

ScalES-PPM - Feature #347

Add support for METIS v4-5 and ParMETIS v3-4

09/26/2017 03:13 AM - Matthew Krupcale

Status:	Feedback	Start date:	09/25/2017
Priority:	Normal	Due date:	
Assignee:		% Done:	100%
Category:		Estimated time:	0:00 hour
Target version:			

Description

Introduction

The current ScalES-PPM API appears to be designed around the METIS v4 and ParMETIS v3 APIs. The attached patch aims to add support for both METIS v4 and v5 as well as ParMETIS v3 and v4.

Summary of the patched files

- configure.ac: detect METIS and ParMETIS version; check idxtype/idx_t compatibility; check real_t compatibility
- include/f77/ppm.inc.in: define parameter PPM_REAL to be the width of real_t
- ppm.settings.in: define ParMETIS/METIS fcrealkind to be the width of real_t
- src/Makefile.am: define HAVE_METIS_V4 and HAVE_PARMETIS_V3 compiler preprocessor macros as needed; only in ParMETIS v3 include src/ppm/parmetis_wrap.c in libscalppm
- src/ppm/ppm_graph_partition_mpi.f90: adjust INTERFACE SUBROUTINE parmetis_v3_partkway C-binding for particular version of ParMETIS; use c_null_ptr instead of all combinations of dummy_{balance,weights}
- ppm_graph_partition_serial.f90: define INTERFACE SUBROUTINE metis_mcpartgraphkway, INTERFACE SUBROUTINE metis_setdefaultoptions, and INTERFACE SUBROUTINE metis_partgraphkway C-bindings for particular version of METIS; use c_null_ptr instead of all combinations of {v,e}w_dummy

Detailed changes

configure.ac

There have been several changes made between METIS v4 and v5 in particular, making configure.ac and ppm/ppm_graph_partition_serial.f90 inoperable with METIS v5. Some of the particular changes between METIS v4 and v5 are:

- Changed idxtype -> idx_t
- Unified routines (METIS_PartGraphKway, METIS_mCPartGraphKway, METIS_WPartGraphKway, METIS_PartGraphVKway, METIS_WPartGraphVKway) -> METIS_PartGraphKway (with a change in API)

ParMETIS v4 now also directly relies on METIS v5 and thus should not in principle have a different idx_t than METIS, unless the user explicitly builds METIS with IDXTYPEWIDTH 32 and IDXTYPEWIDTH 64, then builds a modified version of ParMETIS v4. This would have to be a modified ParMETIS v4 because it directly includes the metis.h, where idx_t is typedef'd. Obviously this is not a supported configuration, and I would hope that the user would not attempt this. Nevertheless, if the user does this, the patched configure.ac still checks both parmetis.h and metis.h for the idxtype/idx_t and ensures that they are compatible.

Similarly to idx_t, METIS v5 now defines real_t rather than float for some of the balancing constraints. Thus, the patched configure.ac also checks for real_t (again in both parmetis.h and metis.h, in case the user tries to do something strange) and makes sure they are compatible. Note that this relies on the patch in #343.

These are the major changes to configure.ac. In addition to the above changes, METIS_HEADER='metis/metis.h' was removed because I could find no reference to such a file in any of the METIS installations and assumed this must be some custom installation location. In this case, the metis.h header location should be specified in the configure CFLAGS instead. If desired I can break this change out into a separate bug/patch, but it was causing issues when configuring my METIS/ParMETIS setups.

src/Makefile.am

A Fortran to C wrapper for ParMETIS_V3_PartKway is included in src/ppm/parmetis_wrap.c since the ParMETIS v3 Fortran wrapper does not include the conversion from Fortran MPI communicator to C MPI communicator, while the v4 API does this properly (see macro FRENAME in parmetis-4.0.3/libparmetis/frename.c), so this wrapper is only included in libscalppm when using ParMETIS v3.

src/ppm/ppm_graph_partition_mpi.f90

ParMETIS_V3_PartKway function has the following prototypes in v 3.2.0 and v 4.0.3, respectively:

```
// ParMETIS v 3.2.0
void __cdecl ParMETIS_V3_PartKway(
    idxtype *vtxdist, idxtype *xadj, idxtype *adjncy, idxtype *vwgt,
    idxtype *adjwgt, int *wgtflag, int *numflag, int *ncon, int *nparts,
    float *tpwgts, float *ubvec, int *options, int *edgcut, idxtype *part,
    MPI_Comm *comm);

// ParMETIS v 4.0.3
int __cdecl ParMETIS_V3_PartKway(
    idx_t *vtxdist, idx_t *xadj, idx_t *adjncy, idx_t *vwgt,
    idx_t *adjwgt, idx_t *wgtflag, idx_t *numflag, idx_t *ncon, idx_t *nparts,
    real_t *tpwgts, real_t *ubvec, idx_t *options, idx_t *edgcut, idx_t *part,
    MPI_Comm *comm);
```

Thus, the types defined in the INTERFACE SUBROUTINE parmetis_v3_partkway had to be adjusted depending on the version of the library used.

