CONF-951026--11

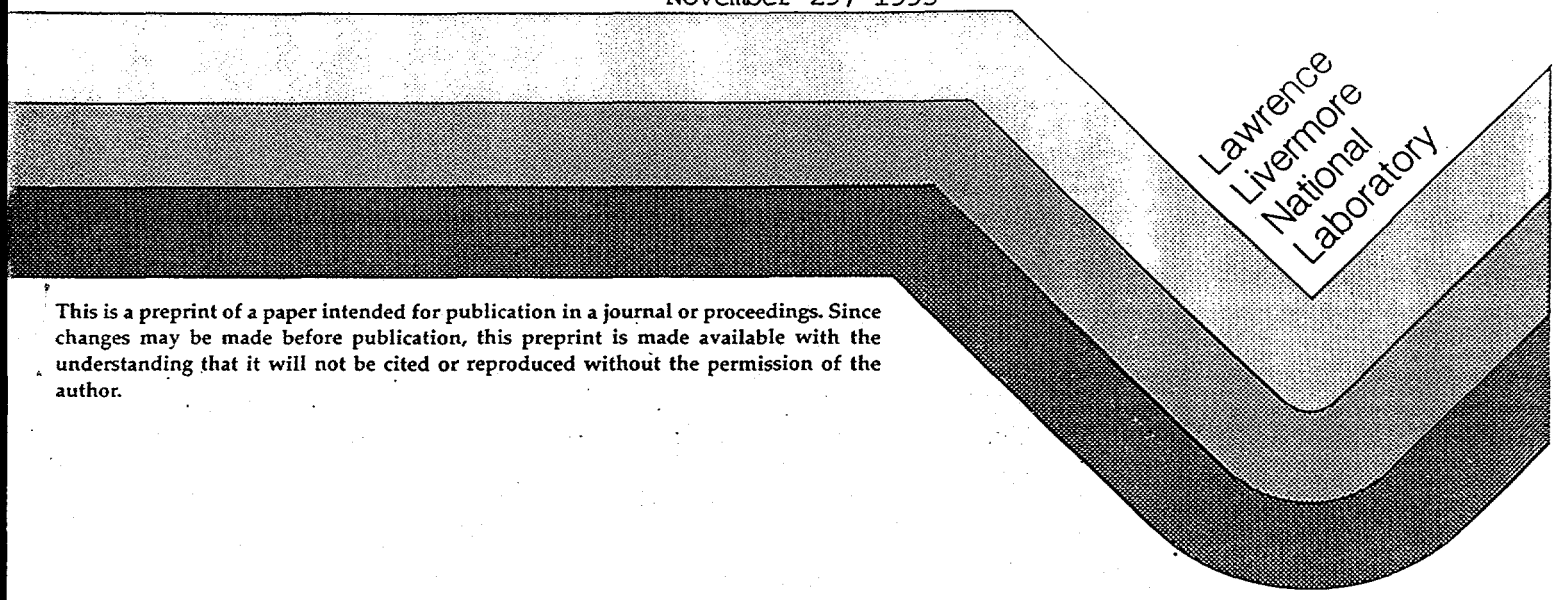
UCRL-JC-122988
PREPRINT

Enhanced Tensile Ductility in Al-Mg Alloys by Solid-Solution Interactions

Eric M. Taleff
Gregory A. Henshall
Donald R. Lesuer
T. G. Nieh
Jeffrey Wadsworth

This paper was prepared for submittal to
Aluminum and Magnesium for Automotive
Applications, Materials Week 1995
Cleveland, OH
October 29-November 2, 1995

November 29, 1995



Enhanced Tensile Ductility in Al-Mg Alloys by Solid-Solution Interactions

Eric M. Taleff [1], Gregory A. Henshall [2], Donald R. Lesuer [3], T. G. Nieh [2], and Jeffrey Wadsworth [2]

1. The University of Texas at Austin, Department of Aerospace Engineering and Engineering Mechanics, Austin, TX 78712.
2. Lawrence Livermore National Laboratory, Chemistry and Materials Science Department, Livermore, CA 94550.
3. Lawrence Livermore National Laboratory, Manufacturing and Materials Engineering Division, Livermore, CA 94550.

Abstract

The development of methods for obtaining high tensile elongation in aluminum alloys is of great importance for the practical forming of near-net-shape parts. Current superplastic aluminum alloys are limited in use by high material costs. The utilization of solute-drag creep processes, the approach used in this study, to obtain enhanced tensile ductility in aluminum alloys has lead to tensile elongations of up to 325% in simple, binary Al–Mg alloys with coarse grain sizes. This method has the advantage of lowering processing costs in comparison with superplastic alloys because a fine grain size is not necessary. Whereas superplastic alloys typically have a strain-rate sensitivity of $m = 0.5$, the enhanced ductility Al–Mg alloys typically exhibit $m = 0.3$ where maximum ductility is observed. Although a strain-rate sensitivity of $m = 0.5$ can lead to elongations of over 1000% (superplastic materials) a value of $m = 0.3$ is shown experimentally to be sufficient for obtaining elongations of 150% to a maximum observed of 325%. Enhanced ductility is also affected strongly by ternary alloying additions, such as Mn, for which a preliminary understanding is pursued.

Presented at Materials Week 1995, Cleveland, OH
Aluminum and Magnesium for Automotive Applications
Edited by J. D. Bryant, P. A. A. Khan, A. M. Sherman, and D. R. White
The Minerals, Metals & Materials Society, 1995

Introduction

Superplastic materials have been successfully applied to the near-net-shape forming of complicated structural components for the aerospace industry [1]. Superplastic forming of lightweight aluminum and titanium materials has lead to significant decreases in manufacturing costs and weight while increasing overall part performance [1]. These successes come from the ability of superplastic materials to be formed into a complicated, near-net shape in a single forming operation, thus reducing the need for multiple stamping operations and machining. Superplastic materials exhibit such ductility without necking or rupture because of the high strain-rate sensitivity value, $m = 0.5$, inherent to grain-boundary-sliding creep [1]. The grain-boundary-sliding creep mechanism generally requires a stable grain size of less than 10 μm [2]. Because of the special alloying and processing necessary to create such a fine, stable grain size, superplastic materials are generally quite expensive. Recent studies, however, have indicated the possibility of conducting similar forming processes using *enhanced ductility* materials, which would cost significantly less than current superplastic materials [3-6]. Enhanced ductility materials can exhibit tensile elongations of over 300% due to a solute-drag-controlled dislocation creep process, which provides a high strain-rate sensitivity of $m = 0.3$ [4, 5, 7]. For most industrial forming operations tensile elongations of 100% are sufficient. Because enhanced ductility materials rely solely on the solute-drag effect, they do not require expensive processing and grain-refining alloy additions. Enhanced ductility, which has been demonstrated in Al–Mg alloys [4-7], could provide a method of producing near-net-shape parts much more economically than with superplastic aluminum materials.

