Tips for Faster MOVES Runs

This document is meant to provide tips and techniques that modelers can leverage to improve MOVES run times. In summary, this includes:

- Reducing the amount of work MOVES needs to do. Selecting unnecessary pollutants or detail slows down a run.
- Splitting up one large MOVES run into several smaller runs. In some cases, several smaller runs performed sequentially may finish 5-25% faster than one large run while still producing the same results.
- Performing MOVES runs in parallel. If you have access to multiple computers, virtual machines, or cloud instances, you can perform the several smaller MOVES runs at the same time (instead of sequentially) to significantly reduce overall run time.
- System configuration and computer hardware can also have an impact on MOVES performance.

It is important to note that MOVES performance depends on many factors, including run parameters, system hardware, MariaDB configuration, and any concurrent processes that a computer is running. Therefore, while these tips serve as a guide for how to speed up MOVES, we are unable to provide precise information on how much run time improves with each technique, or even guarantee that any specific approach will improve performance for all modelers.

Most of these tips apply for both onroad and nonroad runs, except where noted.

1 Background Information

This section defines terminology and explains the basic concepts of how MOVES works, which will be useful to keep in mind when reading the rest of this document.

1.1 Definitions

RunSpec — A run specification file (RunSpec, traditionally given the *.mrs* extension) specifies the modeling domain of the run (time spans and geographical), the vehicle and road types to be included, and the specific pollutants and emission processes that should be modeled. It also includes names of input and output databases, and the level of detail that should be included in the modeling results. RunSpecs are created using the MOVES Graphical User Interface (GUI).

MOVES Main — The primary process started when you execute a RunSpec from the GUI or call `ant run` from the command line.

MOVES Worker — A helper process that assists MOVES Main in performing MOVES calculations.

Bundles — The MOVES Main and MOVES Worker processes communicate via bundles. Bundles are files that contain data and instructions. There are several kinds of bundles:

- *TODO Bundles* are created by MOVES Main and await processing by a MOVES Worker.
- *IN PROGRESS Bundles* have been claimed by a MOVES Worker and are currently being worked on.

- *DONE Bundles* have been finished by a MOVES Worker and await aggregation by MOVES Main.

1.2 Default MOVES Configuration

The following occurs when you execute a RunSpec from the MOVES GUI:

1. MOVES Main will launch, analyze the RunSpec, and generate bundles of work to do.
2. Concurrently, one MOVES Worker will launch and begin to process the bundles from MOVES Main.
3. MOVES Main will pick up the bundles processed by the MOVES Worker and aggregate the results into the output database.

1.3 Limitations

1.3.1 There Can Be Only One MOVES Main Per Run and Per Computer

There can be only one MOVES Main per MOVES run. Additionally, if you are running MOVES on...

- One computer: there can be only one MOVES Main process running at a time per computer.
- Networked virtual machines: there can be only one MOVES Main process running at a time per virtual machine.
- Cloud instances: there can be only one MOVES Main process running at a time per instance.
- A supercomputer cluster: there can be only one MOVES Main process running at a time per node in the cluster.

1.3.2 There Can Be Multiple MOVES Workers

You can have multiple MOVES Worker processes running at the same time per computer, virtual machine, instance, or node. These are not tied to an individual MOVES Main. That is, MOVES Worker processes can run independently of MOVES Main. However, the following conditions must be satisfied for the MOVES Main and MOVES Worker processes to communicate with each other:

- They must be running the same version of MOVES.
- They must have access to the same file storage (e.g., hard drive or shared drive).

2 RunSpec Considerations

If possible, reducing the amount of work that MOVES needs to do is an effective technique for improving MOVES run time. The table below summarizes the available strategies to reduce the amount of work that MOVES needs to do.

#	STRATEGY	APPLIES TO…	EFFECTIVENESS
1.	Remove unnecessary pollutants	All scales	High
2.	Select only the needed output detail	All scales	Varies
3.	Provide drive cycles or operating mode distribution inputs	Project Scale	Medium

2.1 Remove Unnecessary Pollutants

Each pollutant and process included in a MOVES run adds additional calculations which take additional time. Some pollutant and process combinations require significantly more calculations than others. For

limited-scope runs (e.g. analysis of a single hour), this may not lead to a performance issue, but it can have a significant impact on run time for large-scope MOVES runs.

