Experiments for Assessing Computation/Communication Overlap of MPI Nonblocking Collectives

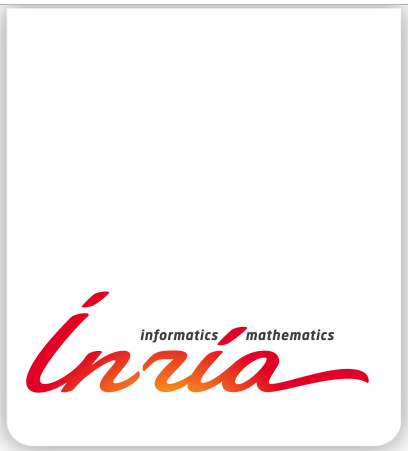
Alexandre Denis, Julien Jaeger, Emmanuel Jeannot, Florian Reynier

Experiments for Assessing Computation/Communication Overlap of MPI Nonblocking Collectives

Alexandre Denis Julien Jaeger Emmanuel Jeannot Florian Reynier

RESEARCH REPORT
N° 9367
October 2020
Project-Team TADaaM

ISSN 0249-6399 ISRN INRIA/RR--9367--FR+ENG



Experiments for Assessing Computation/Communication Overlap of MPI Nonblocking Collectives

Alexandre Denis* Julien Jaeger† Emmanuel Jeannot* Florian Reynier†

Project-Team TADaaM

Research Report n° 9367 — October 2020 — 36 pages

Abstract: We present experimental results for evaluating non-blocking MPI collectives. We compute several metrics to assess the efficiency of the overlap for different MPI library, with different configurations and for different hardware.

Key-words: MPI nonblocking collectives, evaluation metrics, experiments

* Inria Bordeaux — Sud-Ouest
† CEA/DAM/DIF

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Experiences pour évaluer le recouvrement calcul/communication des collectoves non-bloquates de MPI

Résumé : Nous présentons les résultats d'expériences pour l'évaluation des collectives non bloquantes dans MPI. Nous calculons différentes métriques pour évaluer l'efficacité du recouvrement pour différentes bibliothèques MPI avec différentes configurations et pour différents types de matériels.

Mots-clés : MPI, Collectives non bloquantes, métriques d'évaluation, expériences

1 Introduction

In order to execute efficiently parallel applications on large-scale supercomputers, such applications are often programmed using the Message Passing Interface (MPI) standard. MPI uses the SPMD (Simple Program Multiple Data) model to describe how each process composing the applications are communicating. MPI features diverse primitives such as point-to-point communication or collective operations. Collective operations are a set of abstract routines enabling a group of processes to execute in an efficient and concise way a sequence of point-to-point calls. Since the version 3.0 of the MPI standard, all the collective operations of MPI have a nonblocking version. This means that, when a nonblocking collective communication call is made, the data exchanges can continue in the background after the return of the call; the application can continue its execution until an explicit call to a completion function.

Using nonblocking collectives allows the application to execute independent computations between the collective initiation call and the completion call (*e.g.* `MPI_Wait`). Thanks to this, it should be possible to hide the communication behind computation and to save time through computation/communication overlap. However, for such overlap to effectively happen, it is required that the communications *progress* in the background while the application continues to execute its computation. Without this background progress, the MPI library is not able to execute the chosen collective algorithm and the data transfer can only happen when the application explicitly calls other MPI routines.

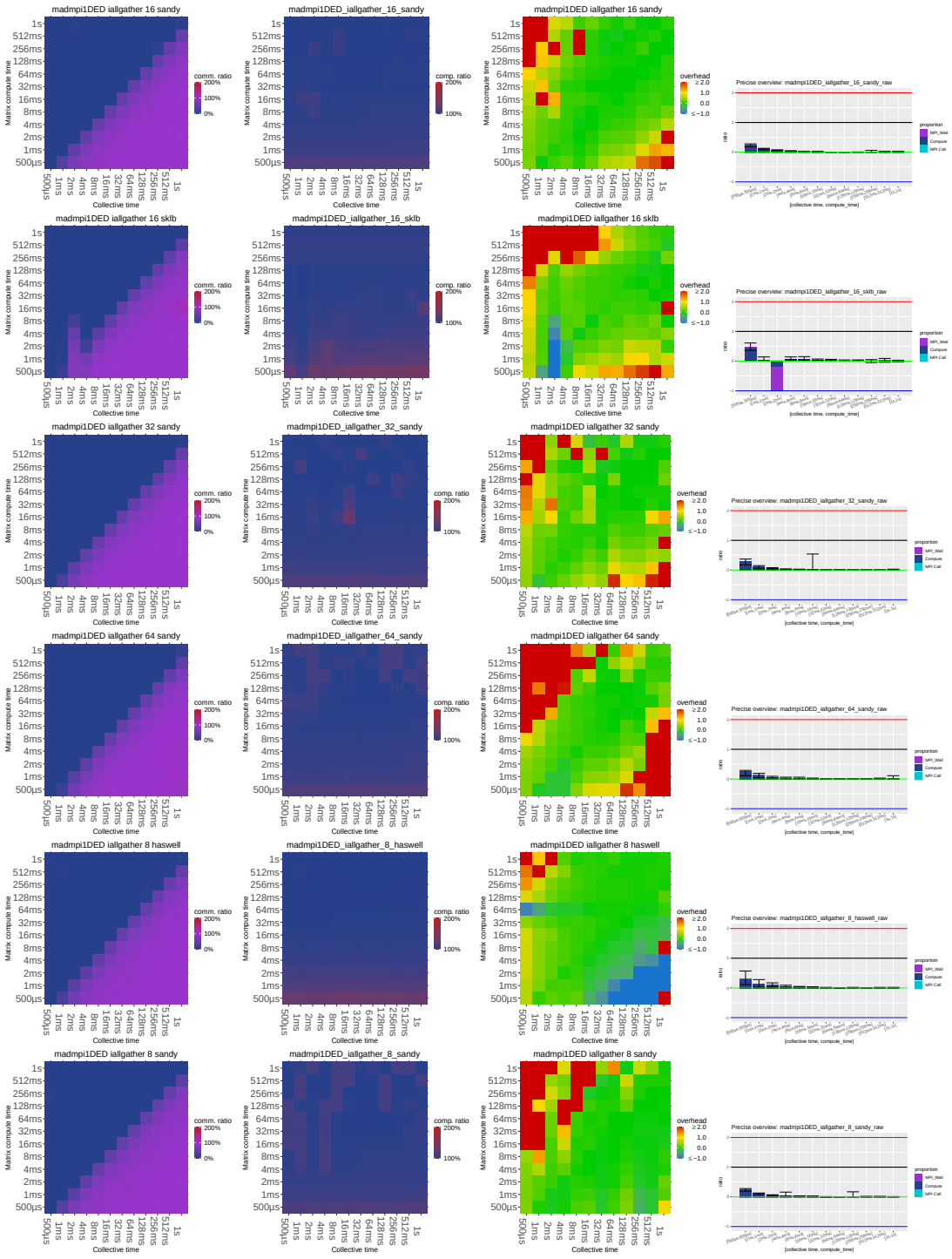
However, implementing a progression engine for collective operation is much more difficult than for point-to-point communication as such collective may require to implement a complex algorithm (*e.g.* diffusion tree) involving computation (*e.g.* for reducing values). This also requires to carefully dedicate resources to execute the engine. A good balance between what is devoted to the progression and what to the application computation is key for performance. Hence, MPI library developers are struggling to implement such efficient progress engine. Therefore, in the literature, up to now, few applications have been reported to use MPI nonblocking collective.