The method for invoking parmetis_v3_partkway was also modified to require only one CALL site with the arguments passed depending on the presence of the OPTIONAL arguments to graph_partition_parmetis. In particular, several TYPE(c_ptr) were used and passed (by value) to the parmetis_v3_partkway Fortran C-binding rather than using an uninitialized dummy array. This scales much better with the number of optional arguments: for N optional arguments, there are 2^N combinations, while defining a pointer to either NULL or the location of the argument when present scales linearly in N .

Although balance is an OPTIONAL argument, tpwgts appears to be required in the ParMETIS v4 API (see parmetis-4.0.3/libparmetis/weird.c:CheckInputsPartKway), and thus balance is constructed with equal weights for each sub-domain/partition when it is not present.

ppm_graph_partition_serial.f90

METIS v4 has separate routines for multi-constraint partitioning (METIS_mCPartGraphKway) and single-constraint partitioning (METIS_PartGraphKway); in METIS v5, these two partitioning methods are unified into METIS_PartGraphKway (albeit with different API than in v4). METIS v5 also has a larger set of options and a corresponding function METIS_SetDefaultOptions to initialize it with defaults. Only METIS_OPTION_NUMBERING is set (corresponding to numflag in the v4 API) to indicate Fortran-style numbering.

The METIS_PartGraphKway function has the following prototypes in v 4.0.3 and v 5.1.1, respectively:

```
// METIS v 4.0.3
void METIS_PartGraphKway(int *nvtxs, idxtype *xadj, idxtype *adjncy, idxtype *vwgt,
    idxtype *adjwgt, int *wgtflag, int *numflag, int *nparts,
    int *options, int *edgcut, idxtype *part);

// METIS v 5.1.0
METIS_API(int) METIS_PartGraphKway(idx_t *nvtxs, idx_t *ncon, idx_t *xadj,
    idx_t *adjncy, idx_t *vwgt, idx_t *vsize, idx_t *adjwgt,
    idx_t *nparts, real_t *tpwgts, real_t *ubvec, idx_t *options,
    idx_t *edgcut, idx_t *part);
```

Thus, similar to in src/ppm/ppm_graph_partition_mpi.f90, the types and parameters defined in the INTERFACE SUBROUTINE metis_partgraphkway had to be adjusted depending on the version of the library used. Additionally, like in src/ppm/ppm_graph_partition_mpi.f90, c_null_ptr was used to indicate the absence of OPTIONAL arguments, and a single CALL site was used for all combinations of OPTIONAL arguments.

Supported combinations of METIS and ParMETIS

While this patch is designed for use with arbitrary combinations of METIS v4/v5 and ParMETIS v3/v4, in practice, when using ParMETIS, I expect this to only work when building ParMETIS with the internally-bundled version of METIS or with ParMETIS v4.0.3 and an internal or external METIS v5.

By default, it is not possible to build binaries linked against ParMETIS v3+ and external METIS v4 package because the ParMETIS v3 version of parmetis.c (modified from the METIS version) defines functions METIS_NodeRefine (needed by ParMETIS v3.2+) and/or METIS_mCPartGraphRecursive2 (needed by ParMETIS v3). So in order to use an external METIS v4 with ParMETIS v3, the METIS v4 parmetis.c would have to be similarly modified to define these functions. However, it is possible to build binaries linked against ParMETIS v4 and external METIS v5 since it defines METIS_NodeRefine. This should also be clear from the fact that ParMETIS v4 now uses and bundles METIS v5 directly.

In spite of still technically supporting METIS v4 and ParMETIS v3, the only combination really recommended is METIS v5 and ParMETIS v4. In testing (using example/graph_partition), METIS v4 seems to be unreliable, returning SIGABRT double free or corrupted double-linked list or SIGSEGV segmentation fault errors within METIS_mCPartGraphKway. Thus, the multi-constraint procedure in METIS v4 seems altogether un-usable. These errors do not appear in the METIS v5 version. Both of these codes are more than 5 years old at this point, so I would hope that this is not much of an issue.

Testing

Since my goal is to package ScaLES-PPM for Fedora, I am working on this on Fedora, and I have some Bash scripts which setup a mock chroot, build METIS v4 and ParMETIS v3/4, and build ScaLES-PPM with various combinations of these. Let me know if you are interested in trying out these tests.

History

#1 - 08/04/2020 08:07 PM - Thomas Jahns

- *Status changed from New to Feedback*

Resolved in commit:575b8e08ca part of release 1.0.6.

#2 - 08/04/2020 08:07 PM - Thomas Jahns

- *% Done changed from 0 to 100*

#3 - 02/07/2022 02:55 PM - Thomas Jahns

I did find some time to investigate the underlying problem some more and hopefully the latest release will improve support for different weights.

Files

scales-ppm-metis-4-5-parmetis-3-4.patch	22.7 KB	09/26/2017	Matthew Krupcale
---	---------	------------	------------------