last MD block. Theoretically, this value should be drawn from an exponential distribution, but this would require a good estimate of the expected transition rate (which is rarely

known *a priori*). Since an exponential distribution is not used in practice, the definition of the transition time will always introduce some error that is proportional to the MD block size. In ReplicaTAD, this error is multiplied by the number of replica processes, because the transition WCT is used to reduce the total t^{High} across all replicas. Since frequent transitions can clearly reduce the efficiency of the simulation, we find that a more practical way to avoid error is to frequently record the geometry of the system over the course of a single MD block. This way, a more accurate transition time can be refined, without increasing n_{quench}.

One final point about the ReplicaTAD algorithm is that the initial thermalization stage must always be long enough to guarantee that the system is properly prepared at T^{High}. If the system is consistently cold when the thermalization stage is concluded, the observed transition rates will be artificially slow, and the errors will be multiplied by the total number of replica processes. We note that this error is easy to avoid by employing a conservative warm-up period for each new state. It is also important to make sure that the random thermal momenta for an initially minimized state correspond to twice the desired temperature. The common mistake of using T^{High}, rather than $2T^{High}$, will require the thermostat to gradually increase the temperature as the system approaches thermal equilibrium, because roughly half the initial kinetic energy will be converted into potential energy.

C. Hybrid Parallelism

In addition to pure speculation and pure replication, one can also employ a hybrid version of SpecTAD, in which each hybrid speculation process behaves as the master for its own local set of replica processes. The hybrid speculation process follows the *HybridSpecTAD* procedure detailed in **FIG. 4**; while the local replica processes still follow the procedure detailed in the **FIG. 3** inset. The ideal behavior results in each speculation process using many local replicas to reduce the time between NEB calculations. If most of the NEB calls are performed after each winning pathway is observed,

then many speculation processes can also be used to overlap the work between sequential states. Although a functioning hybrid SpecTAD code was implemented for this work, we will not present a parametric study of speculation and replication process-count combinations. Instead, we use the hybrid algorithm to demonstrate that neither SpecTAD nor ReplicaTAD is likely to be the optimal choice for a given system, but that a combination of the two will usually be best.

Since the hybrid SpecTAD method aims to combine the advantages of SpecTAD and ReplicaTAD, the theoretical run time is given by

$$WCT_{total} = \max\left(\frac{\overline{w}_{winner}}{\overline{w}_{stop}}, \frac{1}{N_{spec}}\right)\left[\max\left(\overline{w}_{NEB}N_{NEB}, \frac{\overline{w}_{stop} - \overline{w}_{NEB}N_{NEB}}{s_{rep}}\right) + \overline{w}_{therm}\right]N_{accept}, \quad (7)$$

where the replica speedup for each speculation process (s_{rep}) is still determined by **Eq. 5**. Since each speculation process will effectively own N_{rep} replica processes, the simulation will consume a total of $N_{spec}(N_{rep} + 1)$ processors. An important limitation of both ReplicaTAD and hybrid SpecTAD, as they are presented here, is that N_{rep} is always static. When the transition rates remain constant over the course of the simulation, then the optimal number of replica processes may also remain constant. For a very simple physical system, it may be possible to search the space of allowed T_{High}-N_{rep}-N_{spec} combinations to determine an optimal parameter set. However, in most realistic systems, the transition rate will vary from one state to the next, and so the parallel efficiency will also vary. To properly address this limitation, an adaptive resource-management procedure should be used to dynamically allocate how all computational resources are used. An optimal algorithm would most likely resemble that of the Parallel Trajectory Splicing (ParSplice) method [14]. Since an adaptive algorithm requires rather significant modifications to the hybrid algorithm presented here, its implementation has been left for future work.

IV. Parallel Performance

A. Simulation Details

All simulations presented in this paper correspond to a simple physical test system containing 301 silver atoms at room temperature (300K). The geometric configuration corresponds to an isolated adatom on an ideal (100) surface. This system was specifically chosen to match previous investigations of TAD performance [15] [16]. The interatomic potential corresponds to an embedded atom method (EAM) formulation [17], with the Voter parameterization for silver [18]. At room temperature, only two types of activated events are typically accepted during a TAD simulation: (1) An adatom hop with an energy barrier of 0.492 eV, and (2) an adatom exchange with an energy barrier of 0.586 eV. All remaining pathways are rarely accepted, as they correspond to much higher energy barriers (≥0.98 eV).

The presented results also use consistent TAD parameters for accepting transitions. That is, ν_{Min} and δ are always set to 5×10^{11} and 0.05, respectively. To ensure that transitions are resolved with sufficient accuracy, transition checks are performed every 500 time steps, with a system snapshot being recorded every 25 steps. Here, each step is 4 fs, meaning that the transition-time resolution is 0.1 ps. To guarantee that the system is properly prepared in each new state, the initial thermalization stage of a minimized geometry is performed for 1500 steps (6 ps). After a transition is detected, and a hot point is already available, the thermalization stage is only performed for 500 time steps (2 ps). We note that these block sizes were all chosen conservatively, since the intention was to maintain a constant set of algorithm parameters across a range of T^{High} settings and processor counts.