Data have shown that enhanced ductility occurs in the solute-drag creep regime, where a high strain-rate sensitivity of $m = 0.3$ significantly reduces the rate of neck growth [4, 5]. Such investigations have also defined several areas which must be explored if enhanced-ductility materials are to reach practical application [4-6]. The present study will address two of these areas: the effect of grain size, and the effect of ternary alloy additions on enhanced ductility. Four Al–Mg materials are studied in the present investigation, two of binary composition and two with ternary alloying additions of Mn. Both elongation-to-failure and strain-rate-change tests were conducted. The effects of grain size and ternary additions of Mn to Al–Mg materials are investigated using the data obtained from these tests.

Experimental Procedure

A low-impurity, binary alloy containing 2.8 wt pct Mg available from a previous study [5] was processed into two microstructural conditions with respective grain sizes of 30 μm and 451 μm. Two castings of low-impurity Al alloys were purchased from the Kaiser Technology Center, Pleasanton, CA, both of ternary composition. The two ternary alloys contain 3.0 wt pct Mg with respective Mn contents of 0.25 and 0.50 wt pct. These castings were homogenized at 550 °C for eight hours, quenched and then hot-rolled at 370–450 °C to a thickness of approximately 10 mm. These plates then were cold-rolled to a final thickness of 4.57 mm, annealed for one hour at 400 °C and fan cooled. The composition and grain size of each material are given in Table 1, where material identification numbers are also given. Each material will subsequently be referred to by the identification number given in Table 1. Grain sizes were determined by optical microscopy using lead anodization and polarized light. True grain sizes are given in Table 1, where the relationship between true grain size, d, and linear intercept grain size, $\bar{l}$, is given by $d = 1.776\,\bar{l}$ [8]. Material 4 exhibited

Table 1: Materials examined in the present investigation.

Material	Composition, wt pct	Symbol	d, μm
1	Al–2.8Mg	○	30
2	Al–2.8Mg	▼	451
3	Al–3.0Mg–0.25Mn	▲	201
4	Al–3.0Mg–0.50Mn	◇	142*

* bimodal microstructure

a bimodal grain-size distribution, with the majority of material exhibiting a grain size of 142 μm and small patches or bands showing a grain size of 50 μm.

Tensile coupons were machined from each material with a gage length of 25.4 mm. Two types of tension tests were performed on each material. Tensile elongation-to-failure tests were conducted at constant values of true-strain rate, calculated based on uniform deformation in the gage section. Tests were conducted at temperatures from 300 °C to 500 °C over true-strain rates ranging from 10^{-4} to 0.1 s^{-1}. Strain-rate-change tests were also conducted at temperatures of 300 °C and 400 °C. True-strain rate was varied from slow to fast rates over a range from 10^{-4} to 2×10^{-2} s^{-1}. At the beginning of each strain-rate-change test an initial prestrain of approximately 10% was imposed at a low rate in order to stabilize the microstructure before strain-rate jumps.

Results and Discussion

Data from strain-rate-change tests at 400 °C of materials 1 and 2 are given in Figure 1 as a plot of true-strain rate against true stress on dual-logarithmic scales. Despite a significant difference in grain size, 30 μm versus 451 μm, the data from both materials fall onto a single curve. The slope of the data yields the stress exponent, n, which is the inverse of the strain-rate sensitivity, m. The strain-rate sensitivity of both materials 1 and 2 is found to be $m = 0.28$ from Figure 1. This value of m is very close to that predicted, and expected, for solute-drag creep ($m = 0.3$). The data of Figure 1 provide evidence that neither strength nor strain-rate sensitivity in the solute-drag-creep regime are significantly affected by grain size. Because an increase in grain size does not alter the value of m, the coarse-grained material should exhibit similar tensile elongations to the fine-grained material. This assumes that failure occurs by necking to a point, a process controlled by the strain-rate sensitivity of the material.

Tensile elongation data for materials 1 and 2 at various true-strain rates and temperatures are plotted against logarithm of diffusivity-compensated strain rate in Figure 2. The diffusivity coefficient is calculated using a value of $D_0 = 5 \times 10^{-5}$ m^2/s [9] and an activation energy of $Q = 136$ kJ/mol [10]. Figure 2 shows that both materials 1 and 2 exhibit similar elongations, but that elongations of the coarse-grained material lie at the bottom of the range of elongations of the fine-grained material. Material 1 begins to exhibit extraordinarily large elongations below values of $\dot{\epsilon}/D = 7.2 \times 10^{12}$, where solute-drag creep is the controlling deformation mechanism. In order to explain why the data for material 2 do not exhibit these anomalously large elongations, the failed samples were examined. One test each on materials 1 and 2 was performed at 400 °C and $\dot{\epsilon} = 10^{-4}$ s^{-1}, and the failure regions of the resulting samples were compared. Both samples exhibited very sharp, pinpoint failures with gradual necks. The fine-grained material exhibited a slightly less steep neck than the coarse-grained material. This would seem to indicate that there is little

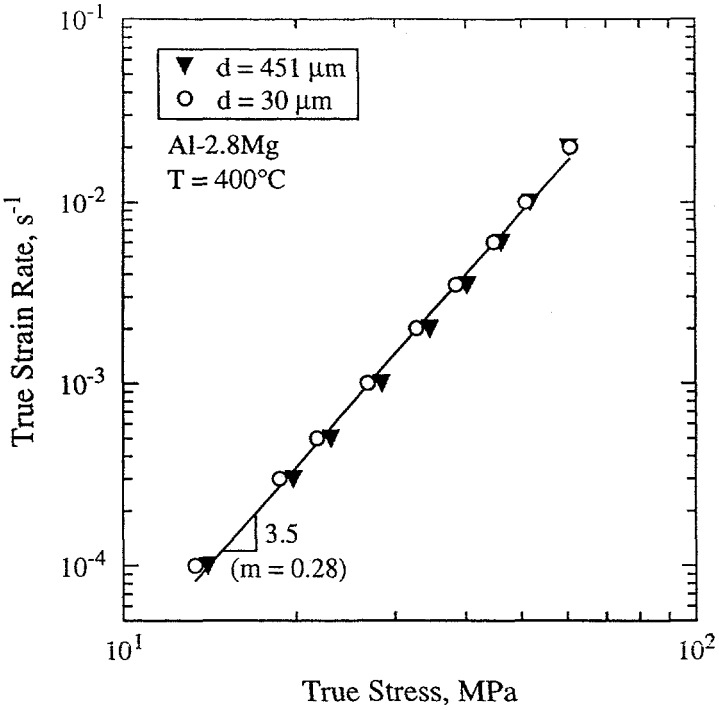


Figure 1: True-strain rate is plotted against true stress on dual-logarithmic scales for Al–2.8Mg in both fine-grained ($d = 30$ μm) and coarse-grained ($d = 451$ μm) conditions.

difference in behavior between the fine and coarse-grained materials. At high values of m, however, small increases in m can lead to significant increases in elongation, as evidenced by the scatter in the data from material 1. It is postulated that a larger number of tests on material 2 might yield a range of elongations similar to that of material 1. This conclusion is important because it indicates that enhanced ductility materials do not require any of the expense that is necessary in superplastic materials to obtain a stable, fine grain size.