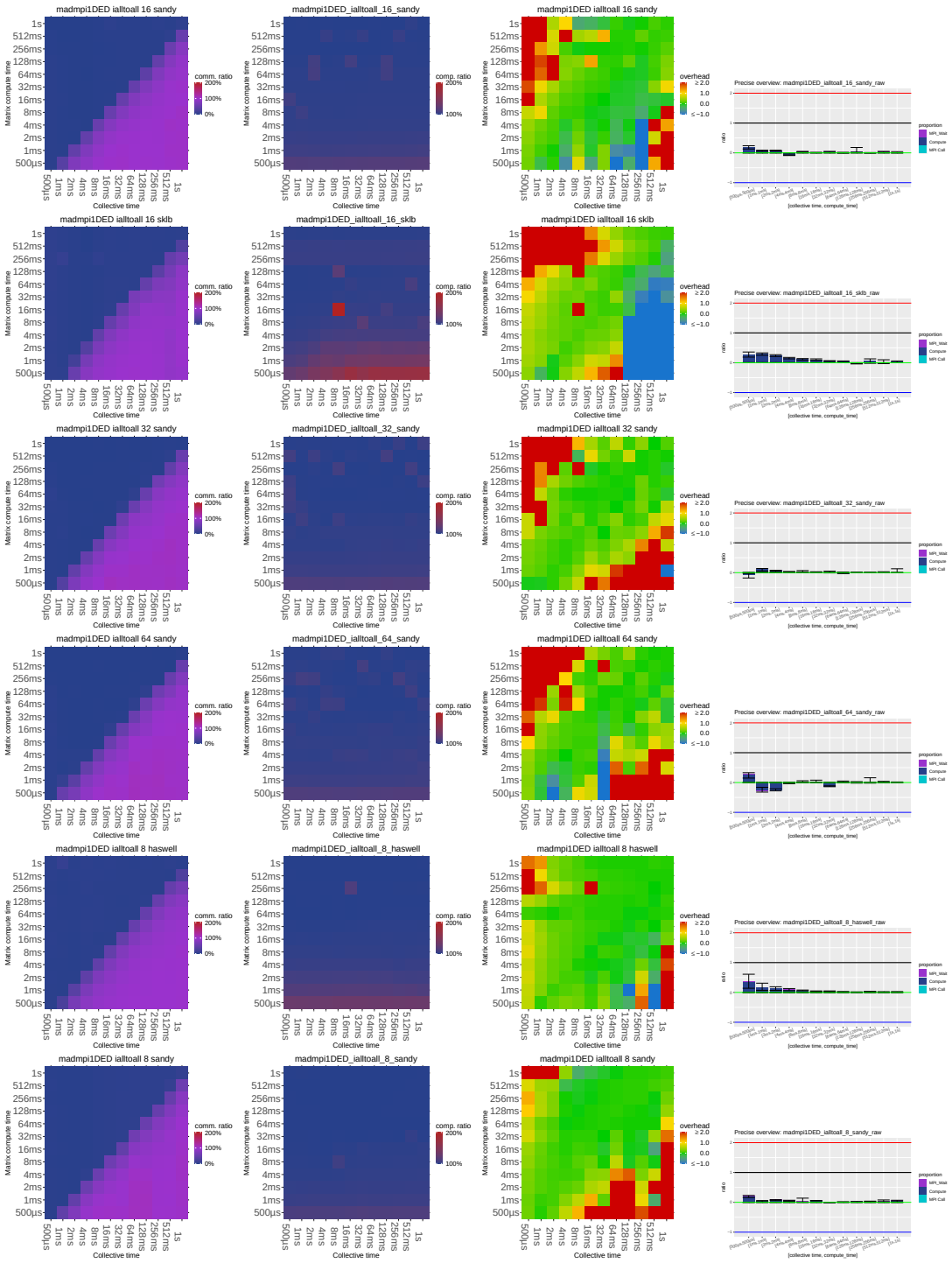
The goal of this report is actually to assess the quality of the progression engine of different MPI implementations in order to objectively determine if computation/communication overlap can efficiently be achieved by using nonblocking collective communications. Based on a set of metrics, we evaluate different MPI libraries on different nonblocking collective routine on several popular MPI implementations to evaluate how the communication/computation overlap behaves in such implementations.

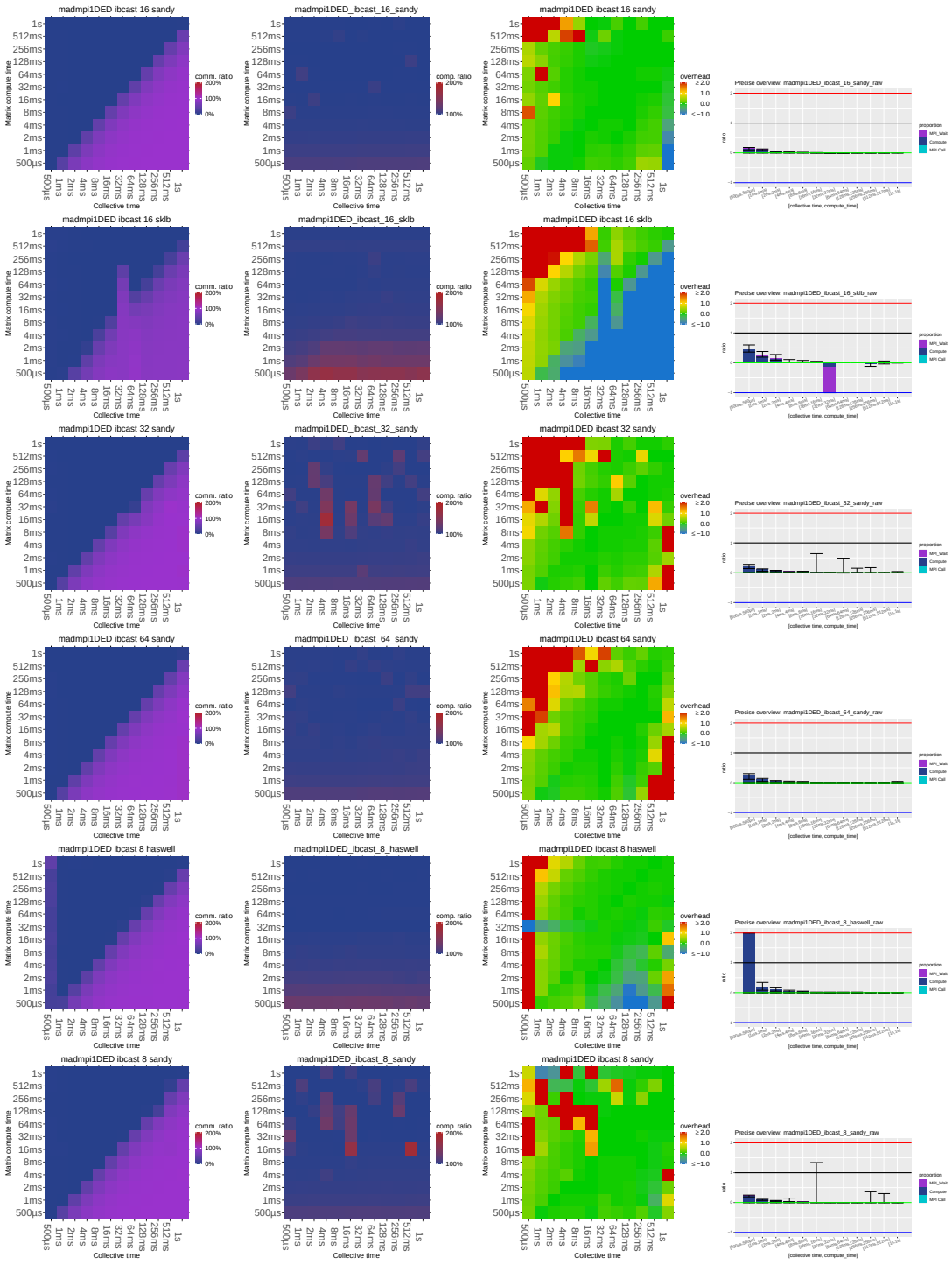
This report contains experiments of of a paper submitted to IPDPS 2020. Metrics presentatoon and discussion of the results are done in this paper.