For the chosen physical system, the behavior of an actual TAD run is illustrated for two different high-temperature settings in **FIG. 5**. Here, the black blocks depict the first 20 accepted transitions, with the unique (ordered) ID of the transition on the x-axis, and the total WCT of the simulation on the y-axis. The blue, green and red blocks represent the same exact run, but with the

ideal scaling behavior for SpecTAD, ReplicaTAD and hybrid SpecTAD, respectively. For SpecTAD, this means that the next official state is explored as soon as the pathway leading to it is discovered. For ReplicaTAD, this means that the total time in each state can be approximated by the cost of NEB calculations, plus the initial thermalization stage. For hybrid SpecTAD, both the SpecTAD and ReplicaTAD assumptions hold. Note that the normal and hybrid SpecTAD blocks only represent the WCT needed to make the winning speculation for each state, while the WCT passed between speculating and accepting the winning transition is represented by a line.

B. Results

In order to compare the performance of SpecTAD and ReplicaTAD, both algorithms were used to simulate 1000 transitions (~8 ms) in the silver adatom system, using 1 through 118 processors, and a T^{High} of 600K through 900K. The results of the scaling study are summarized in **FIG. 6 (a)** & **(b)** for SpecTAD and ReplicaTAD, respectively. In both plots, the x-axis corresponds to the total number of processors used for the simulation, while the y-axis corresponds to the average WCT needed to accept each transition over the course of the many-transition simulation. The results are also summarized in **FIG. 6 (c) & (d)**, where the x-axes correspond to the high-temperature setting, and each curve represents the total processor count. Since the performance is measured in terms of an average run time, it is important to note that the error bars correspond to the standard error of the mean for the specific set of physical processors used to perform the long-timescale simulation. Across all measured runs, the average WCT needed to perform a single force call was approximately 0.72 ms.

For the real simulations considered here, every accepted pathway corresponded to an adatom hop or exchange event. This means that a theoretical run-time prediction can treat every official state as identical. In **FIG. 6**, the dashed-line curves represent simple theoretical predictions that employ this assumption, using **Eq. 4 & 6**. All theoretical scaling curves were produced using the itemized force

call statistics summarized in **TABLE I**. These statistics, which were obtained using serial runs at each of the high temperatures of interest, make it clear that the performance of SpecTAD becomes much better than that of ReplicaTAD when T_{High} = 900K. At this temperature, each winning transition will be discovered (and its corresponding NEB performed) after completing about 5.33% of the total force calls needed to reach t^{Stop}. This implies that the SpecTAD procedure will perform roughly 94.7% of the total work in parallel. ReplicaTAD, on the other hand, is strongly limited by the serial execution of NEB calculations, and can be expected to accelerate less than 87% of the work at 900K. At T^{High} = 600K, the story is very different, because the NEB calculations account for less than 1% of the total simulation time in serial, while speculation can only hope to perform about 90% of the work in parallel, at best.

The measured performance results, shown in **FIG. 6**, agree reasonably well with **Eq. 4-7**. However, since the models do not consider communication and synchronization costs, there are some qualitative discrepancies. The most important discrepancy is that the theoretical model for ReplicaTAD ignores master-replica synchronization delays, which leads to a negative error in the run time prediction, an error that is proportional to the number of processes. It is likely that much of this synchronization time can be avoided by implementing a more efficient communication strategy, but it is important to recognize that the theoretical model leads to optimistic predictions. For SpecTAD, there is less communication between processes, although the real performance is clearly worse than the theoretical prediction when SpecTAD is performed on 22 cores at $T^{High} \geq$700K. For these cases, SpecTAD has a processor count that should be close to the expected scaling limit (note that the theoretical predictions are all flat above 22 cores). Therefore, the performance reduction observed in these runs is most likely due to a delay whenever a speculation process is reassigned to a new state. When there are only a small number of processors available, it is unlikely that a speculation process will be interrupted with a reassignment message. This is because there is already a delay between the

time when a pathway is observed and the time when a speculation process becomes available to explore the corresponding state. When there are too many processors available, there are also fewer reassignment penalties, because there is always a speculation process available when a new pathway is observed.

In addition to pure speculation and replication, **FIG. 6** also shows some results for the hybrid SpecTAD approach detailed in **FIG. 4**. These results demonstrate that a given SpecTAD or ReplicaTAD simulation can often be improved by using the same number of total cores to perform a combination of speculation and replication. At T^{High} = 900K, in **FIG. 6 (a)**, a hybrid SpecTAD simulation was performed using N_{rep} = 2 replica processes and N_{spec} = 1-58 speculation processes. At T^{High} =700K, in **FIG. 6 (b)**, a hybrid SpecTAD simulation was performed using N_{rep} = 23 replica processes and N_{spec} =1-5 speculation processes. Across a range of T^{High} settings, in **FIG. 6 (d)**, a hybrid SpecTAD simulation was performed using N_{rep} = 11 replica processes and N_{spec} = 10 speculation processes. In all three cases, the hybrid simulation led to a clear improvement in performance relative to the best-performing pure algorithm. Some of these hybrid results are also shown as circular markers in **FIG. 7 (a)**, alongside (dashed line) theoretical predictions and (solid line) pure SpecTAD and ReplicaTAD results.