Data from strain-rate-change tests on materials 1 and 3 at 300 °C and 400 °C are shown in Figure 3. It should be noted that both materials 1 and 3 contain similar amounts of Mg (2.8 and 3.0 wt pct), and differ primarily in the ternary addition of 0.25 wt pct Mn in material 3. This figure shows only a slight difference in strength between materials 1 and 3, where the Mn additions in material 3 yield a slightly stronger material under most conditions. A second observation can be made from Figure 3 concerning the strain-rate sensitivity. The strain-rate sensitivity of material 3 at 400 °C is slightly lower than that of material 1, $m = 0.25$ versus $m = 0.29$. The strain-rate sensitivity of material 3 at 300 °C is also lower than that of material 1. The lower strain-rate sensitivity caused by the addition of Mn in material 3 should lead to more rapid necking and lower tensile elongations than in material 1. The data of Figure 3 are replotted in Figure 4 as diffusivity-compensated strain rate versus modulus-compensated stress on dual-logarithmic scales. Figure 4 shows that both materials 1 and 3 exhibit power-law-breakdown at roughly the same value of $\dot{\epsilon}/D = 7.2 \times 10^{12}$ m^{-2}. Above this value, m continuously decreases because of power-law-breakdown, and elongations should decrease as a result. Below this value, solute-drag creep occurs, providing high m values and enhanced ductilities. A slight difference in slope between the data of materials 1 and 3 attests to the difference in strain-rate sensitivities, as evident in Figure 3.

Figure 5 shows elongation-to-failure data for materials 1, 3, and 4 plotted against the logarithm of diffusivity-compensated strain rate. As expected from its slightly lower value

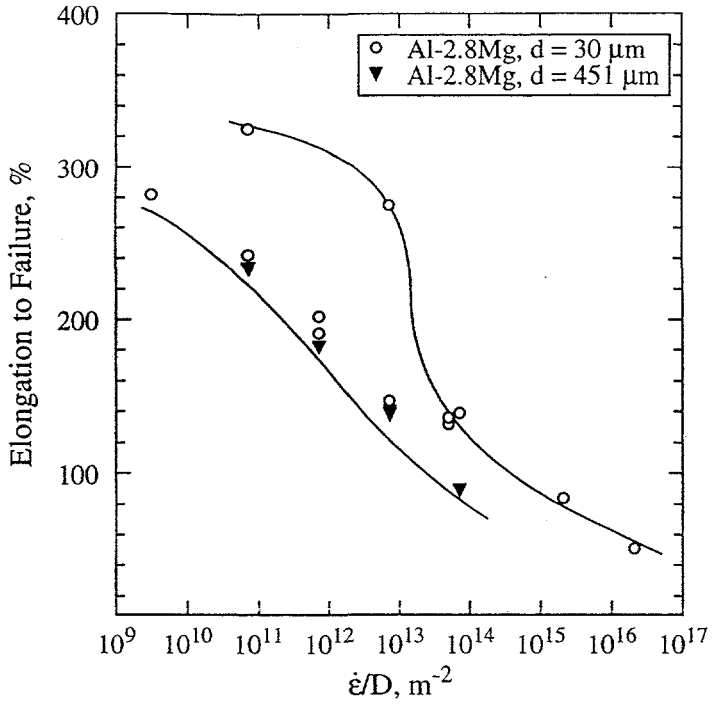


Figure 2: Elongation to failure is plotted against logarithm of diffusivity-compensated strain rate for Al–2.8Mg in both fine-grained ($d = 30$ μm) and coarse-grained ($d = 451$ μm) conditions. The two materials exhibit similar elongations, with data from the coarse-grained material lying at the bottom of the elongations exhibited by the fine-grained material.

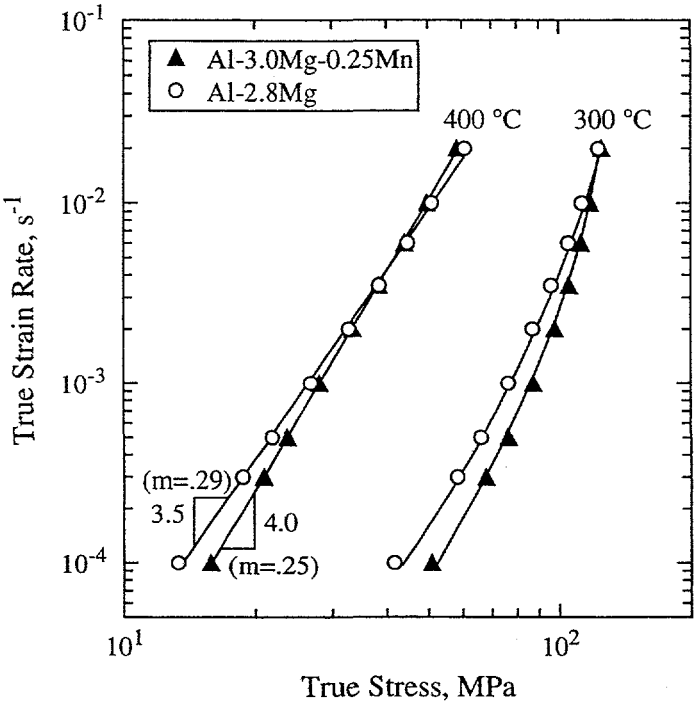


Figure 3: Strain-rate-change test data are plotted for two Al materials with similar Mg contents. One material contains 0.25 wt pct Mn while the other contains no Mn.

