The MDASH Utility for DASH Version 3.1

Introduction

MDASH is a small program designed to speed-up the execution of *DASH* 'grid-type' batch jobs on single PCs by taking advantage of the multiple-core architecture CPUs increasingly found in modern computers. For example, consider a *DASH* job that consists of 20 SA runs, each using 1 x 10^7 SA moves. Such a job might take 2 hours to execute on a CPU with a single core e.g. Intel Pentium, Intel Core 2 Solo. Running the same *DASH* job on a dual-core CPU (e.g. Intel Core 2 Duo) with the same clock speed will still take 2 hours, as the *DASH* program uses only one of the available processing cores. However, by creating a 'grid-type' batch file from within *DASH*, and then executing the batch file using *MDASH*, execution time is cut to only one hour. On a quad-core processor, execution time is cut to only 30 minutes.

Requirements

MDASH runs on MS Windows XP (SP2) and MS Windows Vista. It requires the MS .Net 2.0 Framework (or higher) to be installed. This is present by default in XP SP2 and Vista.

Installation

The *DASH* 3.1 installer places the *MDASH* executable in a subdirectory entitled 'Unsupported Extras'. We recommend creating a shortcut to this file (right-click over the 'Mdash.exe' file and select 'Create shortcut') and use this shortcut to invoke *MDASH*.

Running *MDASH*

On starting *MDASH*, the user is presented with a file selection dialog in order to select a *DASH* '.grd' file that has been setup as the result of a previous *DASH* session (see the main *DASH* documentation for details on how to generate a '.grd' file). Upon selecting the '.grd' file, the main *MDASH* window presents a summary of the job specified by the '.grd' file. The path to the selected '.grd' file is listed in the status bar at the foot of the window (Figure 1).

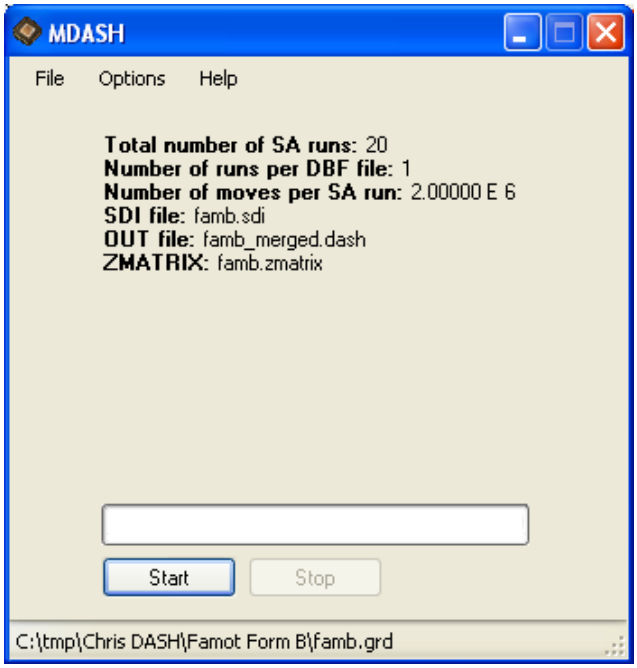


Figure 1: The main *MDASH* window after loading a .grd file.

Note in particular that the output '.dash' file takes its name from the root of the'.grd' file, with the addition of '_merged' e.g. 'famb.grd' will generate 'famb_merged.dash'. This '.dash' file will be placed in a date and time-stamped subdirectory (see Results section for more details) of the one containing the '.grd' file, such that several *MDASH* runs can be started from a single directory, if required.

When using *MDASH* for the first time, before pressing the 'Start' button, it is a good idea to check how the program is configured. To do this, select 'Configuration…' from the Options menu. The configuration dialog is then displayed (Figure 2) which allows the user to check and to alter the current configuration.