The disagreement between the real and estimated results for the 700K hybrid simulations show, once again, that our simple theoretical model does not capture the costs of communication and synchronization. With this limitation in mind, the theoretical model can still be used to perform an approximate sweep of various combinations of T_{High}, N_{rep} and N_{spec}. Several of the performance curves generated by such a sweep are shown in **FIG. 7 (b)**. With the total number of processes being limited to 108, the model tells us that the optimal combination will be N_{rep} = 6 replica processes and N_{spec} =18 speculation processes at T_{High} =800K. While this combination does lead to a better performance than

pure speculation or replication, the average WCT to accept each state is almost a factor of two greater than the theoretical estimation. Note that this real data point is shown as a red circle in **FIG. 7 (b).**

V. Discussion

The results presented in this work provide significant motivation for the use of parallelism in TAD. To fully appreciate the advantages of both TAD and its parallel extensions, it is useful to recognize that direct MD would take approximately four weeks to reach the average time required for a single silver adatom transition at room temperature. Using serial TAD with T_{High} = 900K, the first transition is accepted in approximately four minutes, corresponding to an approximate boost factor of 5×10^3. With speculation introduced, this boost factor further improves to 1×10^5, meaning that the average transition is accepted in less than twelve seconds.

Although the performance of SpecTAD is certainly impressive, the most crucial result of this work is that a mixture of speculation and replication can be used to improve the boost even further. The benefit of trading speculation processes for replication processes becomes even more pronounced as the T_{High} setting is reduced. This is very useful in practice, because the behavior of most physical systems tends to depart from HTST at elevated temperatures. For the silver adatom system used here, it is known that the average TAD transition rate will be faster than the HTST prediction whenever T_{High} exceeds ~500K [15]. As the T_{High} setting is varied from 600K to 900K, the error is expected to vary from approximately 10% to 30%. Therefore, if the goal of this work was to accurately calculate the adatom diffusion rate at room temperature, we would probably want to choose T_{High} < 600K. In this case, a speculation-only approach would perform significantly worse than a replication-only approach (as suggested by the dashed black curves in **FIG. 6 (a) & (b)**). Furthermore, the theoretical and measured performance of hybrid SpecTAD tell us that a combination of speculation and replication is likely to get us to the same boost obtainable at T_{High} = 900K, assuming that sufficient cpu resources are

available.

The fact that temperature-acceleration can be traded for parallel processing is an important and exciting result. It tells us that, to a point, we can alternatively think of a ReplicaTAD simulation as temperature-accelerated PRD. That is, we can start with a PRD simulation in mind ($T_{High} = T_{Low}$), and then we can adjust T_{High} to achieve an optimal balance of acceleration and accuracy. When temperature acceleration is introduced to a PRD simulation in this way, it is likely that the parallel efficiency of the replica processes will be reduced. Although this means that ReplicaTAD will not scale as well as traditional PRD, much of the lost efficiency can be recaptured by the hybrid SpecTAD method. Hybrid parallelism can lead to significantly better scaling, because the replica processes become more efficient when they are split between a number of speculative states.

In order to fully appreciate this last point about hybrid parallelism, it is useful to compare the results plotted in **FIG. 6 (c)** & **(d)**. These results suggest that ReplicaTAD is more scalable than SpecTAD when T_{High} is less than 750K, while the opposite is true when T_{High} is greater than 750K. However, since we know (from **FIG. 6 (a) & (b)**) that both replica and speculation processes will be consistently efficient when there are only ~10 processes, we can expect a combination of 11 replicas and 10 speculations to perform well across all T_{High} settings of interest. In **FIG. 6 (d),** real performance results for this scenario are shown. These results not only confirm that an intuitive combination of replica and speculation processes is effective, but they also show that the combination can yield performance well-beyond the reach of either pure parallelism approach.

Keeping in mind the obvious advantages of hybrid SpecTAD over the other parallel TAD methods, there are still a few reasons why further algorithm development is motivated by this work. As already mentioned, for realistic systems that exhibit significant state-to-state variation in their activated transition rates, the static-replica approach (used in this work) is likely to limit the parallel

scaling. Additionally, the current master-replica synchronization procedure requires conservative parameter choices to avoid multiplicative errors in the total transition rate. With these limitations in mind, the investigation presented here makes it clear that the future development of an adaptive parallel-TAD approach must continue to combine the advantages of both speculation and replication.