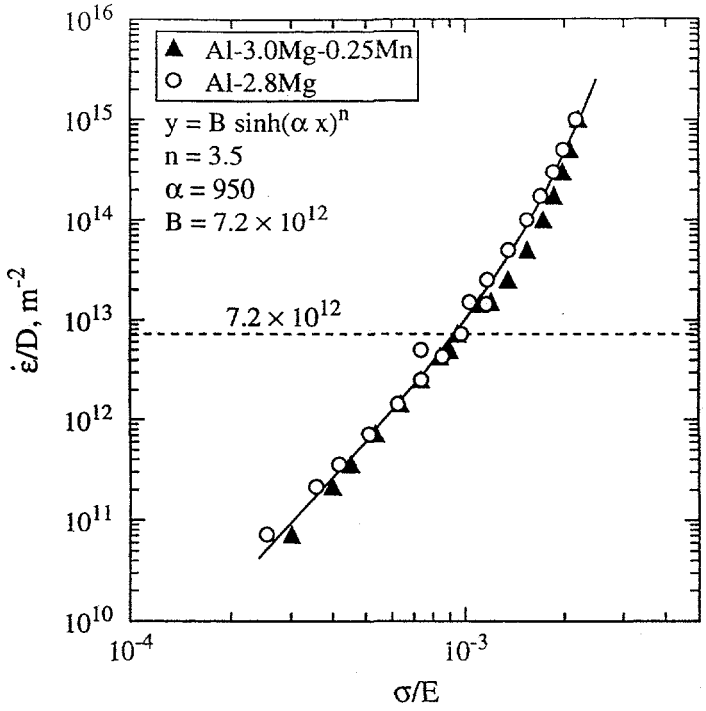


Figure 4: Test data are plotted as the logarithm of diffusivity-compensated strain rate versus the logarithm of modulus-compensated flow stress. The data for Al–3.0Mg–0.25Mn and Al–2.8Mg fall on similar curves, with slight differences in slope. Power-law breakdown occurs at values of diffusivity-compensated strain rate above 7.2×10^{12} m^{-2}.

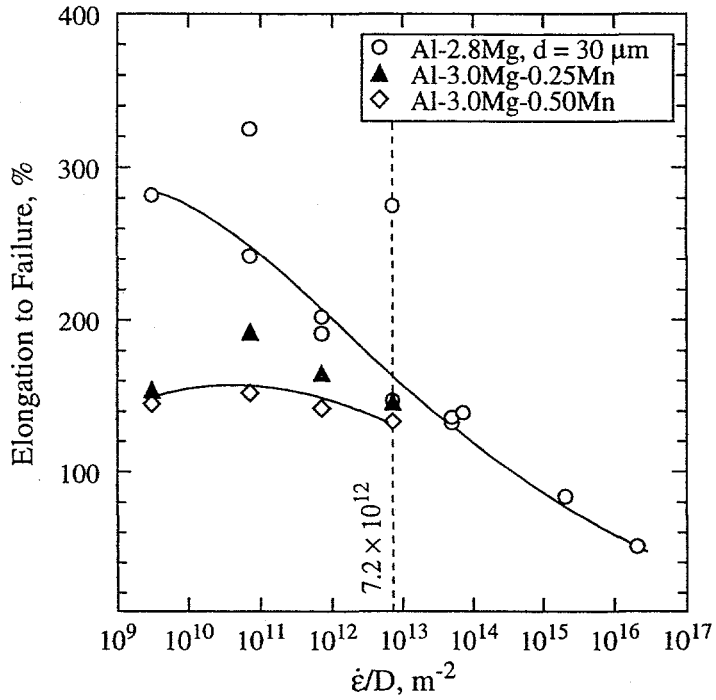


Figure 5: Data for elongation to failure are plotted against the logarithm of diffusivity-compensated strain rate for material with and without Mn. Tensile elongations significantly increase below the region of power-law-breakdown due to the solute-drag-creep process. The materials containing Mn show lower elongations than the binary alloy, which contains no Mn.

Table 2: Elongation data from Mn-containing Al–Mg materials.

Data Source	Symbol	Composition, wt pct	Ref.
Ayres and Wenner	●	4.05 Mg, 0.26 Mn, 0.22 Fe	[12]
Taleff, Lesuer, and Wadsworth	□	2.52 Mg, 0.46 Mn	[6]
Tavassoli, Razavi, and Fallah	▽	4.6 Mg, 0.75 Mn, 0.2 Fe, 0.15 Si	[11]

of strain-rate sensitivity, material 3 shows lower elongations than material 1. Material 4, which contains 0.5 wt pct Mn and for which strain-rate change data are not yet available, shows even lower values of elongation than material 3 (0.24 wt pct Mn). A primary cause of the lower elongations in the ternary materials is the decrease in strain-rate sensitivity, which allows more rapid neck formation and growth. As with materials 1 and 2, elongation-to-failure samples were compared for test conditions of 400 °C and $\dot{\epsilon} = 10^{-4}$ s^{-1}. The failure region of material 3 exhibits a sharp, pinpoint failure with a neck region slightly steeper than that of material 1. Material 4, however, exhibits a fairly flat, ragged failure with only very slight necking. The ragged failure surface of material 4, as opposed to the pinpoint failures of materials 1 and 3, attests to another effect of Mn which limits enhanced ductility. The addition of 0.5 wt pct Mn in material 4 causes premature fracture, presumably by cavitation, before severe necking occurs. Optical and transmission electron microscopy revealed a small number of dispersoids in material 4, which are not present in the other materials. These dispersoids could lead to cavitation and the flat fracture observed in material 4. Another possibility is that some Mn segregates to grain boundaries, weakening them and leading to the observed flat fracture. Further microstructural investigation is currently underway in order to more fully clarify the effects of Mn in material 4.

In order to better understand the role of Mn in the ductility of Al–Mg materials at the elevated temperatures where enhanced ductility occurs, data from several investigators were analyzed [6, 11, 12]. The sources of these data and the alloy compositions studied are given in Table 2. All of these materials fit the criteria for exhibiting solute-drag creep, based on reported alloy compositions and grain sizes. The materials of Table 2 are regarded as commercial-purity alloys, as opposed to the low-impurity alloys of the present study. The data from sources in Table 2 cover Mn compositions from 0.26 to 0.75 wt pct, with Mg compositions reasonably close to those of the present investigation. McNelley *et al.* have shown that the strength and strain-rate sensitivity of binary Al–Mg alloys are not significantly affected by Mg concentration in the range of 2–6 wt pct [10]. Figure 6 utilizes data from strain-rate-change tests to demonstrate the effect of increasing Mn content on strength and strain-rate sensitivity at a temperature of 400 °C. Data are plotted as logarithm of strain rate versus logarithm of flow stress. A clear decrease in strain-rate sensitivity from $m = 0.29$ to 0.25 to 0.21 is seen as Mn content increases from 0 to 0.25 to 0.5 wt pct. A corresponding increase in strength at strain rates below $\dot{\epsilon} = 3 \times 10^{-3}$ s^{-1} is seen for increasing Mn content. The data from Tavassoli *et al.* for 0.75 wt pct Mn exhibit the highest strength, but only the second lowest strain-rate sensitivity, $m = 0.24$. From the strain-rate sensitivity values given in Figure 6, it is expected that increasing Mn content will decrease tensile ductility in the solute-drag creep regime, similar to the effect shown in Figure 5.

