

## Running Distributed Schrödinger Jobs

A number of Schrödinger products can distribute work over multiple processors. There are several algorithms for performing the distribution. Some jobs divide the input structures into batches. Each batch is then submitted to a processor for execution as a subjob. The number of processors used and the number of batches (subjobs) can be different. Others divide the work among the available processors, so that the number of subjobs and the number of processors is the same.

Jaguar and QSite can use multiple processors, but use MPI parallel processing rather than distributed processing. Desmond uses MPI parallel processing, but also distributes multiple simulations.

For each distributed job there is a master job that is responsible for dividing up the work, submitting the subjobs, and collating the results. Thus, if  $N$  processors are requested, the job creates  $N+1$  processes. Depending on the program, this master job can run on the local host or on the remote host. If the master runs on the remote host, there are two separate cases to consider:

- The remote host (or collection of remote hosts) does not involve a queuing system. Even though the master job does not take much time, the overall job does not necessarily run inefficiently, because the operating system can swap the master job out when it is idle and use the time for other processes.
- The remote host is running a queuing system. The master job is run as a separate queued job, and the subjobs are then submitted to the queue from the master job. The master job can then occupy a processor that will be idle most of the time. The exception is Prime loop refinement and Phase, for which the master job runs a subjob locally.

Information on the number of subjobs, number of CPUs, and the master job location is given in [Table 1](#) for running distributed jobs from the command line, and in [Table 2](#) for running distributed jobs from Maestro.

*Table 1. Distributed processing behavior for jobs submitted from the command line.*

Product	Program	Number of subjobs determined by	Number of CPUs for subjobs determined by	Master job location
Epik	para_epik	-NJOBS <i>njobs</i> , -JOBCTS <i>maxstructs</i>	Number of hosts in -HOST list	First host in -HOST list
Desmond	multisim	-maxjob <i>njobs</i>	-cpu <i>cpu-spec</i> or .cfg file; value is per subjob.	Host specified by -HOST
Glide	glide	-NJOBS <i>njobs</i>	Number of hosts in -HOST list	Host specified by -DRIVERHOST; default first host in -HOST list
	para_glide	-n <i>njobs</i>	Number of hosts in -HOST list	Local host
Induced Fit Docking	ifd	Number of ligands (Glide), number of poses (Prime).	NUM_GLIDE_CPUS, NUM_PRIME_CPUS keywords in input file	First host in -HOST list
Ligand & Structure-Based Descriptors	lsbd	Keywords in input file	Keywords in input file	Local host
LigPrep	para_ligprep	-NJOBS <i>njobs</i> , -JOBCTS <i>maxstructs</i>	Number of hosts in -HOST list	First host in -HOST list
MacroModel	para_bmin	-NJOBS <i>njobs</i> , -JOBCTS <i>maxstructs</i>	Number of hosts in -HOST list	First host in -HOST list
MCPRO+	mcpro_ddg	Number of lambda points $\times 2$	Number of hosts in -HOST list	Local host
	mcpro_fep	Number of lambda points	Number of hosts in -HOST list	Local host

Table 1. Distributed processing behavior for jobs submitted from the command line. (Continued)

Product	Program	Number of subjobs determined by	Number of CPUs for subjobs determined by	Master job location
	mcpro_lrm	Number of ligands	Number of hosts in -HOST list	Local host
Phase	phase_dbsearch phasedb_confsites	Number of processors	Number of hosts in -HOST list	First host in -HOST list
Prime	multirefine	Stage of process, MAX_JOBS keyword in input file	Host given by HOST keyword in input file, otherwise host given by -HOST	First host in -HOST list
QM-Polarized Ligand Docking	qpld	-NJOBS <i>njobs</i>	Number of hosts in -HOST list or -host_program lists.	Host specified by -DRIVERHOST; default first host in -HOST list
SiteMap	sitemap	Number of processors	Number of hosts in -HOST list	Local host
Virtual Screening Workflow	vsw	-NJOBS <i>njobs</i> , -adjust	Number of hosts in -HOST list or -host_program lists.	Host specified by -DRIVERHOST; default first host in -HOST list

Table 2. Distributed processing for jobs submitted from Maestro.

Product	Job type	Number of subjobs set in or determined by	Number of CPUs set in or determined by	Master job location
CombiGlide	Docking	Start dialog box	Start dialog box	Local host
Epik		Number of CPUs (passed as -NJOBS)	Start dialog box	Selected host
Glide	Docking	Start dialog box	Start dialog box	Selected host
Induced Fit Docking		Number of ligands (Glide), number of poses (Prime).	Induced Fit Docking panel, Job options section.	Local host
Liaison	Simulation	Equal to number of CPUs	Systems folder, Liaison panel	Local host
LigPrep		Number of CPUs (passed as -NJOBS)	Start dialog box	Selected host
LSBD	Liaison	Number of CPUs	Advanced Options dialog box	Local host
	Prime MM-GBSA	Advanced Options dialog box	Advanced Options dialog box	
MCPRO+	Relative Binding Affinity by FEP	Number of lambda points $\times 2$	Start dialog box	Local host
	Free Energy Difference by FEP	Number of lambda points	Start dialog box	Local host
	Linear Response	Number of ligands	Start dialog box	Local host
Phase	Clean Structures, Generate Conformers, Find Matches	Number of CPUs	Start dialog box	Selected host

Table 2. Distributed processing for jobs submitted from Maestro.

<b>Product</b>	<b>Job type</b>	<b>Number of subjobs set in or determined by</b>	<b>Number of CPUs set in or determined by</b>	<b>Master job location</b>
Prime	Loop Structure	Depends on stage of process and sampling method	Refine Structure - Options dialog box	Selected host
QM-Polarized Ligand Docking		Number of CPUs	Start dialog box	Selected host
SiteMap		Number of CPUs	Start dialog box	Selected host
Virtual Screening Workflow		Start dialog box	Start dialog box	Selected host

The launch directory must be available from the master job host for Induced Fit Docking, Liaison and LSBDD.