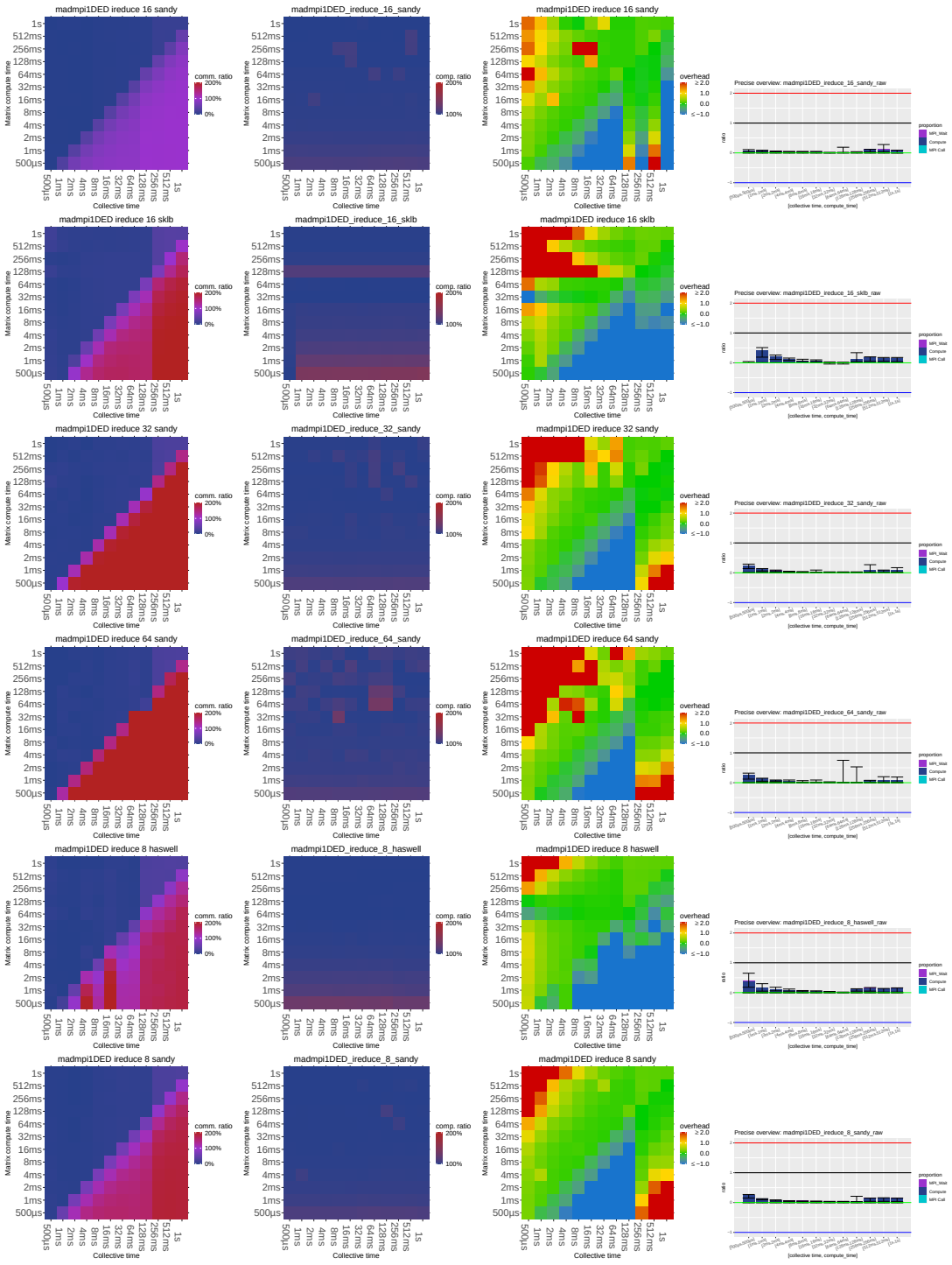
2 MadMPI with dedicated core

This section shows the results obtained with MadMPI using a dedicated core. The dedicated is enabled using the environement variable PIOM_DEDICATED=1. The network used is Infiniband.


madmpi1DED iallgather 16 sandy
madmpi1DED_iallgather_16_sandy
madmpi1DED iallgather 16 sandy
Precise overview: madmpi1DED_iallgather_16_sandy_raw
madmpi1DED iallgather 16 sklb
madmpi1DED_iallgather_16_sklb
madmpi1DED iallgather 16 sklb
Precise overview: madmpi1DED_iallgather_16_sklb_raw
madmpi1DED iallgather 32 sandy
madmpi1DED_iallgather_32_sandy
madmpi1DED iallgather 32 sandy
Precise overview: madmpi1DED_iallgather_32_sandy_raw
madmpi1DED iallgather 64 sandy
madmpi1DED_iallgather_64_sandy
madmpi1DED iallgather 64 sandy
Precise overview: madmpi1DED_iallgather_64_sandy_raw
madmpi1DED iallgather 8 haswell
madmpi1DED_iallgather_8_haswell
madmpi1DED iallgather 8 haswell
Precise overview: madmpi1DED_iallgather_8_haswell_raw
madmpi1DED iallgather 8 sandy
madmpi1DED_iallgather_8_sandy
madmpi1DED iallgather 8 sandy
Precise overview: madmpi1DED_iallgather_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
raw
proportion
MPI_Wait
Compute
MPI Call
[collective time, compute_time]