Figure 7 illustrates the effect on ductility by an addition of 1/4 wt pct Mn. Tensile elongation is plotted against logarithm of diffusivity-compensated strain rate. A vertical, broken line is given at the transition between solute-drag creep and power-law breakdown. Data to the right of this line, in the power-law-breakdown region, exhibit little difference

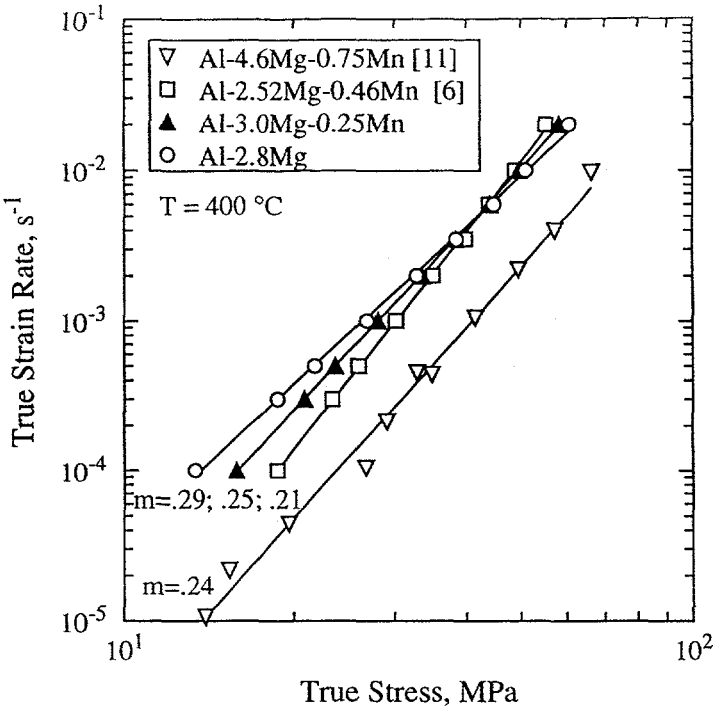


Figure 6: Strain-rate-chage data are plotted as logarithm of strain rate versus logarithm of flow stress. Data are from this study, Taleff *et al.* [6], and Tavassoli *et al.* [11].

between material without Mn and material with 1/4 wt pct Mn. This behavior changes to the left of the vertical line, where solute-drag creep controls deformation. In this region the material without Mn clearly exhibits higher elongations. The difference between elongations becomes even more pronounced at the lowest values of $\dot{\epsilon}/D$, where temperature is highest and strain rate is lowest. The decrease in elongations corresponds to the decrease in strain-rate sensitivity found in Figure 6. This behavior could also be indicative of cavitation-induced failure in the Mn-containing materials, as cavitation generally occurs more easily at high temperatures and low strain rates.

Data for materials with Mn contents ranging from 1/4 wt pct to 3/4 wt pct are given in Figure 8 as a plot of elongation versus logarithm of diffusivity-compensated strain rate. The data in Figure 8 clearly indicate different behaviors for the materials containing 1/4 wt pct Mn and those containing 1/2 to 3/4 wt pct Mn. The materials with 1/2 to 3/4 wt pct Mn exhibit lower elongations than those containing 1/4 wt pct Mn in both the power-law-breakdown and the solute-drag-creep regimes. Again, a decrease in m and an increase in cavitation with an increase of Mn content could explain this behavior. In this case, cavitation would be affecting elongation not only at the lowest values of $\dot{\epsilon}/D$, but under all of the conditions where data are shown.

Conclusions

Solute-drag creep can yield enhanced ductilities of over 300% in simple Al–Mg binary alloys. It is shown that grain size does not affect the creep behavior of binary Al–Mg in the solute-drag-creep regime. Coarse-grained, binary Al–Mg material, therefore, yields elongations virtually identical to those of fine-grained material. Ternary alloys of Al–Mg–Mn yield slightly lower ductilities of 150% to 200% in the solute-drag-creep regime. Ternary additions of Mn decrease the strain-rate sensitivity of Al–Mg alloys during solute-drag creep, resulting

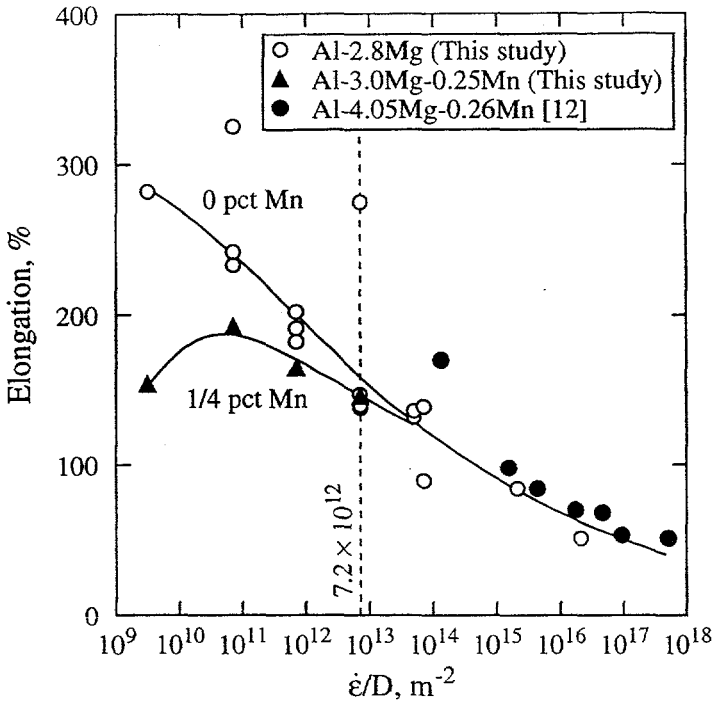


Figure 7: Data are plotted as elongation versus logarithm of diffusivity-compensated strain rate for materials containing zero to 1/4 wt pct Mn. Data are from this study and Ayres and Wenner [12].