Therefore, it is good to consider which outputs are necessary for your purposes before making selections in a RunSpec. Where possible, remove unnecessary pollutants and processes.

2.2 Select Only the Needed Output Detail

The more detail that you tell MOVES to retain, the more data MOVES must copy throughout a run. Additionally, the output data aggregation steps become more complicated and take longer to perform when more detail is requested. Therefore, select only the level of detail needed for your analysis. The more output detail that can be aggregated together, the faster MOVES will run.

Each checkbox selected on the Output Emissions Detail panel expands the number of rows in your output database. The exact increase depends on the other selections made throughout the RunSpec, but in general:[1]

- Selecting Source Use Type increases the number of output rows by the number of source types selected in the run (13x if all source types are selected).
- Selecting Model Year increases the number of output rows by 41x because MOVES calculates emissions for 41 model years.
- Selecting Fuel Type increases the number of output rows by 5x because MOVES includes 5 fuel types.
    - Note: If fuel type is selected along with source type, this only increases the number of output rows by ~4x because most source types have 4 applicable fuel types.
- Selecting Emissions Process increases the number of output rows proportional to the number of processes selected, though this varies substantially depending on the specific pollutants included in the run. Most pollutants have running, start, and hotelling emission processes, which encompass 7 specific processes. Hydrocarbons and related pollutants also include evaporative and refueling emission processes, which encompass 5 specific processes. Therefore, selecting Emissions Process typically increases the number of output rows by ~7-12x.
- Selecting Road Type typically increases the number of output rows by 5x because MOVES has 5 road types.
- Selecting Regulatory Class increases the number of output rows by 9x because there are 9 regulatory classes in MOVES.
    - Note: If regulatory class is selected along with source type, this only increases the number of output rows by ~3x because most source types have between 2-4 applicable regulatory classes.
- Selecting SCC automatically includes road type, source type, fuel type, and emission process details.

The selections in the “Output Aggregation” section of the Output Emissions Detail panel are also important.

[1] This is list for onroad output only. The logic for nonroad runs is similar, but the available output details and associated number of rows are different.

- If you are running all hours, but do not need hourly emissions detail, selecting “24-Hour Day” or “Portion of Week” will reduce the number of output rows by a factor of 24.
    - Note: Selecting “24-Hour Day” will result in emission inventories for a typical weekday and a typical weekend day (if both day types are selected). Selecting “Portion of Week” will result in emission inventories for a typical 5-day week and a typical 2-day weekend. The selection between “24-Hour Day” and “Portion of Week” should have no impact on the number of rows or MOVES overall performance.
- If you are running both day types, but do not need to differentiate between emissions on weekdays vs. weekend days, selecting Month will further reduce the number of output rows by a factor of 2.
- If you are running all months, but need only annual emission inventories for your purposes, selecting Year will further reduce the number of output rows by a factor of 12.

In addition to improving MOVES performance, selecting only the level of detail needed for your analysis purposes will simplify your MOVES output post-processing.

2.3 Provide Drive Cycles or Operating Mode Distribution Inputs

In Project Scale, link activity can be provided as an average speed, a drive schedule, or an operating mode distribution.

- If average speed is provided, MOVES will look for two bounding representative drive schedules: one that has an average speed below the link speed, and one that has an average speed above the link speed. MOVES will calculate operating mode distributions for both driving schedules and then weight the results together to get a representative operating mode distribution for the link. MOVES will do this calculation separately for each source type, regulatory class, and model year group operating on the link.
- If a drive schedule is provided, MOVES will calculate one operating mode distribution (resulting from that drive schedule) for each source type, regulatory class, and model year group operating on the link.
- If operating mode distributions are provided, MOVES will directly use them.

These calculations are performed by MOVES Main for each link, so they can be a bottleneck for Project Scale runs with many links. Providing either the drive cycles or operating mode distributions for each link allows MOVES to skip some of those calculations, which will improve project scale run times. However, it is important to note that user-supplied operating mode distributions vary by only source type—there is no field in the user input table to allow them to vary by regulatory class or model year. Despite this, vehicles in different regulatory classes and model years have different physical properties (e.g., vehicle mass or aerodynamics), which impact the operating mode distribution. Therefore, it may be more accurate to have MOVES calculate the operating mode distribution for you, since MOVES will calculate a separate distribution for each regulatory class and applicable model year group.