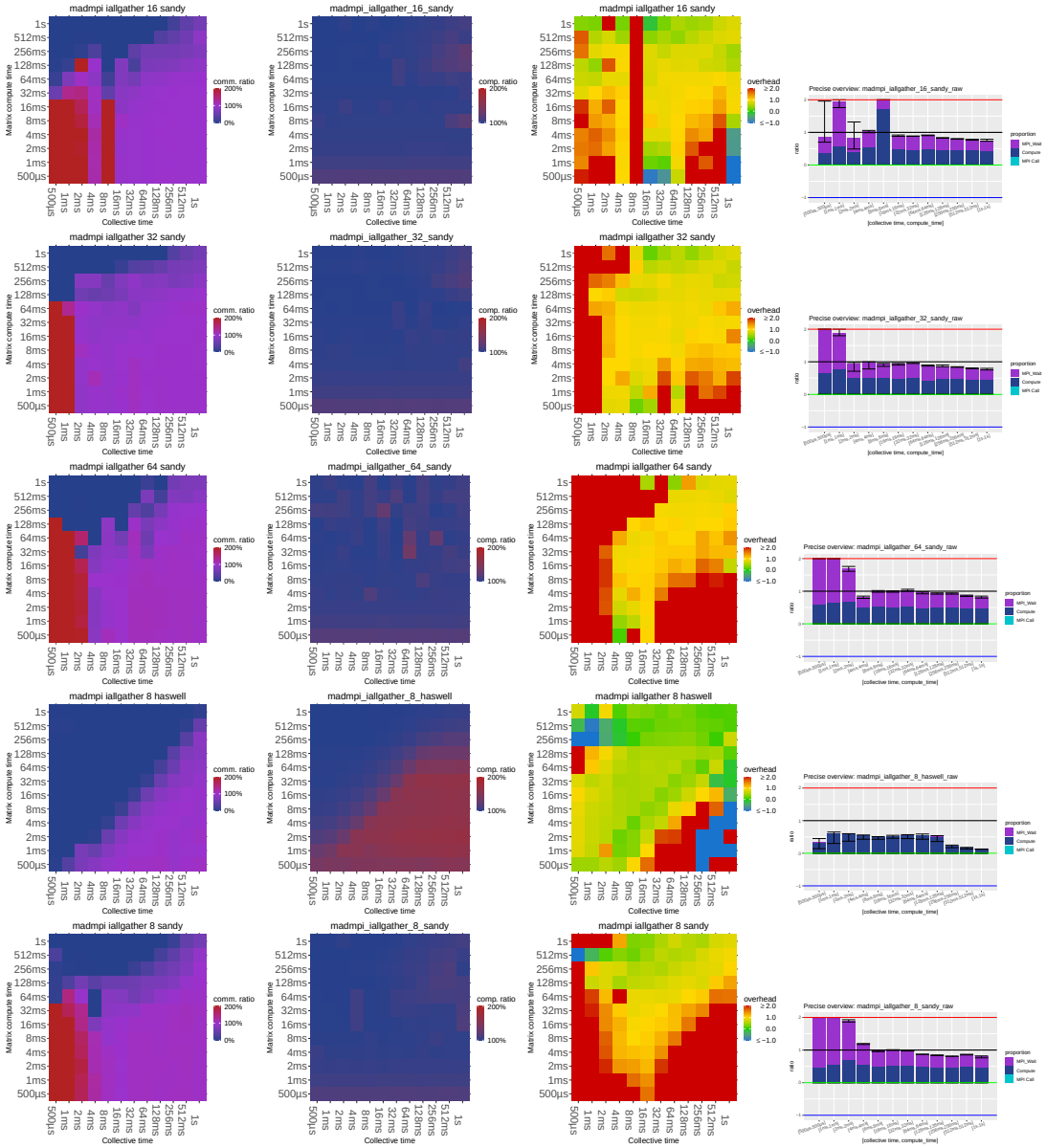

madmpi1DED ialltoall 16 sandy
madmpi1DED_ialltoall_16_sandy
madmpi1DED ialltoall 16 sandy
Precise overview: madmpi1DED_ialltoall_16_sandy_raw
madmpi1DED ialltoall 16 sklb
madmpi1DED_ialltoall_16_sklb
madmpi1DED ialltoall 16 sklb
Precise overview: madmpi1DED_ialltoall_16_sklb_raw
madmpi1DED ialltoall 32 sandy
madmpi1DED_ialltoall_32_sandy
madmpi1DED ialltoall 32 sandy
Precise overview: madmpi1DED_ialltoall_32_sandy_raw
madmpi1DED ialltoall 64 sandy
madmpi1DED_ialltoall_64_sandy
madmpi1DED ialltoall 64 sandy
Precise overview: madmpi1DED_ialltoall_64_sandy_raw
madmpi1DED ialltoall 8 haswell
madmpi1DED_ialltoall_8_haswell
madmpi1DED ialltoall 8 haswell
Precise overview: madmpi1DED_ialltoall_8_haswell_raw
madmpi1DED ialltoall 8 sandy
madmpi1DED_ialltoall_8_sandy
madmpi1DED ialltoall 8 sandy
Precise overview: madmpi1DED_ialltoall_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]