Since the silver adatom system is dominated by only two transition pathways, with similar energy barriers (0.492 eV and 0.586 eV), it is also important to note that the presented comparison between speculation and replication is somewhat limited. In general, the optimal allocation of parallel resources is strongly dependent on the specific distribution of possible transition rates. This is because these rates ultimately determine the $\overline{w}_{\text{Stop}}/\overline{w}_{\text{winner}}$ ratio, which in turn corresponds to the efficiency of parallel speculation. For the silver adatom system, the large disparity in energy between the dominant mechanisms, and the more-complex pathways (~0.6 eV vs. ~1.0+ eV), means that early speculations are especially likely to correspond to *winning* transitions.

What the simplicity of the chosen physical test system ultimately means for the analysis presented above, is that the use of replication is likely to become more advantageous than speculation when the system becomes more complex. For systems with transition rates that are strongly dependent on the current state of the official trajectory low-energy flickers can certainly degrade the efficiency of pure replication. However, it is also much easier to guarantee that all processes will produce work within an officially visited state. Therefore, in practice, a reasonable approach to the simulation of complex systems is to begin with a replication-heavy allocation of processes, and to gradually add speculative character. Since the optimal allocation of resources depends strongly on the exact physical system, this recommended approach represents yet another reason why an adaptive procedure would be valuable.

VI. Conclusion

In summary, we have demonstrated that the speculative approach to TAD can often be improved by incorporating replication-based parallelism. In order to illustrate this point, we have compared three parallel TAD methods: SpecTAD, ReplicaTAD, and hybrid SpecTAD. After reviewing the basics of serial TAD, we discussed the behavior of each parallel algorithm in detail, and presented theoretical predictions for their performance. Using a combination of theoretical models and simulation results, we have investigated the capabilities and limitations of each method. For a given number of cpu resources, the results clearly demonstrate that hybrid SpecTAD will outperform both SpecTAD and ReplicaTAD for some N_{rep}-N_{spec} combination. Therefore, our primary conclusion is that optimal performance will typically call for hybrid parallelism.

VII. Acknowledgements

This work was supported by the United States Department of Energy (U.S. DOE), Office of Science, Office of Basic Energy Sciences, Materials Sciences and Engineering Division. LANL is operated by Los Alamos National Security, LLC, for the National Nuclear Security Administration of the U.S. DOE, under contract DE-AC52- O6NA25396.

VIII. References

1. K. Kadau, T. C. Germann and P. S. Lomdahl: Molecular Dynamics Comes of Age:. 320 Billion Atom Simulation on BlueGene/L. *Int. J. Mod. Phys. C* **17**, 1755-1761, (2006).
2. T. C. Germann and K. Kadau: Trillion-atom molecular dynamics becomes a reality. *Int. J. Mod. Phys. C* **8** 1315-1319 (2008).
3. A. F. Voter: A method for accelerating the molecular dynamics simulation of infrequent events. *J. Chem. Phys.* **106**(11) 4665–4677 (1997).
4. A. F. Voter: Parallel replica method for dynamics of infrequent events. *Phys. Rev. B* **57**(22) R13985 (1998).
5. D. Perez, B. P. Uberuaga and A. F. Voter: The parallel replica dynamics method coming of age. *Comp. Mater. Sci.* **100**(Part B) 90-103 (2015).
6. M. R. Sorensen and A. F. Voter: Temperature-accelerated dynamics for simulation of infrequent events. *J. Chem. Phys.* **112**(1) 9599-9606 (2000).
7. G. H. Vineyard: Frequency factors and isotope effects in solid state rate processes. *J. Phys. Chem. Solids* **3**(1) 121-127 (1957).
8. R. J. Zamora, B. P. Uberuaga, D. Perez and A. F. Voter: The modern temperature-accelerated dynamics approach. *Annu. Rev. Chem. Biomol. Eng.* 7(1) 3.1-3.24 (2016).
9. R. J. Zamora, A. F. Voter, R. Perriot, D. Perez and B. P. Uberuaga: The effects of cation-anion clustering on defect migration in MgAl2O4. *Phys. Chem. Chem. Phys.* **18** 19647–19654 (2016).

10. M. P. Allen and D. J. Tildesley: Computer Simulation of Liquids. New York, NY: Clarendon Press (1989).

11. H. Jonsson, G. Mills and K. W. Jacobsen: Nudged elastic band method for finding minimum energy paths of transition. in *Classical and Quantum Dynamics in Condensed Phase Simulations* (1997).

12. G. Henkelman, B. P. Uberuaga and H. Jónsson: A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *J. Chem. Phys.* **113** 9901 (2000).

13. S. Plimpton: Fast parallel algorithms for short-range molecular dynamics. *J. Comp. Phys.,* **117**(1) 1-19 (1995).

14. D. Perez, E. D. Cubuk, A. Waterland and E. Kaxiras: Long-time dynamics through parallel trajectory splicing. *J. Chem. Theory Comput.* **12**(1) 18-28 (2016).

15. S. M. Mniszewski, C. Junghans, A. F. Voter, D. Perez and S. J. Eidenbenz: TADSim: Discrete event-based performance prediction for temperature-accelerated dynamics. *ACM Trans. Model. Comput. Simul.* **25**(3) 15:1–15:26 (2015).