This potential optimization is unique among the recommendations presented here in that the MOVES output emissions for an otherwise identical Project Scale run will differ based on whether you provide average speeds, drive schedules, or operating mode distributions. It is important to select the input that best meets the needs of your analysis and consider performance only as a secondary factor.

3 Split up MOVES Runs

MOVES does many of its calculations in SQL. The execution time of SQL operations can be highly dependent on the number of rows involved in the calculation. For example, the calculation time of some join operations increases exponentially with the number of rows involved in the join (especially if the data in question takes up more space than there is available RAM). This can have a significant impact, particularly when some of the intermediate tables in MOVES contain millions of records.

Therefore, splitting a single large MOVES run into several smaller ones can reduce the overall MOVES run time required to get results while returning the same output. This works by reducing the size of each bundle. Smaller bundles can be created faster by MOVES Main and processed faster by a MOVES Worker.

This approach is only effective for onroad analyses at County Scale and Default Scale. It is not particularly effective for Project Scale or nonroad runs. Note that for County Scale runs, all of the smaller, split RunSpecs can use the same input database because MOVES allows input databases to have more data (i.e., more source types, months, and days) than would be used for an individual run.

As explained below, this approach requires additional effort to set up the runs and to consolidate the resulting output. See the MOVES Hands-on Training for additional information on how to process MOVES output.

A MOVES run is split up by creating multiple RunSpecs. All split RunSpecs should be identical except for the parameter that you are splitting the run by; that parameter should be unique in each RunSpec. In general, the best practice is to create a complete (that is, unsplit) RunSpec and develop any corresponding input databases with that RunSpec. Then, the split RunSpecs can be created by either:

- Using the GUI to create copies of the complete RunSpec that contain only the portion that you are splitting the run by. For example, if you are splitting by source type, you would remove all source types except for motorcycles in the first copy, delete all source types except for passenger cars in the second copy, etc.
- Use a text editor to create copies of the complete RunSpec that contain only the portion that you are splitting the run by. In the source type example, you would delete all `<onroadvehicleselection/>` entries except motorcycles in the first copy, delete all `<onroadvehicleselection/>` entries except for passenger cars in the second copy, etc. After using a text editor to create the copies, it is helpful to open each split RunSpec in the MOVES GUI to ensure it has all green checks. This will avoid getting errors later.

After a large MOVES run is split into several smaller runs, they can be performed sequentially (that is, one after another) on the same computer or in parallel (simultaneously) on multiple computers.

- If you are performing the runs sequentially on the same computer, it is helpful to run MOVES in batch mode. This allows you to perform a single command line operation to run multiple RunSpecs, reducing the need for manual intervention at the beginning and end of each run.
- See Section 4 (Running MOVES on a Network) for more information on running MOVES in parallel.

The table below summarizes the available strategies for splitting up runs.

#	STRATEGY	APPLIES TO…	EFFECTIVENESS RUNNING SEQUENTIALLY	EFFECTIVENESS RUNNING IN PARALLEL
1.	Split RunSpecs by source type	Default and County Scales	Medium	High
2.	Split RunSpecs by month and day	Default and County Scales	Medium	High
3.	Split RunSpecs by pollutant	Default and County Scales	Low	Medium
4.	Split RunSpecs by calendar year and county	Default Scale	Low	High

The effectiveness of splitting a RunSpec by some parameter relates to how that parameter is used at run time. Most bundles contain calculations for all source types, months, and days in the RunSpec, so bundle size can be reduced by splitting runs by source type, by HPMS type, or by month and day. On the other hand, not all bundles contain all pollutants in a RunSpec, and every bundle contains a single calendar year and county (assuming no geographic pre-aggregation is selected). This means splitting RunSpecs by pollutant or by calendar year and county will generally result in a smaller performance improvement than splitting by source type or by month and day.

3.1 Split RunSpecs by Source Type

One method for speeding up a large MOVES run is to split it into smaller runs where each smaller run includes only a single source type. This means that instead of modeling all source types in one RunSpec, 13 RunSpecs could be developed which are all identical except for the source type selected.

Alternatively, the RunSpec could be split to only include source types within a single HPMS class. This entails developing five RunSpecs with the following source type groupings:

1. Motorcycles
2. Passenger cars, passenger trucks, and light commercial trucks
3. Other buses, transit buses, and school buses
4. Refuse trucks, single unit short-hauls and long-hauls, and motor homes
5. Combination short-hauls and long-hauls

Splitting a RunSpec by HPMS class may not lead to the same performance gains as splitting by source type, but it is simpler as it involves performing fewer runs

When splitting up runs by source type, it is usually helpful to have source type detail in the output database (that is, select Source Type on the Emissions Output Detail panel). This ensures output from each run is identifiable with unique key fields.