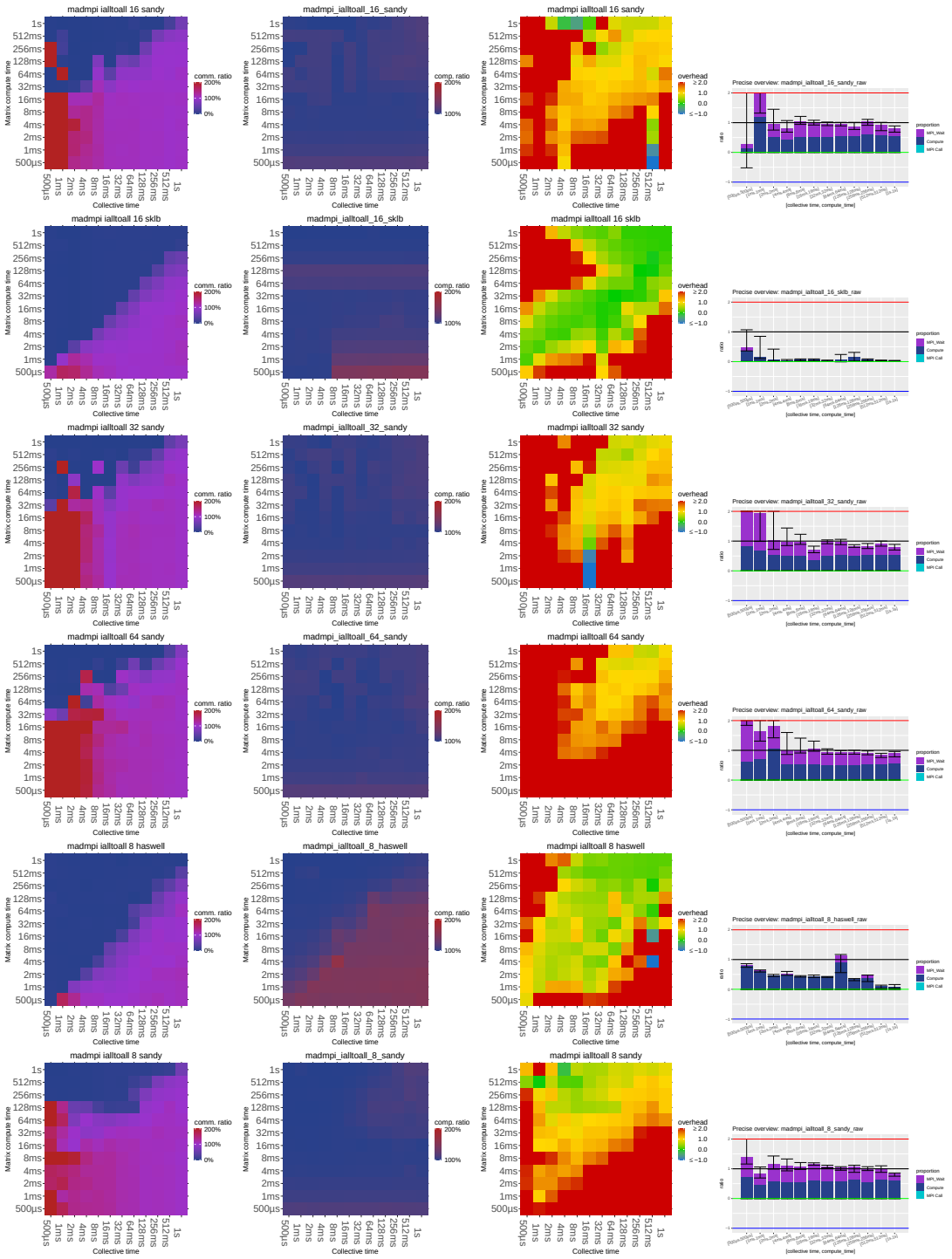

madmpi1DED ibcast 16 sandy
madmpi1DED_ibcast_16_sandy
madmpi1DED ibcast 16 sandy
Precise overview: madmpi1DED_ibcast_16_sandy_raw
madmpi1DED ibcast 16 sklb
madmpi1DED_ibcast_16_sklb
madmpi1DED ibcast 16 sklb
Precise overview: madmpi1DED_ibcast_16_sklb_raw
madmpi1DED ibcast 32 sandy
madmpi1DED_ibcast_32_sandy
madmpi1DED ibcast 32 sandy
Precise overview: madmpi1DED_ibcast_32_sandy_raw
madmpi1DED ibcast 64 sandy
madmpi1DED_ibcast_64_sandy
madmpi1DED ibcast 64 sandy
Precise overview: madmpi1DED_ibcast_64_sandy_raw
madmpi1DED ibcast 8 haswell
madmpi1DED_ibcast_8_haswell
madmpi1DED ibcast 8 haswell
Precise overview: madmpi1DED_ibcast_8_haswell_raw
madmpi1DED ibcast 8 sandy
madmpi1DED_ibcast_8_sandy
madmpi1DED ibcast 8 sandy
Precise overview: madmpi1DED_ibcast_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]

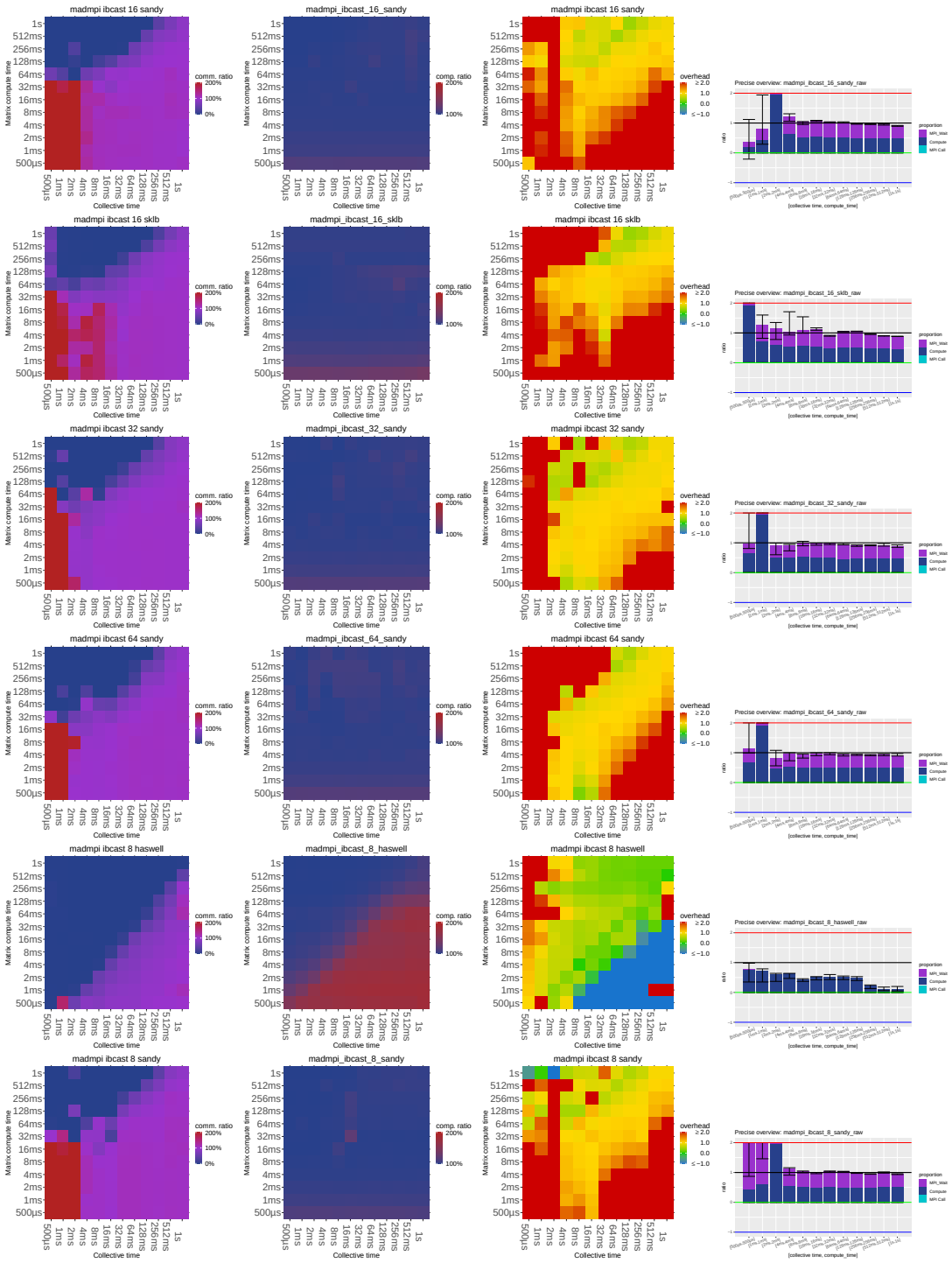

madmpi1DED ireduce 16 sandy
madmpi1DED_ireduce_16_sandy
madmpi1DED ireduce 16 sandy
Precise overview: madmpi1DED_ireduce_16_sandy_raw
madmpi1DED ireduce 16 sklb
madmpi1DED_ireduce_16_sklb
madmpi1DED ireduce 16 sklb
Precise overview: madmpi1DED_ireduce_16_sklb_raw
madmpi1DED ireduce 32 sandy
madmpi1DED_ireduce_32_sandy
madmpi1DED ireduce 32 sandy
Precise overview: madmpi1DED_ireduce_32_sandy_raw
madmpi1DED ireduce 64 sandy
madmpi1DED_ireduce_64_sandy
madmpi1DED ireduce 64 sandy
Precise overview: madmpi1DED_ireduce_64_sandy_raw
madmpi1DED ireduce 8 haswell
madmpi1DED_ireduce_8_haswell
madmpi1DED ireduce 8 haswell
Precise overview: madmpi1DED_ireduce_8_haswell_raw
madmpi1DED ireduce 8 sandy
madmpi1DED_ireduce_8_sandy
madmpi1DED ireduce 8 sandy
Precise overview: madmpi1DED_ireduce_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]