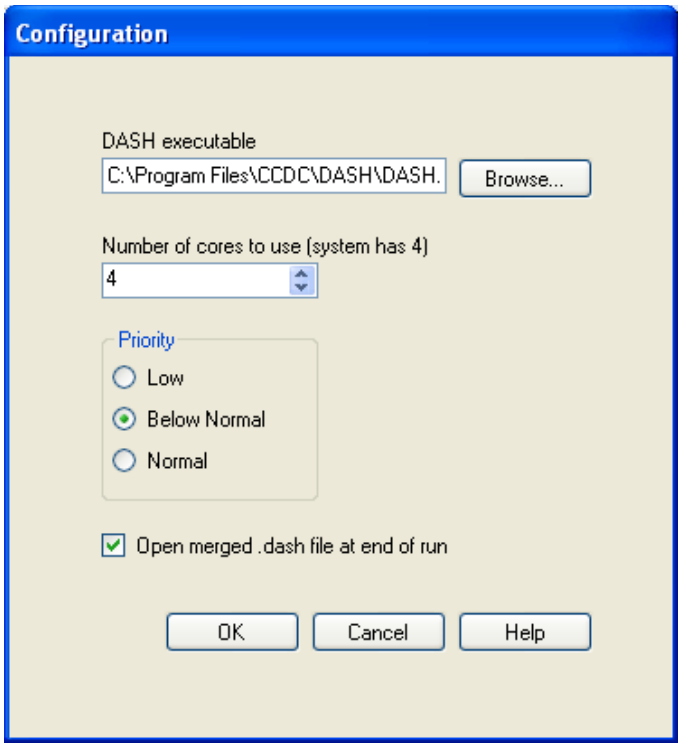


Figure 2: The *MDASH* configuration dialog.

- **DASH executable**: By default, the *DASH* executable invoked by *MDASH* is the one created by the *DASH* installer and you should not change it from this default. However, in some circumstances it might be useful to point *MDASH* towards a different version of *DASH* (e.g. an evaluation copy of a new version of *DASH*); the browse button allows you to do this. Not however that *MDASH* does not 'remember' any such changes i.e. the next time *MDASH* is started, it will point to the default version of *DASH*.

- **Number of cores to use**: *MDASH* queries the system to find out how many cores ('processors') are available to run *DASH* executables; by default, one *DASH* executable is started for every core reported. You can change the number of cores to be used via this control if, for example, the system is not correctly reporting the number of physical cores available (e.g. if your processor has Hyper-Threading turned on, it may appear to have more cores than are physically present on your processor) or if you wish to leave some cores fully available for other applications. Note however, that even if *DASH* is set to use all the available cores, this does not mean that you PC will become non-responsive. The default priority of 'Below normal' assigned to *DASH* runs ensures that the *DASH* jobs run effectively 'in the background' and should not interfere with you continuing to use your PC as normal. Although the control will allow you to set the number of cores higher than the reported value, this is <u>not recommended</u> and it may result in your computer becoming less responsive when the job is started.

- **Priority**: The default setting of 'Below normal' means that *DASH* runs started using *MDASH* are executed by the operating system at a lower priority than the majority of other processes that run on your PC. This ensures that the *DASH* runs will not interfere with your normal interaction with your PC.

  Although the control will allow you to set the priority of the runs to 'Normal', this is not recommended, as it is *likely* to result in your computer becoming unresponsive when the job is started.

  Although the control will allow you to set the priority of the runs to 'Low', this is not recommended, as it places the *DASH* runs on an equal priority with other jobs that may be running on your computer if it is part of a grid-type environment.

- **Open results in DASH**: When this box is ticked, results are automatically opened and displayed in *DASH* when the job is completed. If the box is not ticked, the results file is still generated but is not opened automatically.

Once you are happy with the configuration options, you can return to the main *MDASH* window and press the 'Start' button. The job is started and the status of the job is updated every 2 seconds (Figure 3).