3.2 Split RunSpecs by Month and Day

Another method for speeding up a large MOVES run is to split it into smaller runs where each smaller run includes a single month and day combination. For example, instead of modeling an annual inventory with one RunSpec, it can be modeled with 24 RunSpecs that are all identical except for the month and day selection. The first RunSpec would be for January weekends, the second for January weekdays, the third for February weekends, etc.

When splitting up runs by month and day, it is essential that the output aggregation is set to Hour, 24-Hour Day, or Portion of Week. When calculating an annual inventory, weekday emissions must be weighted differently than weekend emissions. When the output aggregation is set to Month or Year, MOVES aggregates the daily detail away, making this calculation impossible.

After performing runs in this manner, an annual inventory may be calculated from the results following these steps:

1. If the output aggregation is set to Hour, sum all the emissionQuant values across runs by pollutantID, dayID, and any other desired columns (such as sourceTypeID, for example). This results in daily emission inventories matching the MOVES output aggregation level of "24-Hour Day".
2. With a 24-Hour Day inventory, multiply the daily emissionQuant values by the dayID. This results in emission inventories that match the MOVES output aggregation level of "Portion of Week".
3. Sum the Portion of Week inventories together and multiply by the number of weeks in the month. The number of weeks in the month is calculated by dividing the number of days in the month by 7. The "MonthOfAnyYear" table in the MOVES default database contains the number of days in each month. This step results in monthly inventories, which would match the MOVES output aggregation level of Month.
4. Sum the monthly inventories together to get an annual inventory.

3.3 Split RunSpecs by Pollutant

This method of splitting up a run is less effective at improving performance than splitting a run by source type or by month and day. However, post-processing runs split by pollutant is generally the same as post-processing an unsplit run, which may be desirable.

When splitting RunSpecs by pollutant, it is important to keep all chained pollutants together in the same RunSpec. For example, both VOCs and methane require Total Gaseous Hydrocarbons as a prerequisite pollutant. If VOCs and methane are run in different RunSpecs, both runs will perform the same calculations for Total Gaseous Hydrocarbons. Furthermore, Total Gaseous Hydrocarbons will appear in the output for both runs, making it possible to accidentally double count those emissions during post-processing.

In general, we recommend following pollutant groupings:

- Total Gaseous Hydrocarbons and chained pollutants (such as VOC, NMHC, NMOG, CH_4, air toxics, PAH gases, metals, dioxins, and furans)
- PM and chained pollutants (such as most $PM_{2.5}$ species and PAH particles)
- NOx and chained pollutants (includes NO, NO_2, and HONO)
- Total Energy Consumption and chained pollutants (includes CO_2 and SO_2)
- Remaining unchained pollutants (includes CO, NH_3, and N_2O)

Note: manganese compounds, which are listed under $PM_{2.5}$ Species, are chained to Total Gaseous Hydrocarbons, not PM.

3.4 Split RunSpecs by Calendar Year and County

Generally speaking, there is not much performance to be gained by splitting a large RunSpec in this way because it doesn't impact bundle size. However, splitting a run like this is a natural choice when performing MOVES runs in parallel. Post-processing runs split up using this method generally follows the same steps as post-processing an unsplit run. To ensure output from each run is identifiable with unique key fields, make sure that the geographic output aggregation is set to County on the Output Emissions Detail panel.

4 Running MOVES on a Network

It is possible to significantly speed up a MOVES analysis by running MOVES in a networked environment. The table below summarizes such strategies.

#	STRATEGY	EFFECTIVENESS
1.	Perform MOVES runs in parallel on multiple computers, virtual machines, cloud instances, etc.	High
2.	Run MOVES with multiple MOVES Worker processes on networked computers or VMs	Low

4.1 Perform MOVES Runs in Parallel on Multiple Computers, Virtual Machines, Cloud Instances, etc.

It is not possible to run more than one MOVES Main process on one computer. However, if you have access to multiple computers, virtual machines, or a cloud computing environment, you can launch one MOVES Main process per environment, effectively allowing you to perform multiple runs simultaneously. When a large run is split up into several smaller runs and those smaller runs are performed in parallel, the entire processing time is equal to the longest individual run time in the group.