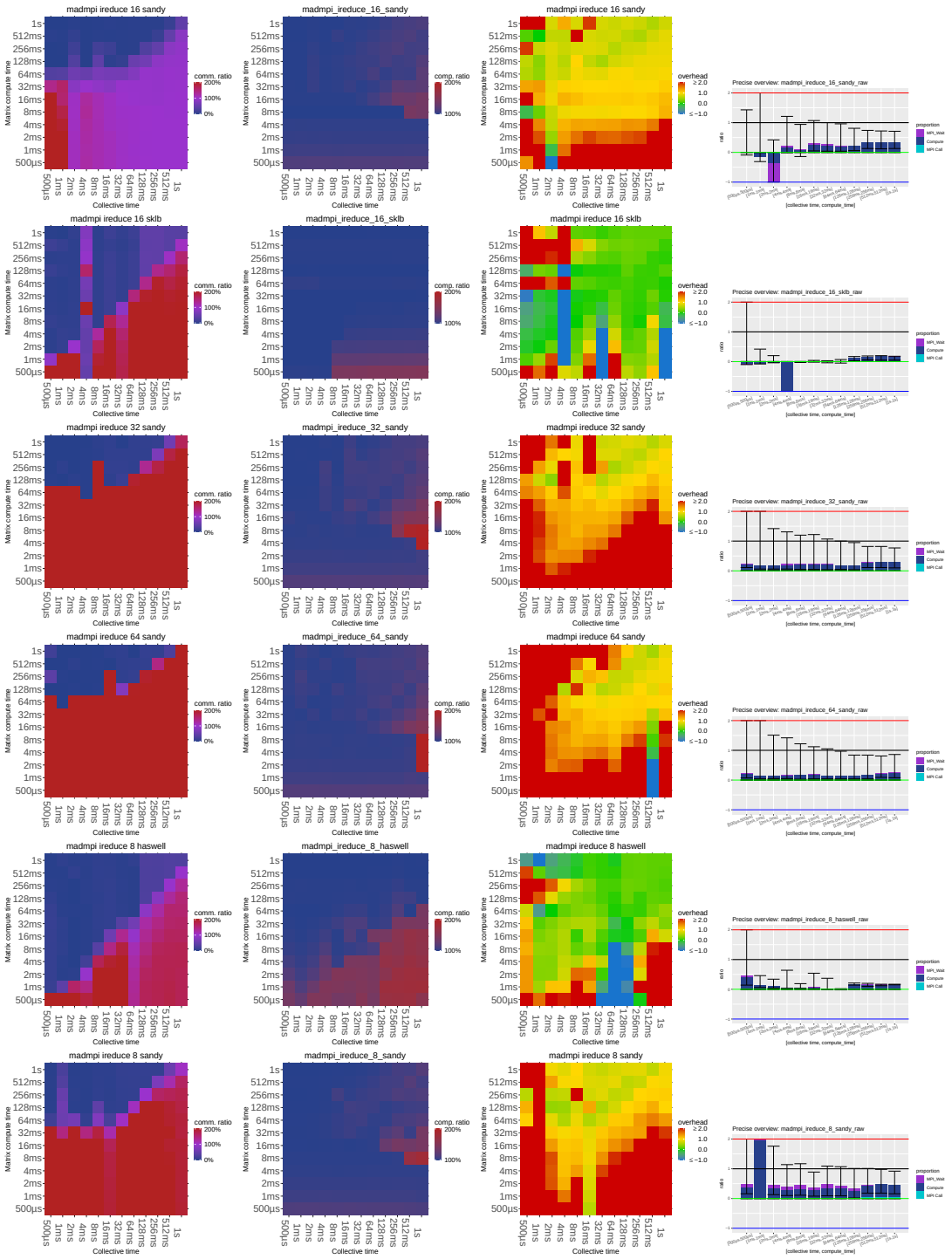
3 MadMPI

The following results are obtained using MadMPI with the default configuration. This is also executed on an Infiniband network.



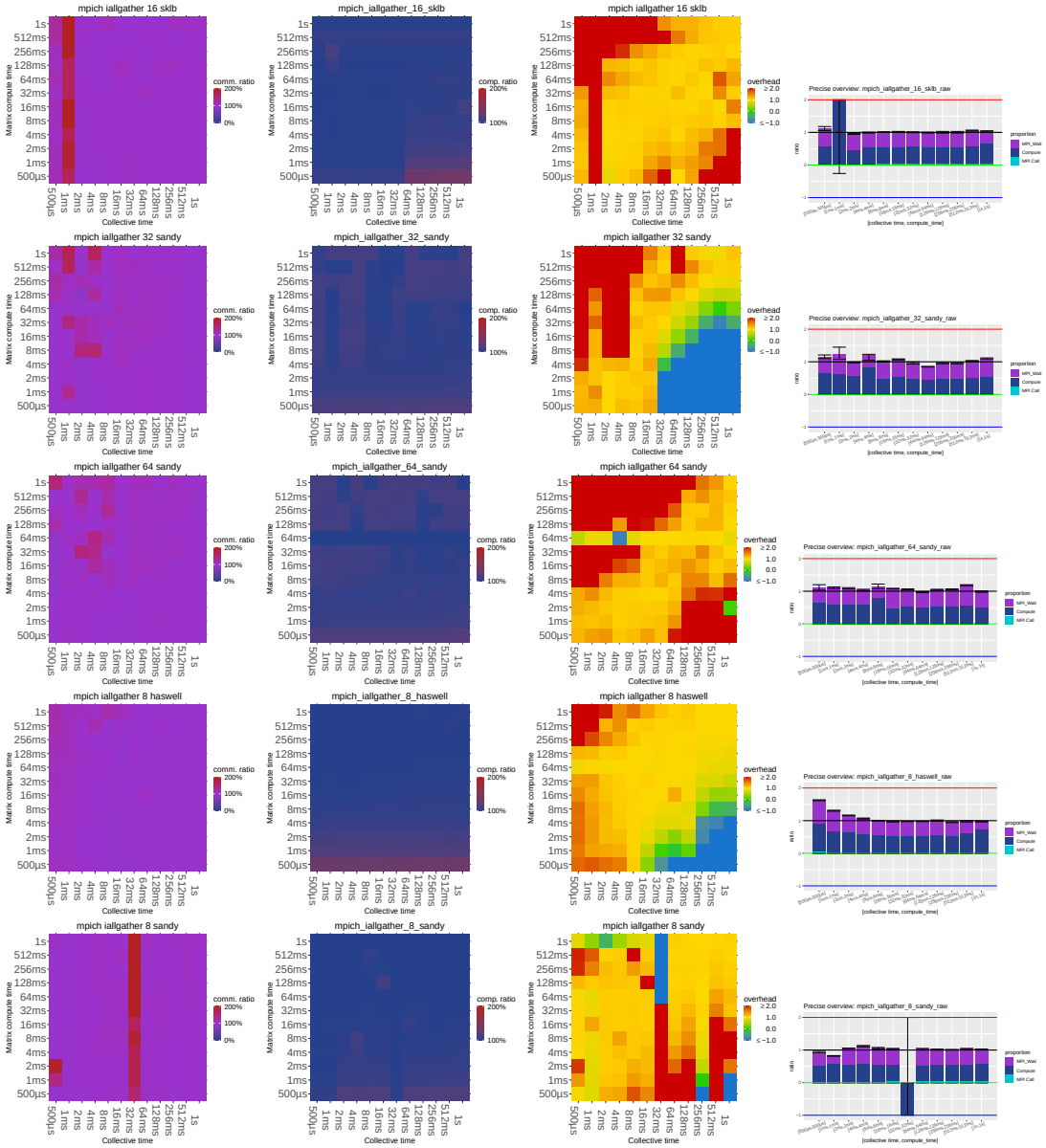

madmpi ialltoall 16 sandy
madmpi_ialltoall_16_sandy
madmpi ialltoall 16 sandy
Precise overview: madmpi_ialltoall_16_sandy_raw
madmpi ialltoall 16 sklb
madmpi_ialltoall_16_sklb
madmpi ialltoall 16 sklb
Precise overview: madmpi_ialltoall_16_sklb_raw
madmpi ialltoall 32 sandy
madmpi_ialltoall_32_sandy
madmpi ialltoall 32 sandy
Precise overview: madmpi_ialltoall_32_sandy_raw
madmpi ialltoall 64 sandy
madmpi_ialltoall_64_sandy
madmpi ialltoall 64 sandy
Precise overview: madmpi_ialltoall_64_sandy_raw
madmpi ialltoall 8 haswell
madmpi_ialltoall_8_haswell
madmpi ialltoall 8 haswell
Precise overview: madmpi_ialltoall_8_haswell_raw
madmpi ialltoall 8 sandy
madmpi_ialltoall_8_sandy
madmpi ialltoall 8 sandy
Precise overview: madmpi_ialltoall_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]