16. R. J. Zamora, A. F. Voter, D. Perez, N. Santhi, S. M. Mniszewski, S. Thulasidasan and S. J. Eidenbenz: Discrete event performance prediction of speculatively parallel temperature-accelerated dynamics. *Simulation* **92**(12) 1065–1086 (2016).

17. M. S. Daw and M. I. Baskes: Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Phys. Rev. B* **29** 6443–6453 (1984).

18. A. F. Voter: Simulation of the layer-growth dynamics in silver films: Dynamics of adatom and vacancy clusters on ag (100). in *31st Annual Technical Symposium of the International Society for Optics and Photonics* (1988).

IX. Figures

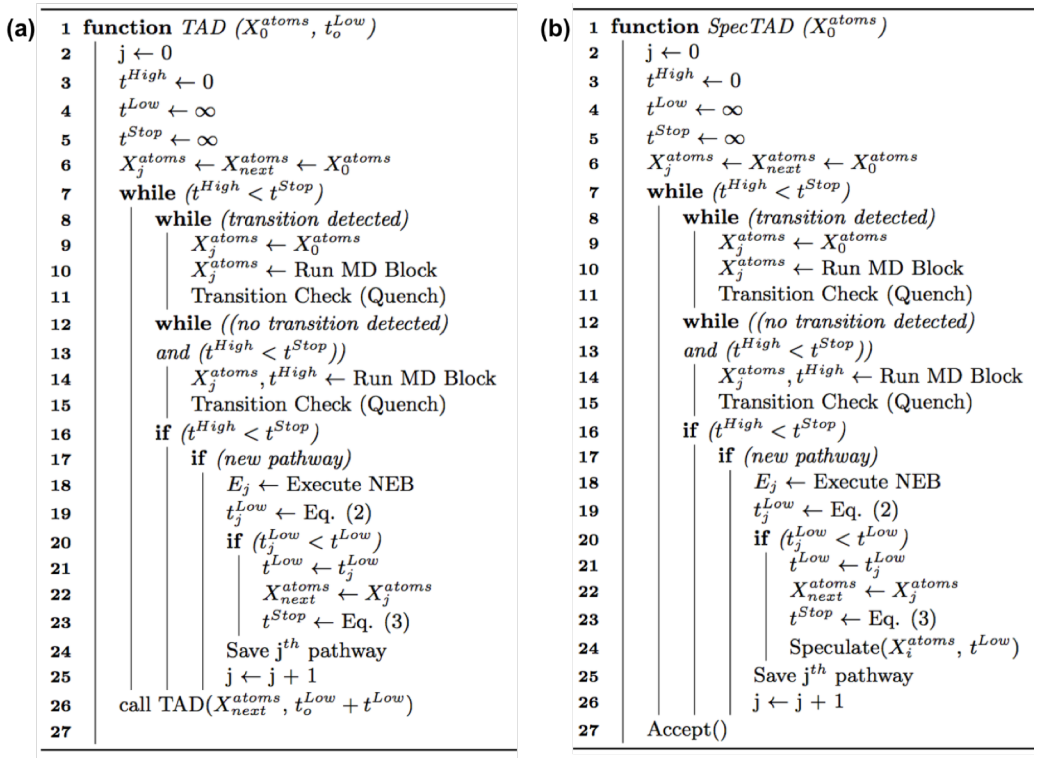

FIG. 1. (a) Simple pseudocode description of the TAD algorithm, formulated as a recursive function. (b) The algorithm to be executed by each speculation process in SpecTAD. These processes must wait until an initial state (X_i^{atoms}) is provided by the master process.

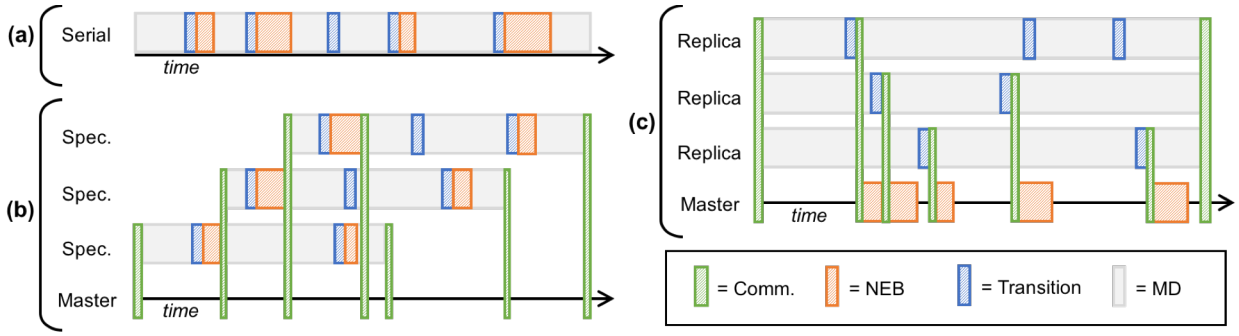

FIG. 2. The hypothetical run time behaviors of (a) TAD, (b) SpecTAD, and (c) ReplicaTAD. The diagrams are intended to be educational aids, and are not drawn to scale. In all diagrams: The solid grey rectangles depict many separate blocks of basin-constrained MD, with transition-check quenches not shown between each block. Blue rectangles depict an MD block, and corresponding quench, during which a transition was detected. Orange blocks depict the NEB calculations for new pathways. Green blocks depict communication.