There are many different ways to manage computer networks and virtual machines and many different kinds of cloud computing environments available, so we are unable to provide specific, step-by-step instructions here on how to best perform MOVES runs in parallel. Additionally, EPA does not endorse, recommend, or favor any specific commercial product, process, or service.

The following sections contain general advice when running MOVES in parallel on multiple computers, virtual machines, or in cloud environments.

4.1.1 Install and Configure MOVES and MariaDB the Same Way in Each Environment

MOVES and MariaDB must be installed on each computer, virtual machine, or cloud computing environment. When performing runs in parallel, use the same versions of MOVES and MariaDB. Even small differences between patch versions of MOVES or MariaDB can lead to inconsistencies or incompatibilities between individual runs.

It is also helpful for installation paths and data directories to be in the same location in each environment, so that it is easier to use scripts to move data files and launch runs.

4.1.2 Set up Runs Using a Consistent Naming System for Output Databases

When splitting up a MOVES RunSpec for sequential runs, the individual RunSpecs should be identical except for the parameter that you are splitting the run by (e.g., source types). When splitting up a MOVES RunSpec for parallel runs, it is important that each RunSpec also has a unique output database

name. Each run performed in parallel will utilize its own output database. Therefore, it will not be possible to post-process the output data across all runs if the output databases all have the same name.

To reduce confusion later, take care to develop a consistent naming system when naming each RunSpec and its corresponding output database.

4.1.3 Each Environment Needs Shared Persistent Data Storage

You will need access to a shared persistent data storage solution in each environment. For example, this could be a shared network drive or data storage in the cloud. This is used to store RunSpecs and input and output databases.

Note that it is beneficial to move data to and store data in a shared persistent data storage solution using a compressed format. This reduces data transfer times and data storage space requirements.[2]

In general, a typical setup process includes the following steps:

1. Create a complete (that is, unsplit) RunSpec in the MOVES GUI and prepare its input database using the MOVES Data Manager.
2. Split up the RunSpec using one of the techniques described in Section 3. Save the split RunSpecs to the shared persistent data storage.
3. Run a "FLUSH TABLES" command in MariaDB to ensure all data are written to disk.
4. Locate your MariaDB data folder and compress the input database. Copy the compressed input database to the shared persistent data storage.

4.1.4 Use Scripts to Copy Data, Start Runs, and Clean Up

Starting individual runs in multiple locations can be tedious and prone to error. Therefore, it is best to use a batch file or a scripting language such as Python to automate the steps necessary to start a run. While these specifics vary across environments, such scripts will generally contain the following steps:

1. Copy the compressed input database from the shared persistent data storage to the local data storage, then uncompress to the MariaDB data directory.
2. Launch two or three MOVES Worker processes, e.g., `ant 3workers -Dnoshutdown=1`.
3. Launch MOVES Main via `ant run`.
4. When MOVES Main finishes:
    a. Terminate the MOVES Worker processes.
    b. Compress the output database directory and move it to the shared persistent data storage.
    c. Compress moveslog.txt, give the resulting file a descriptive name, and move it to the shared persistent data storage.

[2] MOVES uses MariaDB's MyISAM data storage engine. A MyISAM database on the file system consists of a single directory containing a db.opt file and three files per table in the database (an .frm, .MYD, and .MYI file per table). To compress a database, first run a "FLUSH TABLES" command in MariaDB to ensure all files are written to disk, then compress the database directory. For example, this can be done by selecting all the database files in Windows Explorer, right-clicking, and selecting Send to > Compressed (zipped) folder. However, it is important to note that these compressed databases cannot be used directly. For MOVES to use a database, or for you to post-process a database, it must be uncompressed in the MariaDB data directory. Compression is useful only when transferring databases or storing them in a shared location.

d. Delete the input database directory, output database directory, MOVESExecution database, and any MOVESWorker databases.
e. Delete moveslog.txt and any files remaining in the following MOVES directories:
    i. SharedWork
    ii. WorkerFolder
    iii. MOVESTemporary

4.1.5 Group Individual Output Databases to Simplify Post-Processing

One downside to running MOVES in parallel is that each MOVES Main must write to its own output database. This can greatly complicate post-processing MOVES output. To assist modelers in this situation, MOVES includes a tool called the "MOVES Output Grouper". This tool is currently in beta development, and it can be found in the Tools directory.