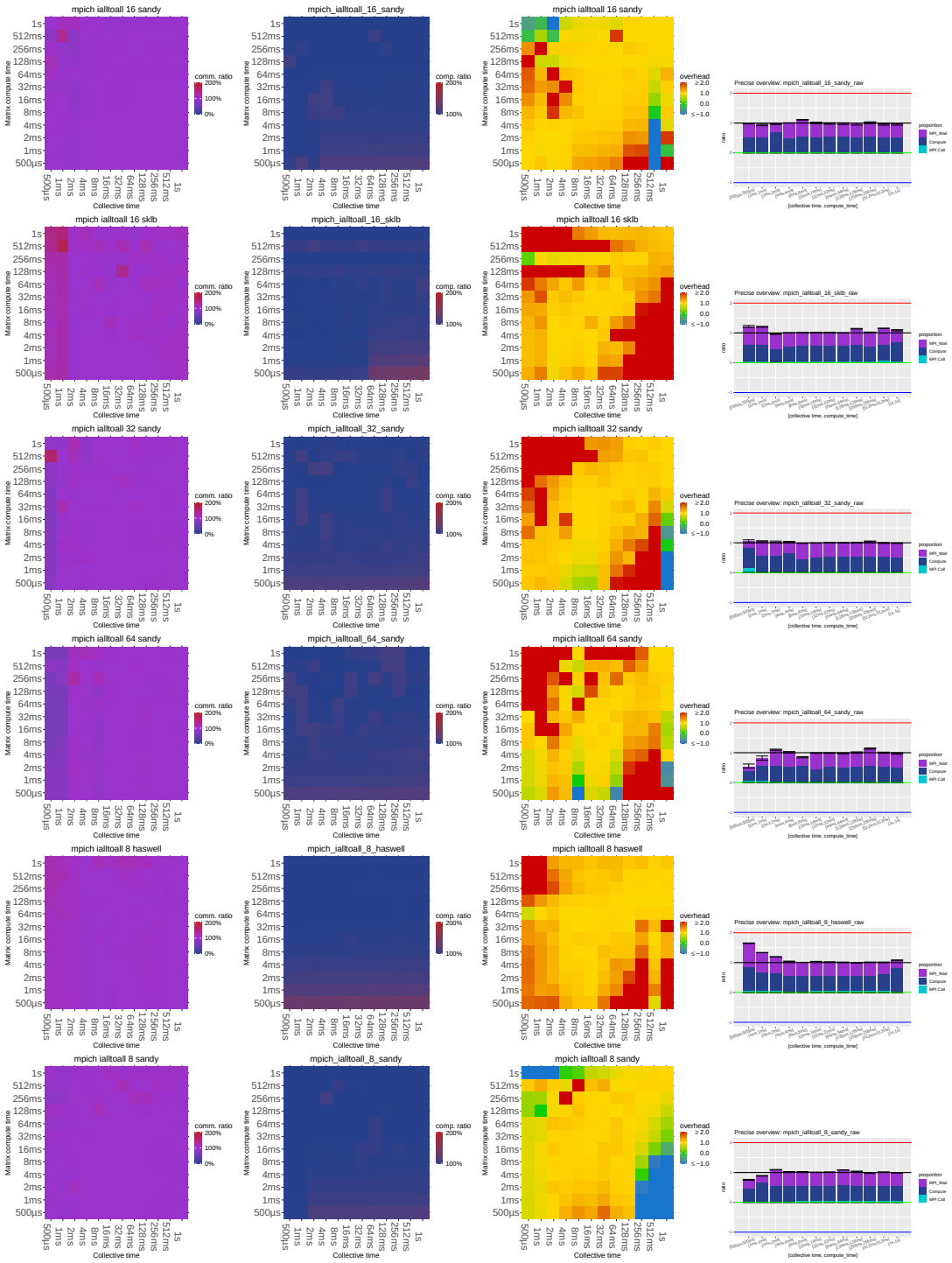

madmpi ibcast 16 sandy
madmpi_ibcast_16_sandy
madmpi ibcast 16 sandy
Precise overview: madmpi_ibcast_16_sandy_raw
madmpi ibcast 16 sklb
madmpi_ibcast_16_sklb
madmpi ibcast 16 sklb
Precise overview: madmpi_ibcast_16_sklb_raw
madmpi ibcast 32 sandy
madmpi_ibcast_32_sandy
madmpi ibcast 32 sandy
Precise overview: madmpi_ibcast_32_sandy_raw
madmpi ibcast 64 sandy
madmpi_ibcast_64_sandy
madmpi ibcast 64 sandy
Precise overview: madmpi_ibcast_64_sandy_raw
madmpi ibcast 8 haswell
madmpi_ibcast_8_haswell
madmpi ibcast 8 haswell
Precise overview: madmpi_ibcast_8_haswell_raw
madmpi ibcast 8 sandy
madmpi_ibcast_8_sandy
madmpi ibcast 8 sandy
Precise overview: madmpi_ibcast_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
ratio
proportion
MPI_Wait
Compute
MPI Call
[collective time, compute_time]