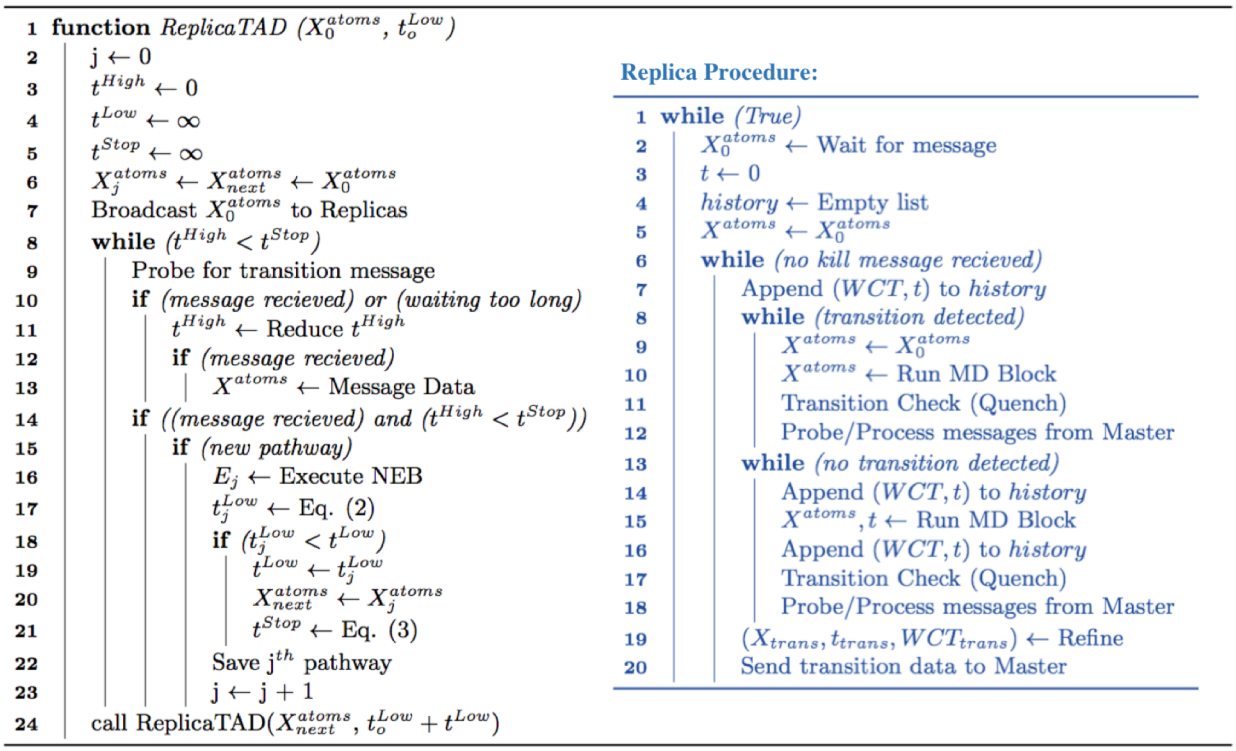

FIG. 3. The recursive *ReplicaTAD* function executed by the master process in the ReplicaTAD method. Note that the master process must communicate with the replica processes. The inset (blue) shows the procedure executed by each replica process.

```
1  function HybridSpecTAD (X_0^atoms)
2      j ← 0
3      t^High ← 0
4      t^Low ← ∞
5      t^Stop ← ∞
6      X_j^atoms ← X_next^atoms ← X_0^atoms
7      Broadcast X_0^atoms to Replicas
8      while (t^High < t^Stop)
9          Probe for transition message
10         if (message recieved) or (waiting too long)
11             t^High ← Reduce t^High
12             if (message recieved)
13                 X^atoms ← Message Data
14         if ((message recieved) and (t^High < t^Stop))
15             if (new pathway)
16                 E_j ← Execute NEB
17                 t_j^Low ← Eq. (2)
18                 if (t_j^Low < t^Low)
19                     t^Low ← t_j^Low
20                     X_next^atoms ← X_j^atoms
21                     t^Stop ← Eq. (3)
22                     Speculate(X_i^atoms, t^Low)
23                 Save j^th pathway
24                 j ← j + 1
25     Accept()
```

FIG. 4. The function executed by the master process within the hybrid SpecTAD method. Note that the replica processes still execute the procedure detailed in the inset algorithm in **FIG. 3**.