This tool combines MOVES results that are stored across multiple output databases into a single database, so that the results may be post-processed as if they were run sequentially on a single computer. Full instructions for the tool are available at MOVESOutputGrouper.md.

4.2 Run MOVES with Multiple MOVES Worker Processes on Networked Computers or VMs

As described in the background section, MOVES Main generates bundles which are processed by MOVES Workers. If the bundles are generated faster than a single MOVES Worker can process them, then MOVES run time can be improved by launching multiple MOVES Worker processes. Furthermore, MOVES Workers can run on separate computers from MOVES Main. To do so:

1. Install the same version of MOVES on all computers (or virtual machines, etc.).
2. For all installations, modify the sharedDistributedFolderPath setting in the following configuration files[3] to point to a common directory (e.g., \\sharedlocation\MOVESSharedWork):
    - maketodo.txt
    - manyworkers.txt
    - MOVESConfiguration.txt
    - WorkerConfiguration.txt
3. Along with starting a run, log onto the other computers and start the worker(s). See CommandLineMOVES.md for specific instructions on how to do this step.

When a run is started, MOVES Main reads the RunSpec and generates bundles of work. These bundles are named with a randomly determined MOVES Main ID (this changes with every run), the bundle number, and the label "TODO". They are saved in the sharedDistributedFolderPath.

Any MOVES Worker that is monitoring the sharedDistributedFolderPath will pick up "TODO" bundles. While it is working on the bundle, it will rename it to "InProgress" so that other workers don't pick up the same bundle. When it is done, the worker will rename it to "DONE" and then look for more bundles.

The MOVES Main process will look for "DONE" bundles with its ID, and then delete the bundle when it is done absorbing the data from it.

[3] MOVES configuration files are located in its root installation directory.

A few additional notes:

- Since MOVES labels bundles with a unique identifier for each run, multiple instances of MOVES can simultaneously use the same sharedDistributedFolderPath with the same pool of workers. However, each MOVES Main must be running on its own computer (or virtual computer, etc.).
- When starting MOVES Workers with the `-Dnoshutdown=1` flag, workers can run independently of MOVES Main. That is, a pool of workers can be started and left running for a series of MOVES runs.
- MOVES and its workers can be identified in the sharedDistributedFolderPath by their ID file. The timestamp on these files will update every minute to signify that the process is still running. If a file's timestamp stops updating, it is likely that the process has died.
- Only one instance of MOVES Main can be run at a time on a single computer.
- One or more MOVES Workers can run on a single computer.
- One instance of MOVES Main and one or more MOVES Workers can be run simultaneously on the same computer, but this is not required.
- For onroad runs, there are limits to the performance gains that can be achieved by adding more workers. Not all bundles generated in a MOVES run are the same size. Often a run will have many small bundles and a few very large ones which are the run's bottlenecks. If there are more workers running than large bundles, the small files will be processed quickly leaving extra workers idle while the large files are being processed. Since each active worker adds read and write operations that exacerbate bottlenecks caused by a slow hard drive, EPA generally recommends using only two or three MOVES Workers for each MOVES run.
- For nonroad runs, most bundles are approximately the same size. Therefore, adding more MOVES Workers will generally improve performance.
- The MOVES Overview Report contains more information on MOVES structure.

5 System Configuration Considerations

The table below gives an overview of the system configuration considerations that can be taken to improve MOVES run time, along with their relative effectiveness.

#	STRATEGY	EFFECTIVENESS
1.	Run MOVES on a computer with a solid-state hard drive	High
2.	Prevent antivirus applications from scanning MOVES temporary files	High
3.	Run MOVES on a computer with multiple logical processors	Medium
4.	Run MOVES with a clean database data folder	Medium
5.	Run MOVES on a partition or hard drive separate from the operating system	Low
6.	Run MOVES with multiple MOVES Worker processes on the same computer	Low (onroad) High (nonroad)

5.1 Run MOVES on a Computer with a Solid-State Hard Drive

MOVES calculations involve querying the default and input databases, creating temporary tables and files, doing table joins, and writing output data. These operations mean that the bottleneck for most MOVES runs is not CPU or RAM availability, but the read/write speed of the computer's hard drive.