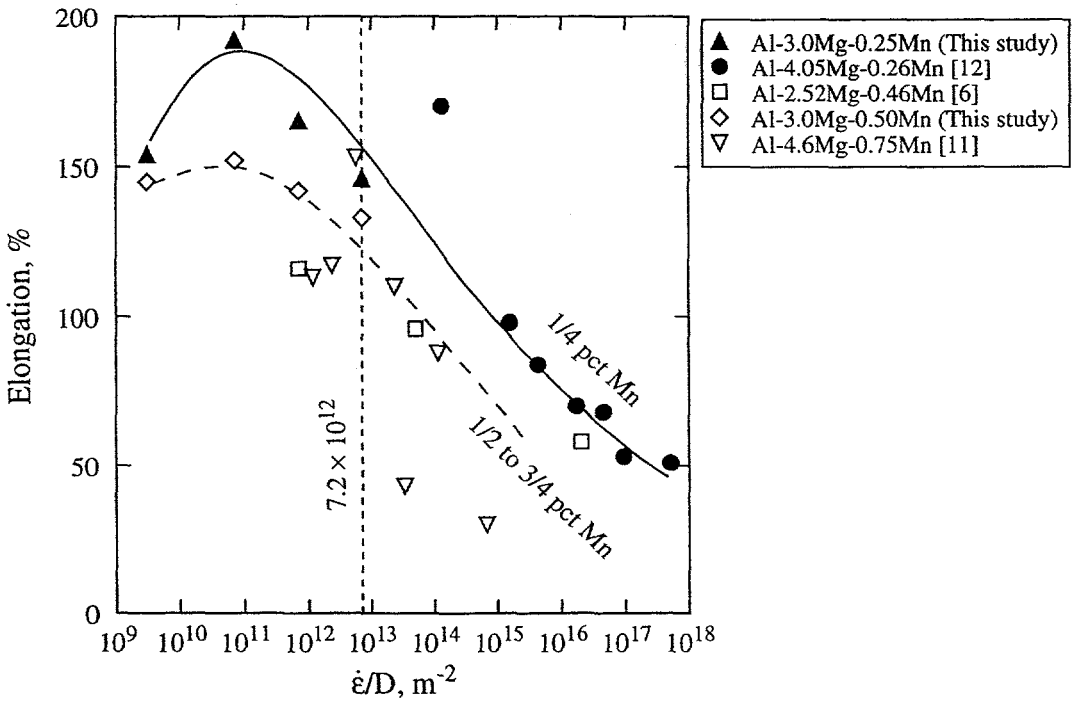


Figure 8: Data are plotted as elongation versus logarithm of diffusivity-compensated strain rate for materials containing 1/2 to 3/4 wt pct Mn. Data are from this study, Ayres and Wenner [12], Taleff *et al.* [6], and Tavassoli *et al.* [11].

in lower elongations. Mn has a secondary effect of causing ductile fracture before necking to a point when added to Al–Mg in a concentration of 0.5 wt pct. Materials with Mn compositions from 1/2 wt pct to 3/4 wt pct exhibit lower ductilities than materials with 1/4 wt pct Mn or less in both the solute-drag-creep and power-law-breakdown regimes.

Acknowledgments

The authors thank Mr. Rick Gross for performing mechanical testing, Mr. Robert Kershaw for optical metallography, and Mr. Mark Wall for performing TEM. This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract no. W-7405-ENG-48.

References

[1] O. D. Sherby and J. Wadsworth, "Superplasticity-Recent Advances and Future Directions," *Prog. Mater. Sci.*, **33** (1989), 169–221.

[2] O. D. Sherby and J. Wadsworth, "Development and Characterization of Fine-Grain Superplastic Materials," *Deformation, Processing, and Structure* (Metals Park, Ohio: American Society for Metals, 1982), 355–389.

[3] H. J. McQueen and M. E. Kassner, "Extended Ductility of Al and α-Fe Alloys Through Dynamic Recovery Mechanisms," *Superplasticity in Aerospace*, eds. H. C. Heikkenen and T. R. McNelley (Warrendale, PA: The Metallurgical Society, 1988), 77–96.

[4] E. M. Taleff, G. A. Henshall, D. R. Lesuer, and T. G. Nieh, "Warm Formability of Aluminum-Magnesium Alloys," *Aluminum Alloys: Their Physical Properties and Mechanical Properties (ICAA4)*. eds. T. H. Sander, Jr. and E. A. Starke, Jr. (Atlanta, Georgia: Georgia Institute of Technology, 1994), 338–345,

[5] E. M. Taleff, G. A. Henshall, D. R. Lesuer, T. G. Nieh, and J. Wadsworth, "Enhanced Ductility of Coarse-Grain Al-Mg Alloys," *Superplasticity and Superplastic Forming*, ed. A. K. Ghosh (Warrendale, PA: TMS, 1995), 3–10.

[6] E. M. Taleff, D. R. Lesuer, and J. Wadsworth, "Enhanced Ductility in Coarse-Grained Al-Mg Alloys," *Metall. Mater. Trans.*, (Accepted for publication, 1995).

[7] L. Ivanchev and D. Chavderova, *Proceedings of the International Symposium on Plasticity and Resistance to Metal Deformation*, ed. S. Blecic (Titograd: Fac. Metall., 1986), 348–361.

[8] A. Ball and M. M. Hutchison, "Superplasticity in the Aluminium-Zinc Eutectoid," *Matals Sci. J.*, **3** (1969) 1–7.

[9] CRC Press, Cleveland, Ohio. *Handbook of Chemistry and Physics*, 55th edition, (1974).

[10] T. R. McNelley, D. J. Michel, and A. Salama, "The Mg-Concentration Dependence of the Strength of Al-Mg Alloys During Glide-Controlled Deformation," *Scripta Metall.*, **23** (1989) 1657–1662.

[11] A. A. Tavassoli, S. E. Razavi, and N. M. Fallah, "Superplastic Forming of a Commercial Aluminum Alloy," *Metall. Trans.*, **6A** (1975) 591–594.

[12] R. A. Ayres and M. L. Wenner, "Strain and Strain-Rate Hardening Effects in Punch Stretching of 5182-0 Aluminum at Elevated Temperatures," *Metall. Trans.*, **10A** (1979) 41–46.

functions. The server thread executes the code that issues requests to HPSS.

One server thread manages a given open file on behalf of all the nodes, and servers on different processors can manage different open files. This prevents any single node from becoming a bottleneck or bearing the burden of managing all the open files. We call this aspect of the architecture the *distributed server model.* Conceptually, the server and client threads could be separate processes, since they share no data structures. However MPI cannot at present direct messages to different processes on the same node, so using MPI for communication requires the server and client to reside in the same process.

Any time a client thread needs to operate on a file, it sends a request via MPI to the server thread on the appropriate node. Each server maintains a table of the open files it manages. When a request arrives, the server looks up the HPSS file descriptor and other information about the file and then spawns a *driver thread* to issue the HPSS request. When this request is complete, the driver thread sends a response message to the client and then terminates. The client thread receives the message and the original MPI-IO call returns a result to the application program.

4.1 Opening and closing a file

Opening a file in MPI-IO is always a collective operation, which means that all the nodes in the program (or a specific subset of them) participate. The nodes select a server by hashing the file name and other parameters to the **open** call to produce a node number. Since all the nodes must specify the same parameters to the call, they will all select the same node without needing to communicate with each other. The server's node number is stored in a local file table on each node for use in future requests.

Each node sends a request to the server as soon as it is ready; there is no barrier synchronization upon opening a file. When the server receives the first **open** request for a given file, it creates an entry in the file table and spawns a driver thread to call HPSS. Subsequent requests from other nodes to open the same file will find a matching request in the file table. If the HPSS **open** call has already completed, the server will send a reply containing the data from the completed (or possibly failed) call. If the HPSS call is still pending, the new request will be placed in a queue. As soon as the driver thread completes the HPSS call, it will send responses to the nodes with queued requests. This arrangement guarantees that each **open** request

generates exactly one call to HPSS, and requests from other nodes to open the same file share the results of this call.

