

Computer Science 415.340

Operating systems

MANAGEMENT

JOB SCHEDULING NOW

This chapter is mostly taken from a fairly recent article^{MAN3}, and it has two functions. The primary function, and the reason for its appearance among the other material about scheduling, is as a real example of some of the issues we've been discussing. The secondary function is to make the point that not only is batch processing still alive and well, but that people are still producing (and selling) new software for its administration. Batch processing is still recognised as the best way to get a lot of work done quickly and efficiently; looking back to the end of the previous chapter, it really does make scheduling easier if you can get the people off the system.

We should perhaps state that we have no personal knowledge of any of the software mentioned, other than what we have learnt from this and similar articles. If you buy it, don't say we told you to.

Here are some definitions, in case you need them :

Job scheduling : Organising a set of jobs for batch processing over a given period of time.

Batch window : The period during which a processor is used in batch mode, typically overnight or the weekend.

Documentation : Essentially specifications for a command file.

Management by exception : Only report the unexpected events – that is, those for which appropriate responses haven't been predefined.

Forecasting : Much batch work is repetitive – the same job is run daily, or weekly, or whatever. The behaviour is usually much the same each time, so statistics gathered on previous runs can be used to predict reasonably reliably how it will behave this time.

Simulation : Given the control file and a log of what happened, you can work out the sequence of events without actually reexecuting everything. This can often show up strange behaviour and help to track down things that went wrong.

Master-slave architecture : Using a separate controlling machine to coordinate the activities of several computers. (The coordination might be essential if a job running on one computer needs output from a job running on another.)

Peer-to-peer architecture : Another way to coordinate the activities of several machines, using controlling software running in one of the participating machines.

Users seem to have an insatiable appetite for fresh information and this demand is squeezing the batch window into a tight corner. Operators may argue that never has so much been demanded of so few, but specialist tools are available to ease the strain. Job-scheduling packages, in particular, can help operators hit their deadlines.

The traditional job-scheduling package is well-established – a kind of specialist calculator that sorts out the order in which jobs have to be run for any given period at any given time.

Today, however, operators want these packages to be a lot more flexible when they are compiling and running a schedule. Flexibility is needed because of the increasing number of ad hoc jobs

requested by users, which need to be fitted in at short notice between regular tasks. How quickly the user's job has to be carried out is determined by service-level agreements already set up at the data centre.

The package must be able to put jobs in the right order. For example, the payroll job can't run until all the data about employee attendance is in place. The package must also recognise the standalone properties of an ad hoc report, say, and shoot that in wherever possible.

Legent, Boole & Babbage, IBM and Computer Associates and Altai are among the suppliers offering packages to build documentation about each job. These not only detail what the job is designed to do but also outline what steps it takes to achieve this. Directions on where to restart the program if it falls over during one of these steps, or who to contact, can be built into the documentation. The system can then follow these problem-solving instructions and cut down on mundane and time-wasting tasks for the operator.

Legent's Jobtrac has built-in voice technology which enables the system to call the operator at home or on a mobile phone when an error crops up. It then reads out options for what it should do next and the operator merely presses in the selection on a touch-tone phone. This could be a welcome feature if you're sick of being dragged out of bed for practically no reason when you've left the system running.

The latest in job-scheduling packages can itemise tasks in great detail. Rather than waiting for one job to be finished before starting the next, you can set triggers for one job to start when, say, step six of a previous job has been completed. Triggers may also be set on the basis of information coming from outside the mainframe. This could be data held on machines in different environments that has been uploaded by shops at the end of the day's trading.

Most suppliers claim to provide management by exception – a useful feature that automatically alerts the operator to irregularities or problems in the system.

Another feature found in the new generation of job schedulers is a mechanism to plan around all the

system's resources to make the most of all the equipment. Whether you call it work-load balancing or resource management, it's beneficial to share the processing power round the entire schedule rather than just working on the principle of allowing each job to consume maximum power. Another way to refine the schedule is to use a forecasting function. This allows the operator to simulate the flow of a planned schedule based on historical data about jobs. If it highlights problems, these can be tackled before it's too late, and if operators feel something more could be done, they can change the schedule accordingly. Alternatively some packages, such as ControlM, use the simulation function to model the flow of a schedule that ran, say, yesterday. This gives the user a chance to examine what went wrong more closely.

Job scheduling obviously doesn't run in isolation, but is one part of the entire process. As such it needs to communicate with other programs. Although all suppliers design their systems to run with other suppliers' programs, there is a limit to the depth of integration. IBM, Legent, Computer Associates and Boole & Babbage all have their own set of production tools, so it's not surprising they trumpet the benefits of the tighter integration available between their own systems. It's not just a marketing ploy, though, and it's worth examining the benefits a complete suite will offer – namely, that data is less likely to be duplicated and, better still, that information received in one part of the system will pass automatically to any other part.

IBM's SystemView architecture – should it ever be fully implemented – will change this by offering everyone a common repository of data with a standard interface to all other programs.

Another prime concern for data centre staff is scheduling jobs over several mainframes that may all be running different operating systems. Such problems are common thanks to the mergers, acquisitions and downsizing programmes that have changed the face of the computer industry. Suppliers are beginning to address this by bringing out versions of job-scheduling software to run on more platforms. But that still doesn't solve the problem of how all the systems will be co-ordinated. Boole & Babbage is trying to overcome this

obstacle with its Enterprise-Wide Control system. It manages the separate boxes through another standalone (Unix) machine, which provides a graphical display of what's going on. Broadly speaking, this uses a master-slave architecture.

Other manufacturers, such as Legent, favour a peer-to-peer architecture, where any one of the systems on that network can take on overall control of scheduling. However, they have no products to implement this as yet. In the meantime, links to other systems can usually be designed as a short-term solution by most vendors.

Another innovation uses PCs to give better graphical representation of what is happening on the job-scheduling system. Legent offers a PC module, whereas Boole & Babbage has opted for the Unix workstation model. Through its PC product for OS/2, Computer Associates has taken the idea in a slightly different direction, designing a totally PC-oriented package that schedules jobs across a LAN.

Whatever your configuration is – the message from vendors is they are willing to take on cross-platform scheduling and strive to make it more flexible by building in enough intelligence for you to automate decisions and procedures.

REFERENCE

MAN3 : *Computing*, 13 May 1993, page 22.