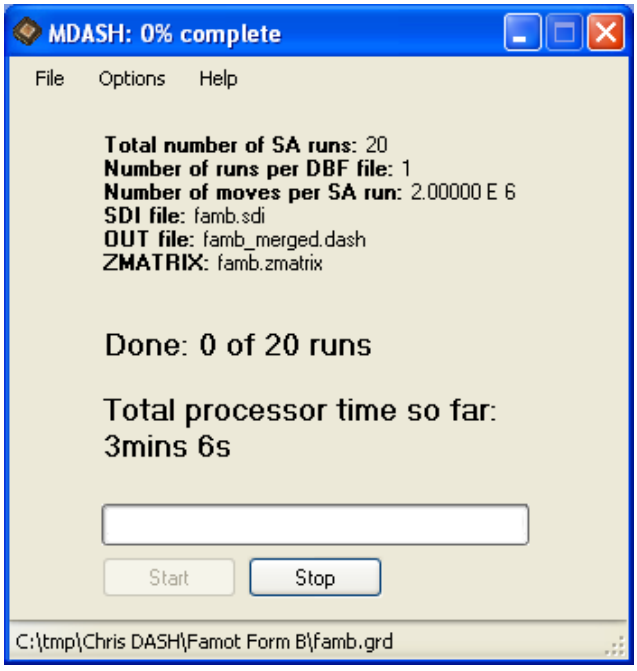


Figure 3: The main *MDASH* window, when a job is running.

You can use the Windows Task Manager to check on the system's status (Figure 4).

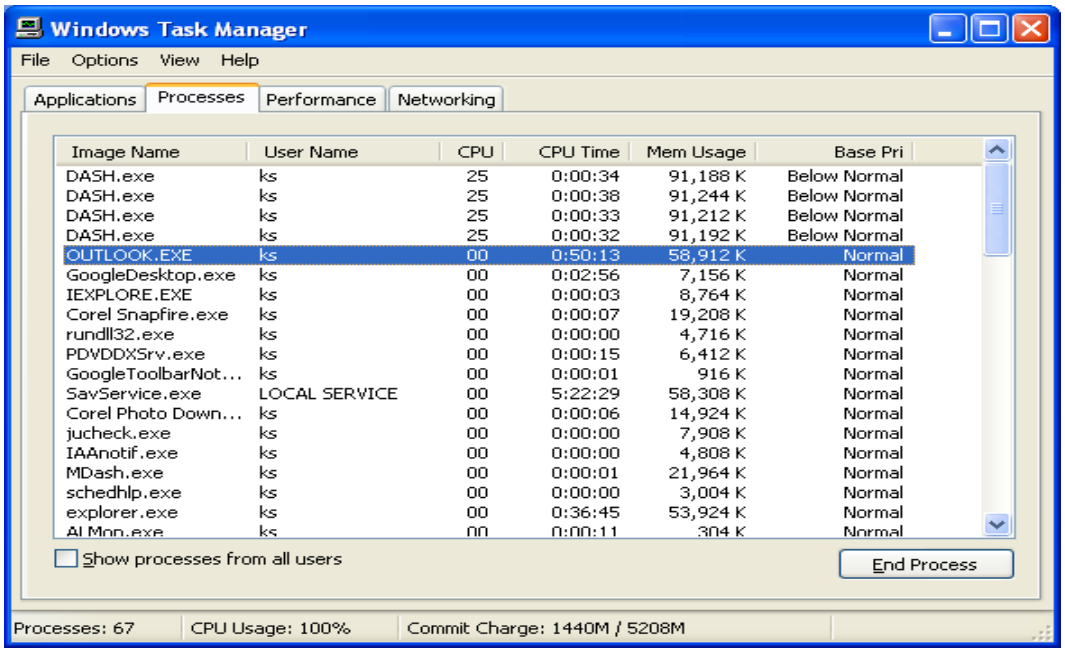


Figure 4: Four *DASH* executables running under *MDASH*'s control on a quad-core processor PC. Note that each *DASH* executable is fully utilising a single core and that the total CPU usage is 100%. Note also the 'Below Normal' priority of the *DASH* tasks.