Closing a file is also a collective operation, and the nodes again send individual requests to the server. This time, however, the server delays spawning a thread to issue the HPSS **close** call until all the requests have arrived. Therefore, closing a file is a synchronizing operation. This is necessary because the file cannot be closed until all the nodes are finished with it, and any errors that occur when HPSS closes the file must be reported to all participating nodes. Moreover, if the close operation does not synchronize, a node might treat a file as if it were closed and its buffers flushed when the file is in fact still open and handling requests from other nodes.

4.2 Reading and writing

Programs can read and write files collectively or independently, and they can intermix these operations freely on the same file (provided that all nodes that open a file participate in the collective operations). Figure 3 shows how these operations work.

For an independent read or write operation, the client first spawns a mover thread that will copy data between the memory buffer and the network channel to the storage device. When this thread has started, the client sends a read or write request to the server. The request includes the information that the server will need to construct an HPSS IOD (see Section 2). The server spawns a driver thread to issue the HPSS **readlist** or **writelist** call. HPSS transfers the data directly between the node and the storage device and then returns from the **readlist** or **writelist** call. Part of the return data is a structure called an IOR (I/O reply), which the driver thread sends back to the mover before terminating. The mover compares the IOR to its own record of the transfer, then returns status information to the client thread and terminates. The SIOF API code in the client thread transforms the status information into MPI-IO return data before finally returning from the MPI-IO call.

Collective operations require a few extra steps. The details appear in Section 5, but the main difference from independent operations is that the server may gather up several requests from different nodes and issue them together in a single HPSS call.

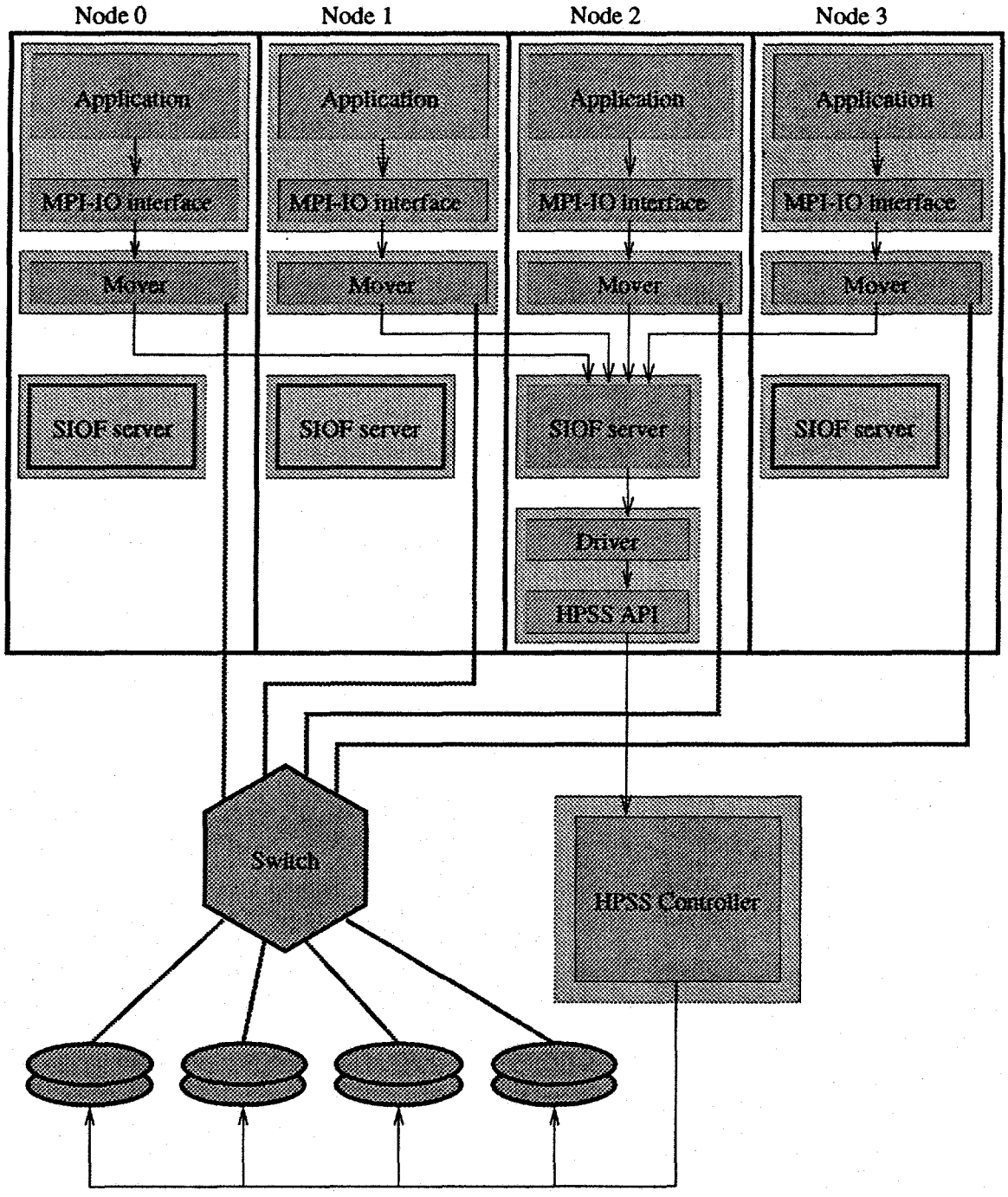


Figure 3: The SIOF API is implemented in several threads on each node (shown here as four large, vertical rectangles). Outer shaded boxes represent threads; inner boxes are functional modules within a thread. Outlined boxes show modules not participating in an operation. For read and write operations, control is centralized at a server thread, but data travels through separate, high-bandwidth channels between storage devices and compute nodes.

4.3 File types and buffer types

Section 3 noted that MPI-IO programs can use file types and buffer types to access discontiguous regions of data. MPI-IO translates these datatypes into an internal format called a *chunk map*. A chunk map is a list of contiguous data blocks, and it contains only the information that the SIOF API needs from an MPI datatype to construct an IOD.

Because MPI specifies no functions for accessing the layout information in a datatype, the SIOF API code must explicitly read the internal data structures of the MPI implementation on which it is based (MPICH [1]). One reason for using chunk maps is to isolate the system-dependent code as much as possible, so most of the SIOF API code works with chunk maps rather than MPI datatype structures.

The SIOF API stores the chunk map of the file type for each node and each open file in the server thread's file table. When a file is read or written, the server constructs an HPSS IOD for the data to be transferred, with source and sink mappings for each contiguous chunk of data to be accessed. It passes this IOD to a single HPSS call.

Meanwhile, the mover thread parses the chunk map corresponding to the buffer type to determine which data to access in memory. The SIOF API does not compare buffer types with file types or decompose them with respect to each other; HPSS and the client mover thread can each behave as if the other is accessing a single, contiguous stream of data.