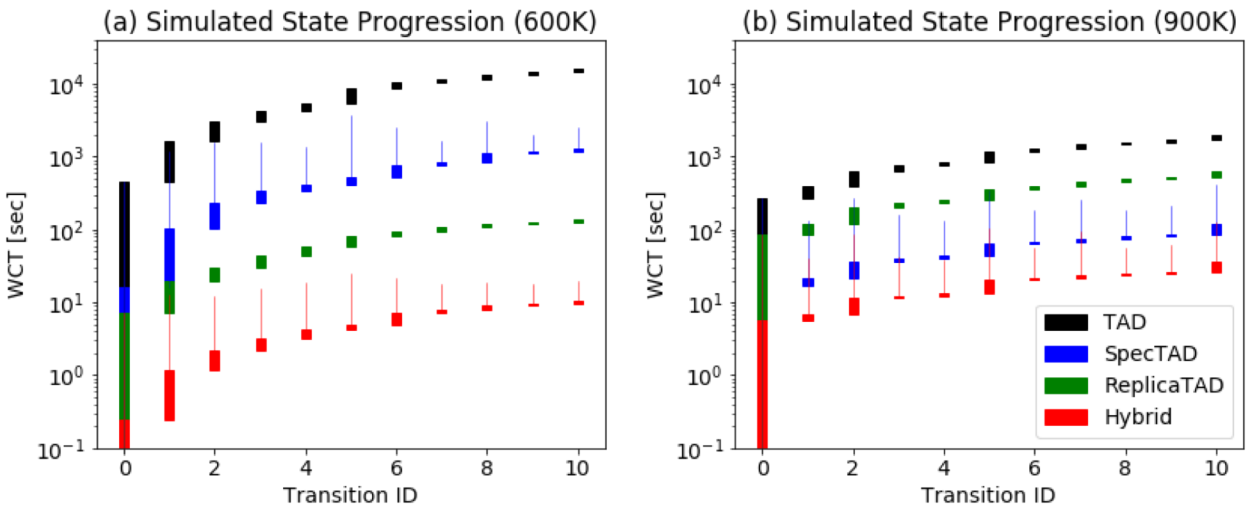

FIG. 5. The ideal parallel execution of states using SpecTAD, ReplicaTAD, and hybrid SpecTAD at high temperatures of (a) 600K and (b) 900K. The TAD curve corresponds to the performance of a real serial simulation, while all other curves correspond to the hypothetical scaling of the same trajectory. For serial TAD (black), and ReplicaTAD (green), each block represents the time needed to accept a single transition. For normal SpecTAD (blue) and hybrid SpecTAD (red), each block represents the time needed to find the winning transition, and the line represents the remaining time needed to reach the ultimate stop time.