Using hard drives with faster read and write speeds will therefore improve MOVES performance. Solid-state hard drives have read/write speeds that are much faster than traditional hard disk drives and are becoming cheaper and more common. This translates directly to faster MOVES runs.

Note: If your computer has multiple hard drives, install MOVES and MariaDB on the fastest drive.

5.2 Prevent Antivirus Applications from Scanning MOVES Temporary Files

If an active antivirus application must scan each temporary file that MOVES generates before it can be used, runs may take twice as long to complete as they normally would. In this case, a significant performance increase may be obtained by excluding the following directories from the antivirus application's checks:

- manyworkers
- MOVESTemporary
- NonroadTemporaryData
- sharedwork
- WorkerFolder

In the default configuration, all the above folders are found in the MOVES installation directory.

An alternative to adding an exclusion for these specific directories is to reconfigure MOVES to use alternative locations that are already not scanned by antivirus.

- manyworkers is set via the workFolderPath directive in manyworkers.txt
- sharedwork is set via the sharedDistributedFolderPath directive in maketodo.txt, manyworkers.txt, MOVESConfiguration.txt, and WorkerConfiguration.txt
- workerfolder is set via the workFolderPath directive in WorkerConfiguration.txt

Note that the MOVESTemporary and NonroadTemporaryData directories are not configurable.

MOVES executables should also be excluded from antivirus scans. The executables used during MOVES run time are:

- externalcalculatorgo64.exe
- externalgenerator64.exe
- java.exe
- nonroad.exe

In the default configuration, java.exe is located in the jre/bin subfolder of the MOVES installation directory. If MOVES is running from source, it is likely that java.exe is located wherever the Java JDK is installed. The other executables (externalcalculatorgo64.exe, externalgenerator64.exe, and nonroad.exe) are run from within subdirectories of manyworkers and workerfolder.

In addition to MOVES folders and executables, the MariaDB executables (mysqld.exe and mariadbd.exe) and data folder should also be excluded from the antivirus application's checks.

5.3 Run MOVES on a Computer with Multiple Logical Processors

MOVES is written to use multiple CPU threads and multiple SQL connections concurrently. Computers with few logical processors (1 or 2) cannot take advantage of this concurrency, but those with more logical processors can.

One way that MOVES takes advantage of concurrent processing is by utilizing separate MOVES Main and the MOVES Worker processes. See Section 5.6 below for more discussion regarding multiple MOVES Worker processes.

5.4 Run MOVES with a Clean Database Data Folder

Database servers work faster if they are maintaining less data. The more data a database engine is tracking, the more it uses the hard disk and CPU for overhead operations. Routinely archiving MOVES input and output databases to locations outside the database data folder helps minimize this overhead.

5.5 Run MOVES on a Partition or Hard Drive Separate from the Operating System

MOVES runs are demanding of disk usage, but many operating systems perform regular disk operations in the background. The operating system will always take priority over MOVES, causing MOVES to slow down.

Running MOVES on a drive that can handle many read/writes simultaneously and which doesn't have an operating system running on it can reduce competition and help MOVES run faster. However, this generally provides only a small improvement.

5.6 Run MOVES with Multiple MOVES Worker Processes on the Same Computer

This is related to running MOVES in a networked configuration with multiple MOVES Worker processes except this tip discusses launching multiple MOVES Worker processes on the same computer as MOVES Main.

As described in the background section, MOVES Main generates bundles which are processed by MOVES Workers. If the bundles are generated faster than a single MOVES Worker can process them, then MOVES run time can be improved by launching multiple MOVES Worker processes. The following commands can be used to launch multiple MOVES Workers on the same computer as MOVES Main:

- `ant 2workers -Dnoshutdown=1`
- `ant 3workers -Dnoshutdown=1`
- `ant manyworkers -Dmaxworkers=N -Dnoshutdown=1`

Notes:

- `-Dnoshutdown=1` is used to keep the MOVES Worker process alive. By default, the process will search for TODO bundles for a couple of minutes. If it doesn't find any work to do, it automatically quits. If `-Dnoshutdown=1` is used when starting workers, they will never automatically quit. Instead, you will need to manually shut them down by pressing CTRL-C in the terminal window when they are no longer needed.
- When passing `-Dmaxworkers=N`, `N` represents the desired number of workers.

- As we discuss in the section related to running MOVES workers in a networked configuration, there are limits to the performance benefit that is gained from running many workers for onroad runs. However, this is an effective technique for speeding up nonroad runs.