As each SA run completes, the *MDASH* window updates to reflect progress (Fig 5).

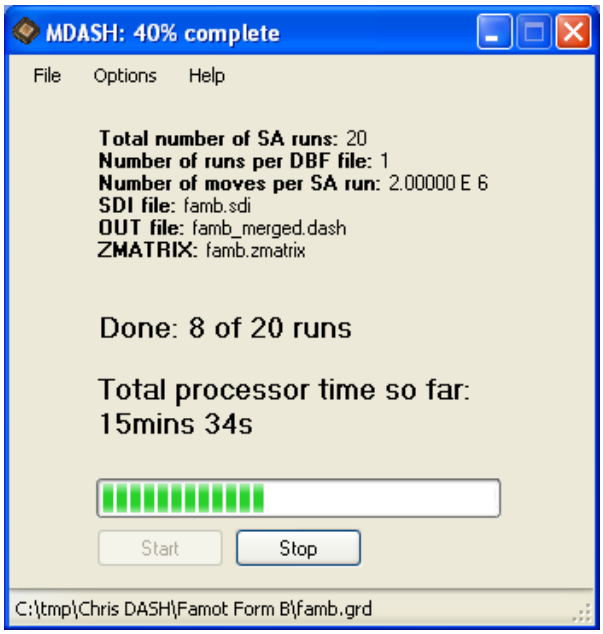


Figure 5: The main *MDASH* window, with a job at 40% completeness.

Note that even if the *MDASH* window is iconised, progress can still be monitored by looking at the taskbar (Figure 6).

Figure 6: A section of the Windows XP taskbar with *MDASH* iconised.

When the job is complete, if the 'Open results in *DASH*' configuration option is checked, the results file is opened and displayed for analysis, as if the entire job had been run from a single instance of *DASH* (Figure 7).

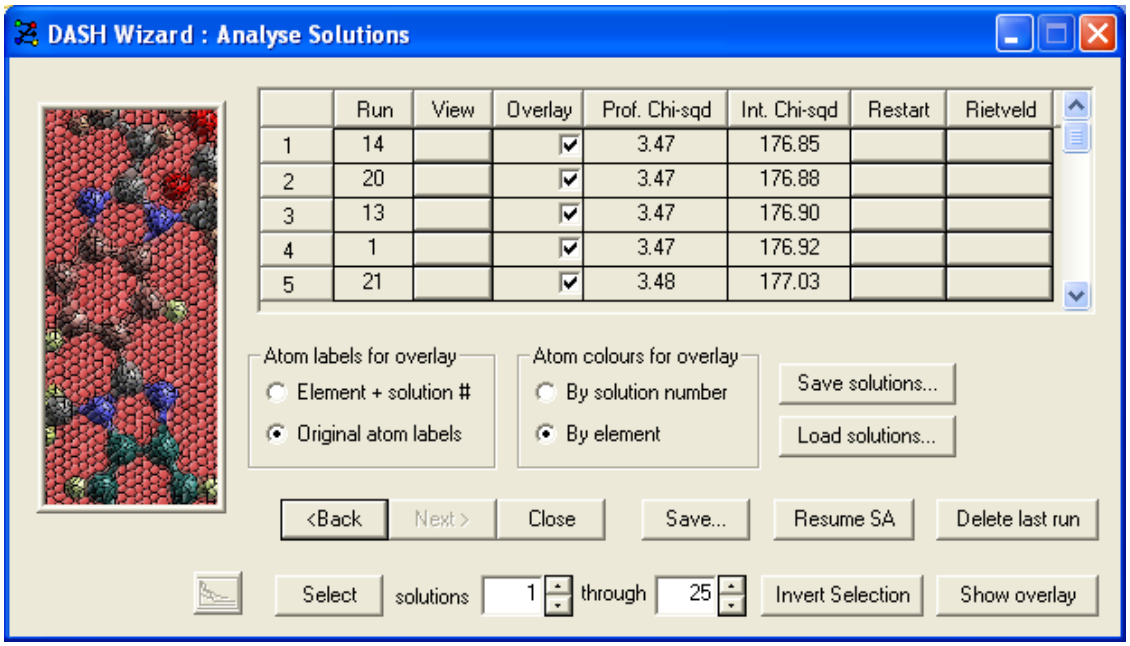
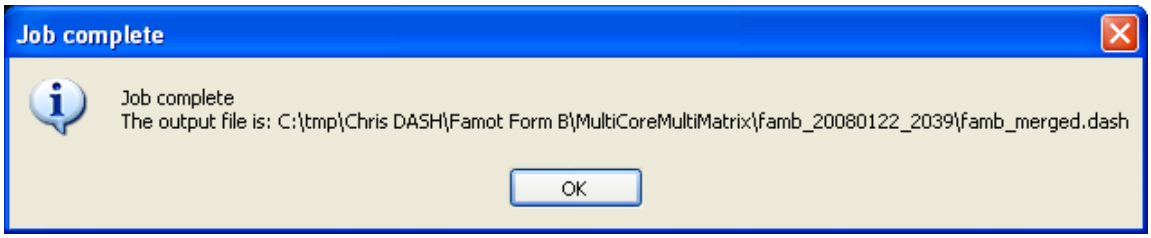