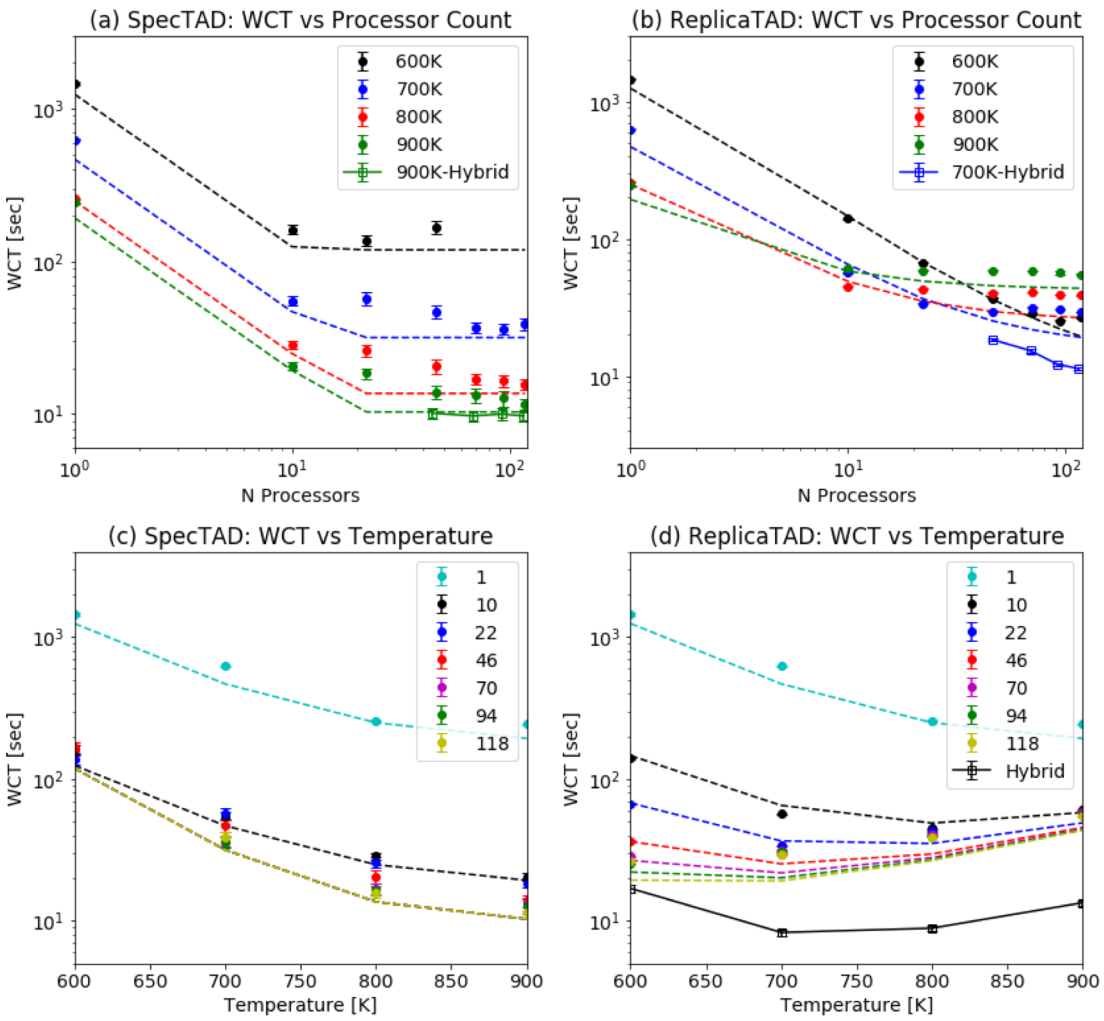

FIG. 6. Summary of performance results for (a) & (c) SpecTAD and (b) & (d) ReplicaTAD. In (a) and (b), the x-axes correspond to the number of processes used for the simulation, while in (c) and (d), the x-axes correspond to the high-temperature setting. In all plots, the y-axis corresponds to the average WCT needed to accept each of 1000 transitions (~8 ms) in the silver adatom system. The circular data points represent real simulations, while the dashed lines represent ideal (communication-free) theoretical predictions using Eq. 4 & 6. In (a), the solid green line represents real hybrid simulations with 2 replica processes belonging to every speculation process. In (b), the solid blue line represents real hybrid simulations with 23 replica processes belonging to each speculation process. In (d), the solid black line represents real hybrid simulations with 11 replica processes and 10 speculation processes.

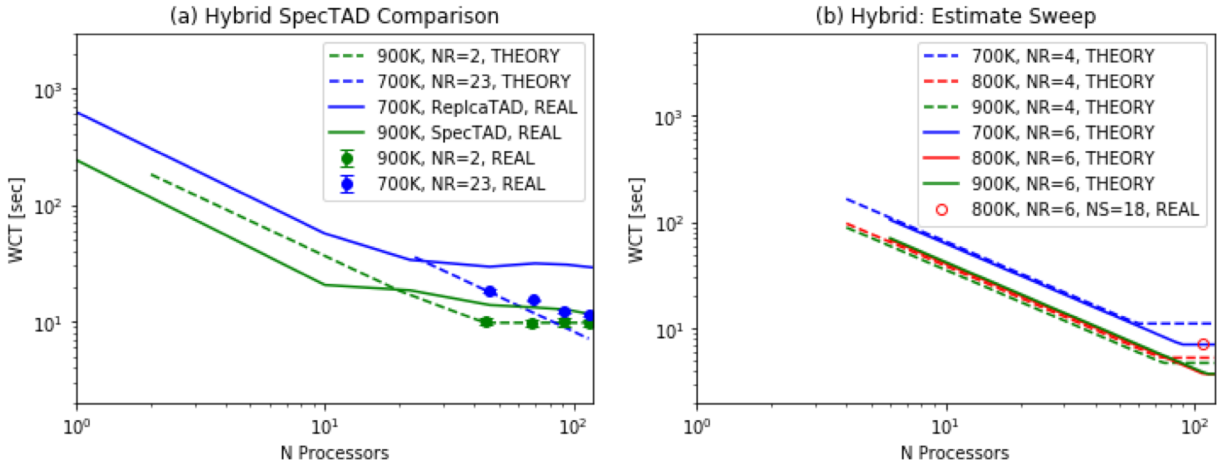

FIG. 7. (a) A comparison of SpecTAD and ReplicaTAD results (solid lines) with selected hybrid SpecTAD results (circular markers). Dashed lines correspond to the theoretical prediction from equation 7. (b) Results of a theoretical sweep over various high-temperatures and N_{rep}-N_{spec} combinations. The open circular marker represents to a real hybrid simulation using the parameters corresponding to the optimal theoretical prediction.

X. Tables

T_{High} K	Winner %	Total	MD %	NEB %	Quench %	Therm %	Warmup %	N_{NEB}	N_{trans}
600	9.53	1.74E+06	93.0	0.51	5.95	0.45	0.09	6.37	14.5
700	6.76	6.54E+05	87.3	2.18	7.31	2.12	0.25	8.66	25.0
800	5.47	3.48E+05	76.3	6.22	8.20	6.51	0.57	11.2	37.4
900	5.33	2.70E+05	58.1	12.9	8.85	14.6	1.03	15.2	54.5

TABLE I. Breakdown of the average force-call counts per state visit. Total corresponds to the average total force calls needed for each state visit. N_{NEB} and N_{trans} correspond to the average number of saddle-point searches and transitions detected, respectively.