



Fermi National Accelerator Laboratory

FERMILAB-Pub-93/365

Effects of Automated Transfer Coalescing on Production Physics

Mark Fischler

*Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510*

November 1993

To be published in *Nuclear Physics B Proc. Supp: Proceedings of Lattice '93*

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Effects of Automated Transfer Coalescing on Production Physics

Mark Fischler ^{a *}

^aFermi National Accelerator Laboratory
Batavia, IL 60510 USA

Diverse lattice gauge applications coded using the Canopy paradigm (augmented by transfer coalescing techniques) have been making effective use of the 50 Gflop ACPMAPS system at Fermilab for heavy-quark physics. These investigations have uncovered necessities and limitations beyond the usual CPU-Memory-Communications triad. Mass storage bandwidth and system robustness requirements are discussed in light of this experience.

1. ACPMAPS

The 50 Gflop ACPMAPS [1] system at Fermilab has been in full physics production since April, 1993. ACPMAPS is a distributed memory, MIMD system. It comprises over 600 i860 CPU's, with 32 Mbytes of memory for each. Each processor node has "flat global" access to the memory of any other node.

The system has features which facilitate rapid development of physics code, and allow for flexible investigations. The Canopy [2] software underpinning provides a library of concepts applicable to problems on the grid, allowing scientists to quickly code complicated algorithms in a clear and modular fashion. ACPMAPS is a shared system; applications can co-exist with each using an arbitrary set of nodes—this enables multiple simultaneous "streams" of physics investigation. And there is a large, high-bandwidth parallel I/O subsystem with close to 50 Gbytes of distributed disk space and 32 helical scan tape drives, for storage and re-use of configurations, propagators, and other intermediate results.

The Canopy paradigm breaks up an algorithm at the level of work done on a single site. This assumes that communications overheads are not large, since the typical block of data transferred is small (e.g. one SU(3) matrix). The ACPMAPS internode communication is designed for low latency. However, the nature of the i860

forces both CPU's to participate in data transfers; this imposes a software overhead including the time required for the target node to respond to an interrupt. In order for the system to run with reasonable efficiency, the Canopy software was augmented with techniques to coalesce many transfers between the same two nodes. These techniques[3], which involve light-weight context switching when off-node data is required, are transparent to the user.

2. Communications—Transfer Coalescing

The effective cost of an off-node read (or write) communication on ACPMAPS ranges from 25–60 μ sec. If each transfer needed to process work for each site were done separately, these overheads would impact typical Canopy applications by a factor of 2–6, relative to single-node performance. To coalesce transfers, the computation for each site is considered a separate *thread* of activity. When off-node data is needed, the processor can switch to another thread; later, blocks requested by many sites may be fetched in a single transfer. The "degree of coalescing" varies with number of threads allowed (N_θ), the distribution and ordering of sites, and locality of the algorithm. The observed behavior for a wide range of applications is coalescing up to $.8N_\theta$ transfers for local algorithms (e.g. link updating) and $.15N_\theta$ for non-local (but patterned) algorithms such as FFT. Since the only per-thread cost is local stack memory for each one (8K bytes is adequate), this degree of coalescing can always be made to be

^{*}Fermilab is operated by Universities Research Association, Inc. under contract with the U.S. Department of Energy

10 or more—communications overheads no longer dominate the time taken.

With coalescing enabled, applications get linear speedup with number of nodes (this is important for automated allocation of resources to each job). The efficiency is 40–80% of single-node performance; this provides enough computational power (6–12 effective Gflops) that physics productivity is often limited by I/O and other issues. An important exceptional algorithm is the incomplete LU preconditioned propagator inversion, which runs extremely poorly without coalescing transfers. Due to complex tradeoffs between nodes waiting for valid data and desire to combine transfers, this runs at 25% efficiency with coalescing enabled. Nonetheless, since it converges in up to 10 times fewer sweeps than competing methods, it is worth using.

3. Physics Usage

These proceedings include results [4–6] of recent calculations done on the upgraded ACPMAPS. This work extends the earlier studies which focus on extracting measurements in systems involving at least one heavy quark [7,8]. The system has not yet been used for serious calculations with dynamic fermions; the observations below apply to quenched physics.

Physicists have used ACPMAPS (including the 5 Gflop system) for 4 years, running hundreds of distinct applications; we can assess the advantages of its MIMD architecture with low latency flat global communication. Only one algorithm which might truly require this full flexibility has been run extensively: Minimum Residual LU-preconditioned inversion. However, this is quite an important algorithm, saving a factor of 2–3 on computations which would otherwise consume more than half the system time. Other routines including various FFT methods require flexibility for practical development, although it might be possible to code specific instances for lockstep systems. Probably the most important value of flexibility is that it allows for the Canopy paradigm, which in turn supports a wide variety of measurement programs and explorations of improved actions and other physical concepts. Fi-

nally, the flexible architecture allows for smooth system sharing, which is valuable when program development must co-exist with production running.