Figure 7: A *DASH* results file, as displayed automatically at the end of an *MDASH* job.

At this point, the user can safely exit *MDASH* and proceed with examining the results of the job using *DASH*. If the 'Open results in *DASH*' configuration option is not checked, then the job is complete when an information box appears as shown below:



The resultant '.dash' file can be opened using the 'Analyse solutions' option in *DASH*.

Exiting *MDASH* whilst a job is running

The 'Stop' button on the main *MDASH* window gives the user the option of stopping an *MDASH* run (Figure 8).

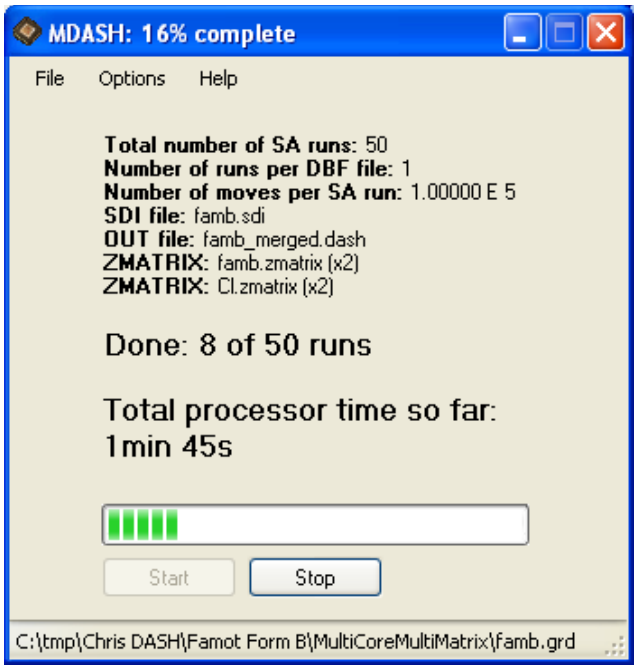


Figure 8: The main *MDASH* window, with a job at 16% completeness, showing the option to stop the job.

When the Stop button is pressed, a dialog asks the user to confirm that they really wish to stop running the job, before terminating all copies of *DASH* running under *MDASH*'s control (Figure 9) and deleting all completed SA runs of the partially completed job.