5 Managing collective operations

The SIOF API currently supports four types of data access: the independent **read** and **write** operations, and collective versions called **read-all** and **write-all**. Structuring the server to permit collective operations on reads and writes requires that several issues be addressed:

- How are collective operations implemented?
- How is the decision made to dispatch them?
- What optimizations are available for collective operations?

This section discusses these three issues.

5.1 Collective implementation

From the server's point of view, the life-cycle of a collective operation begins when the server receives the first MPI message containing a data access request from a client thread. The request is added to a list of pending data accesses for that file. This *data access list* is traversed after either the receipt of a client's message for a file operation, or a predetermined period of time has elapsed, whichever comes first. As the server traverses the list, it updates a *dispatch priority* for each pending collective operation. The dispatch priority determines when the server will initiate the data access; if the priority is over a predetermined threshold, the server spawns a thread to issue the HPSS **readlist** or **writelist** call. If a collective write request includes overlapping file accesses by different nodes, the server constructs an IOD that resolves the conflict in a well-defined way.

The data access list also records the number of outstanding clients, which is needed to handle cases where the server dispatches a request before all clients have checked in. The number of clients is initially the number of nodes that have jointly opened the same file, but if two or more dispatches are used for the same operation, it will be the number of clients remaining for the operation (*i.e.*, the number not already checked in and previously dispatched).

5.2 Determining dispatch priority

How the dispatch priority is determined will have a strong effect on performance and utilization of the I/O system. For example, one can imagine a scenario in which 15 clients of a 16-client application check in at nearly the same time, but the 16th client checks in much later. In such a scenario, it may be advantageous to forgo waiting for the last node to check in before dispatching the requests for the first 15 nodes. On the other hand, issuing a request too soon will reduce the ability of the SIOF API library to amortize latency costs involved in setting up a data access. A number of factors may play a part in determining the dispatch priority. At the present, our implementation for MPI-IO **read-all** and **write-all** operations blocks until all client nodes have checked in. However, we plan to investigate several algorithms to determine their effect

on utilization and performance. These algorithms will consider, to varying degrees, the time since a request was first issued, information on which clients have checked in, the transfer size, the granularity of the file types, and whether the access is to tape or disk.

5.3 Optimizations

The architecture of the SIOF API makes several optimizations feasible. These include:

- Asynchronous operation.
- Grouping accesses on the same storage device.
- Grouping accesses on the same processor.
- Coalescing small accesses.

The first optimization reduces a server's sensitivity to the latency of HPSS calls. By spawning a thread for each such call, the server can handle multiple requests concurrently.

Grouping accesses to the same storage device can help improve cache performance. For example, certain decompositions of matrices among processors can produce requests for small, interleaved chunks of data [6]. By constructing IODs so that requests for sequential data appear in order, the server can increase the probability of cache hits on a disk. On the other hand, sending small blocks of data between one disk and multiple nodes in round-robin order may produce excessive switching latency in the external network. In that case, it may be better to group requests so that data residing on one node is accessed sequentially. Performance tuning will help us determine how best to arrange the parts of a collective request.

Even if there is no locality to be exploited in a collective operation, grouping requests from multiple nodes into a single **readlist** or **writelist** call can amortize one-time expenses incurred in I/O operations, such as the cost of an RPC transaction between the parallel computer and the HPSS controller.

6 Current status

The SIOF API is currently undergoing integration testing, and we expect to do performance testing and tuning by the end of 1995. Our initial version of the code includes both independent and collective read and write operations. MPI-IO file types and buffer types are fully functional. The main features of MPI-IO that we have not yet implemented are nonblocking I/O calls, shared file pointers, and exception handling. We expect to implement these features over the next year. Over the longer term, we will investigate new features that will simplify access to nonuniform data layouts.

7 Conclusion

The SIOF API is a new implementation of the proposed MPI-IO standard. It is designed as a high-level user interface for the HPSS file system, and its initial implementation is on a Meiko CS-2 parallel computer. Because HPSS supports third-party transfers over an external network, our implementation can transfer data in parallel between processors and storage devices while presenting a global view of the file system that all nodes can access. Our distributed server model spreads the burden of coordinating data transfers over multiple nodes. Control of a given open file is centralized, but data transfer can proceed in parallel. We believe this combination of features will offer the high aggregate I/O bandwidth for large data transfers that many parallel scientific codes need.

References

[1] P. Bridges, N. Doss, W. Gropp, E. Karrels, E. Lusk, and A. Skjellum, *Users' guide to MPICH, a portable implementation of MPI.* http://www.mcs.anl.gov/mpi/mpiuserguide/paper.html.

[2] P. Corbett, D. Feitelson, Y. Hsu, J.-P. Prost, M. Snir, S. Fineberg, B. Nitzberg, B. Traversat, and P. Wong, *MPI-IO: A parallel file I/O interface for MPI, version 0.3*, NAS Technical Report NAS-95-002, IBM T.J. Watson Research Center and NASA Ames Research Center, January 1995.

[3] R. COYNE, H. HULEN, AND R. WATSON, *The High Performance Storage System*, in Proceedings of Supercomputing '93, November 1993.

[4] FIBRE CHANNEL ASSOCIATION, *Fibre Channel: Connection to the Future*, Austin, Texas, 1994.

[5] W. GROPP, E. LUSK, AND A. SKJELLUM, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, MIT Press, Cambridge, Mass., 1994.

[6] D. KOTZ AND N. NIEUWEJAAR, *Dynamic file-access characteristics of a production parallel scientific workload*, in Proceedings of Supercomputing '94, November 1994.

[7] OPEN SOFTWARE FOUNDATATION, *OSF DCE Application Development Reference*, Prentice-Hall, Englewood Cliffs, N.J., 1993.

[8] *Portable Operating System Interface (POSIX)—Part 1: System application programming interface (API)*. IEEE Standard 1003.1-1990.

[9] *Scalable I/O Facility.* http://www.llnl.gov/liv_comp/siof/siof.html.

[10] *Scalable I/O Initiative.* http://www.cacr.caltech.edu/SIO/.

[11] W. R. STEVENS, *Unix Network Programming*, Prentice-Hall, Englewood Cliffs, N.J., 1990.

[12] D. TEAFF, R. W. WATSON, AND R. A. COYNE, *The architecture of the High Performance Storage System (HPSS)*, in Proceedings of the Goddard Conference on Mass Storage and Technologies, March 1995.

[13] TRANSARC CORPORATION, *Encina product information.* http://www.encina.com/Public/ProdServ-/Product/Encina.

[14] R. WATSON AND R. COYNE, *The parallel I/O architecture of the High Performance Storage System (HPSS)*, in IEEE Symposium on Mass Storage Systems, September 1995.

[15] D. WILTZIUS, *Network-attached peripherals (NAP) for HPSS/SIOF.* http://www.llnl.gov/liv_comp-/siof/siof-nap.html.