Other distinguishing features of ACPMAPS include its large memory [9] and powerful I/O subsystem. The benefits of large memory are obvious—lattices significantly larger than $32^3 \times 48$ can be studied if necessary, and algorithms can often gain speed if copious memory is available.

Applications fall into two categories: CPU-bound jobs (gauge configuration generation, gauge fixing, and propagator computation each consume roughly equal resources); and smaller I/O-bound jobs, which extract physics measurements from the configurations and propagators. The new more powerful system runs CPU-bound jobs more quickly, so diverse, I/O-intensive job streams are now more prevalent. This stresses features other than CPU power and internode communication, exposing I/O-related requirements for the next physics steps.

4. Current and Future Limits

Among the usual system productivity factors—CPU power, memory, and communications—ACPMAPS is now balanced between communications and CPU limitations. Yet other issues concerning mass storage and automated running are emerging as equally important in determining how fast physics progresses.

4.1. Mass Storage

Configurations and propagators are being produced at a rate of 5 Tbytes of archived data per year. This field data is later used for several streams of analysis, including construction of wave functions and measurement of quantities based on smeared operators. This allows feedback of wavefunctions into other measurements, and iterative improvement of measurement methods. Although this mode of investigation—involving frequent passes through saved field data—is valuable, it is not practical on other supercomputing systems with less emphasis on flexibility and mass storage.

The I/O subsystem supports these needs by a

combination of parallel disk and tape systems. Currently, disk space is used as a staging area for data coming from tape, so a short job requiring a large memory space does not occupy many nodes during the (slow) tape input.

The performance of the I/O subsystem determines how fast measurements can be extracted from field configurations. To permit rapid transfers to memory, field data is striped over several disks. Archived data is typically striped onto 4–6 tape “sets” to shorten the tape read-in time. Most measurements involve reading one configuration and propagators for 1–3 mass values. Typical field files have been re-used about 20 times in the course of various analyses.

Disk space limits the number of fields that can be staged simultaneously; and the aggregate tape bandwidth is 8 Mbytes/sec. This induces a practical limit on the number of analysis streams that can co-exist: If too many streams of analysis require different configurations, then tapes will be dismantled before all their fields have been staged to disk—costly “tape thrashing” occurs. (The alternative of having job streams reserve tape drives while doing computation is even more wasteful.) Currently, this limits us to two main analysis production streams (requiring an average of 20 tape set mounts per day) co-existing with CPU-bound jobs.

By comparing node usage during periods when the system is used extensively for analysis, to periods when only large propagator calculations are run, we can determine that tape I/O limitations impact the analysis by at least a factor of 2. We estimate that to take full advantage of the CPU power without distorting scientists analysis efforts would require a bandwidth to automated archival tape of at least 15 Mbytes/sec, given 50–100 Gbytes of disk space.

4.2. Automated Job Streams

At any given time, users have several ongoing efforts scanning hundreds of field files. And typical investigations involve complex sequences of generating, gauge fixing, and examining configurations and propagators. It is impractical to directly supervise each job; instead, scripts are created to supervise the staging of data and initi-

ation of jobs. This puts tremendous emphasis on system robustness, since any exception which is not well understood will abort a stream of physics until a scientist can determine what went wrong and how to recover.

Certain well-controlled exceptions can be tolerated. For example, ACPMAPS suffers about .5 job-aborting parity errors per day; this rate is determined by DRAM usage, which averages 8 Gbytes at any instant. There is a tool provided to automatically restart those jobs, and the impact on total production is 1–5%. This is acceptable, though it indicates that larger systems will require error-correcting memory designs.

Other more complex errors cause substantially more difficulty, and cannot be tolerated at appreciable frequencies. At the extreme are “error not caught” conditions: Unless there is confidence that these do not occur, a prudent investigator must duplicate jobs for which internal consistency checks are unavailable. A single error of this sort can easily halve the effective power of a system.

REFERENCES

1. *ACPMAPS—A Detailed Overview*, M. Fischler, FERMILAB-TM-1780 (1992)
2. *Canopy 7.0 Manual*, M. Fischler, G. Hockney, M. Uchima, P. Mackenzie, available from the Fermilab Computing Division
3. M. Fischler, M. Gao, G. Hockney, M. Isely, M. Uchima, Nucl. Phys. B30 (Proc. Supp), 301 (1993)
4. E. Eichten, B. Hill, and H. Thacker, these proceedings.
5. G. Hockney, these proceedings
6. P. Mackenzie, these proceedings
7. A. El-Khadra, G. Hockney, A. Kronfeld, P. Mackenzie, Phys. Rev. Letters 69(1992) 729
8. A. Duncan, E. Eichten, A. El-Khadra, J. Flynn, B. Hill, H. Thacker, Nucl. Phys. B30 (Proc. Supp), 301 (1993)
9. The review by Y. Iwasaki (these proceedings) compares memory size, relative to power, for various high-end lattice systems.