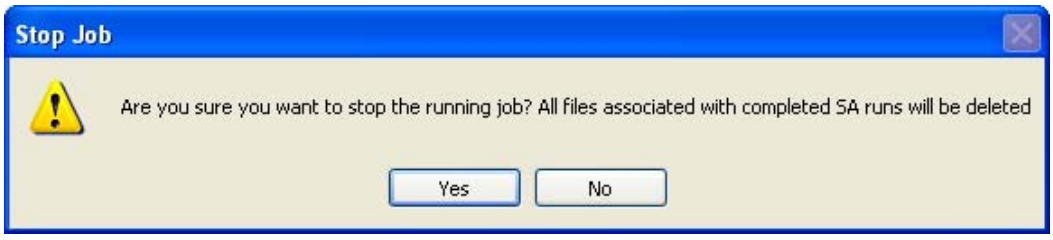


Figure 9: The 'stop job' warning dialog.

Similarly, if the user attempts to exit *MDASH*, either using File→Exit from the menubar or by clicking on the  button), a dialog again warns the user that exiting *MDASH* will terminate all copies of *DASH* running under *MDASH*'s control (Figure 10) and delete all completed SA runs of the partially completed job.

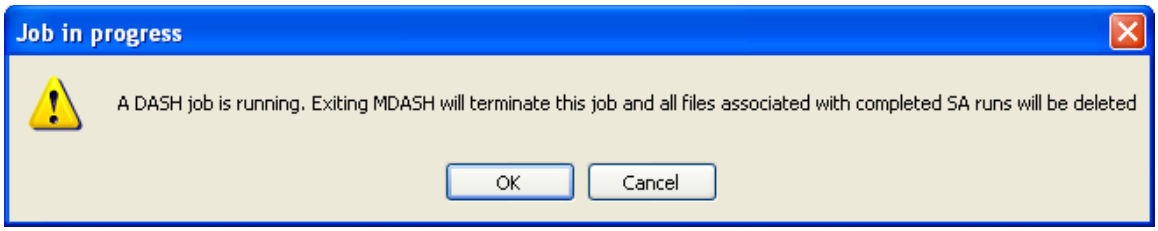


Figure 10: The 'exiting *MDASH*' warning dialog.

The *MDASH* results file

As mentioned in the 'Running *MDASH*' section, the merged '.dash' file is written to a subdirectory of the one containing the '.grd' file. This allows multiple *MDASH* runs to be started in the same working directory if required. The subdirectory is assigned a unique name in order to allow it to be easily identified and sorted. It takes the form

<rootname>_<date in 'year month day' format>_<time in 24 hr clock format>

For example, were a job started using 'famb.grd' to complete at 11:23 am on the 22nd of January 2008, then the subdirectory name would be 'famb_20080122_1123' and the results filename would be 'famb_merged.dash'.

Useful hints regarding processors

From the *MDASH* point of view, it is the number of *physical* processing units (cores) in your PC that is important, as each physical core is capable of executing a *DASH* job at effectively the same speed as a single dedicated CPU. Thus one can expect *DASH* jobs executed using *MDASH* to run approximately *n* times faster on an *n* core CPU than they will on a single core CPU, all other factors being equal. By way of example:

Description	Example	# of physical cores
Single CPU, single core	Pentium 4	1
Two CPUs, single core each	Motherboard with two Xeons	2
Single CPU, two cores	Core 2 Duo	2
Single CPU, four cores	Core 2 Quad	4

If you are at all unsure about the capabilities of your CPU, then you can use the freely available CPU-Z program (www.cpuid.com) to query your system. Note that Hyper-Threaded processors can give the illusion of having more one core e.g. a Pentium 4 with Hyper-Threading enabled is treated by the operating system as two processors instead of one. It is important to realise that the second processor is *logical* rather than physical. If two *DASH* runs are started on a Hyper-Threaded processor which has only one physical core, they will each run at about 50% of the speed of a single *DASH* job, giving little or no gain in the net execution time.