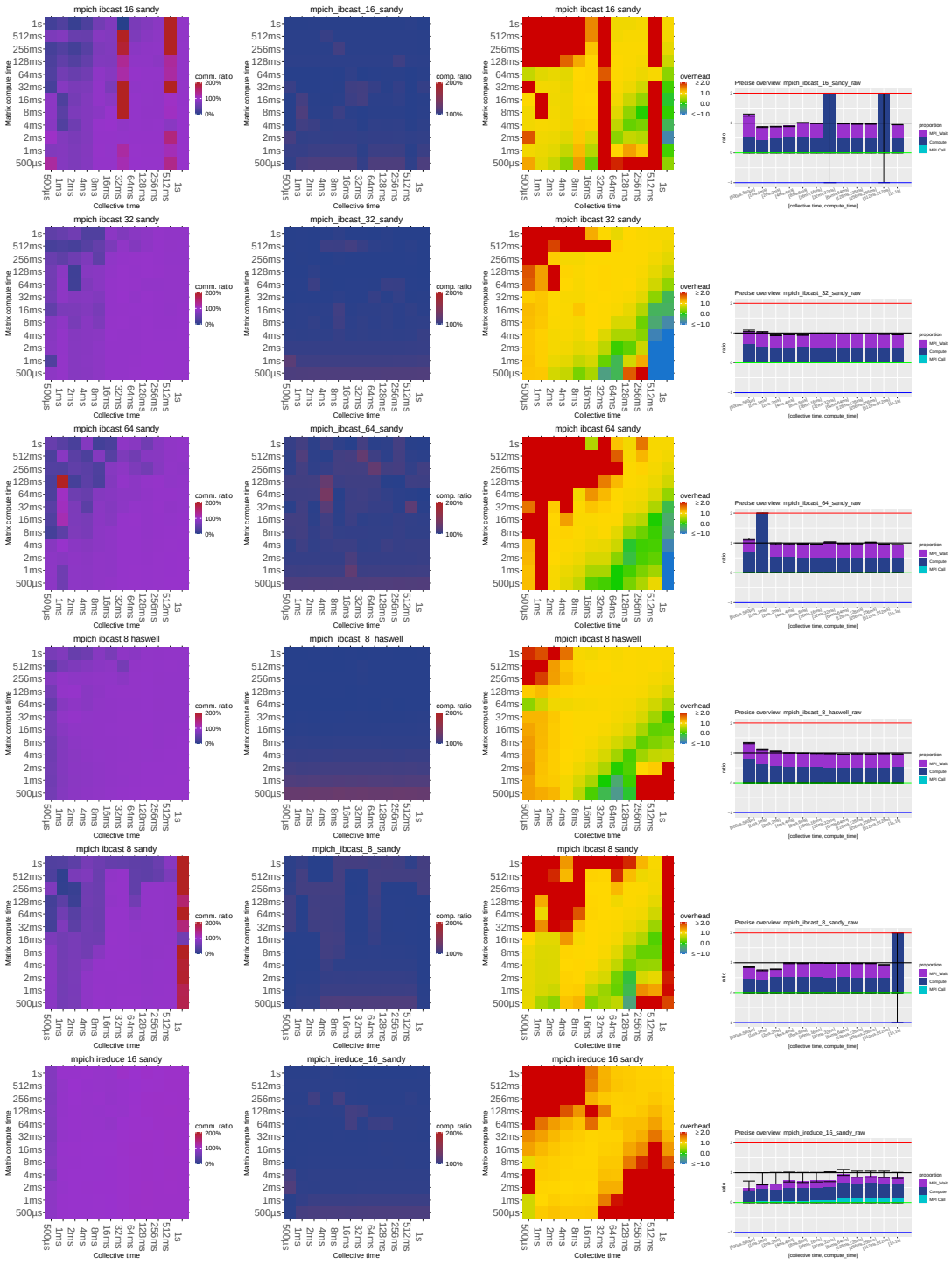

madmpi ireduce 16 sandy
madmpi_ireduce_16_sandy
madmpi ireduce 16 sandy
Precise overview: madmpi_ireduce_16_sandy_raw
madmpi ireduce 16 sklb
madmpi_ireduce_16_sklb
madmpi ireduce 16 sklb
Precise overview: madmpi_ireduce_16_sklb_raw
madmpi ireduce 32 sandy
madmpi_ireduce_32_sandy
madmpi ireduce 32 sandy
Precise overview: madmpi_ireduce_32_sandy_raw
madmpi ireduce 64 sandy
madmpi_ireduce_64_sandy
madmpi ireduce 64 sandy
Precise overview: madmpi_ireduce_64_sandy_raw
madmpi ireduce 8 haswell
madmpi_ireduce_8_haswell
madmpi ireduce 8 haswell
Precise overview: madmpi_ireduce_8_haswell_raw
madmpi ireduce 8 sandy
madmpi_ireduce_8_sandy
madmpi ireduce 8 sandy
Precise overview: madmpi_ireduce_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]

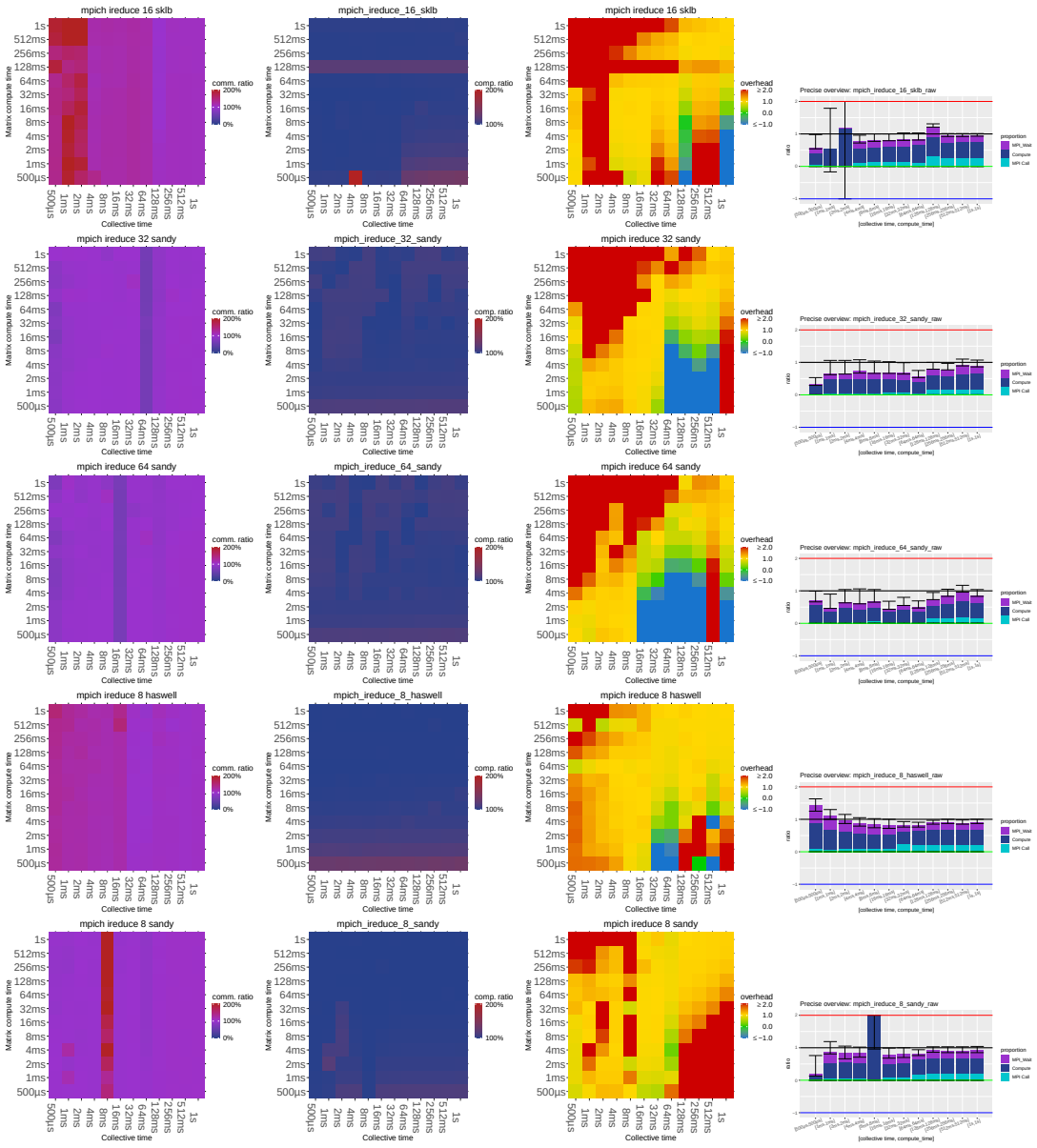
4 MPICH

These experiments are done using MPICH 3.3 with the default configuration. The network used is Infiniband.



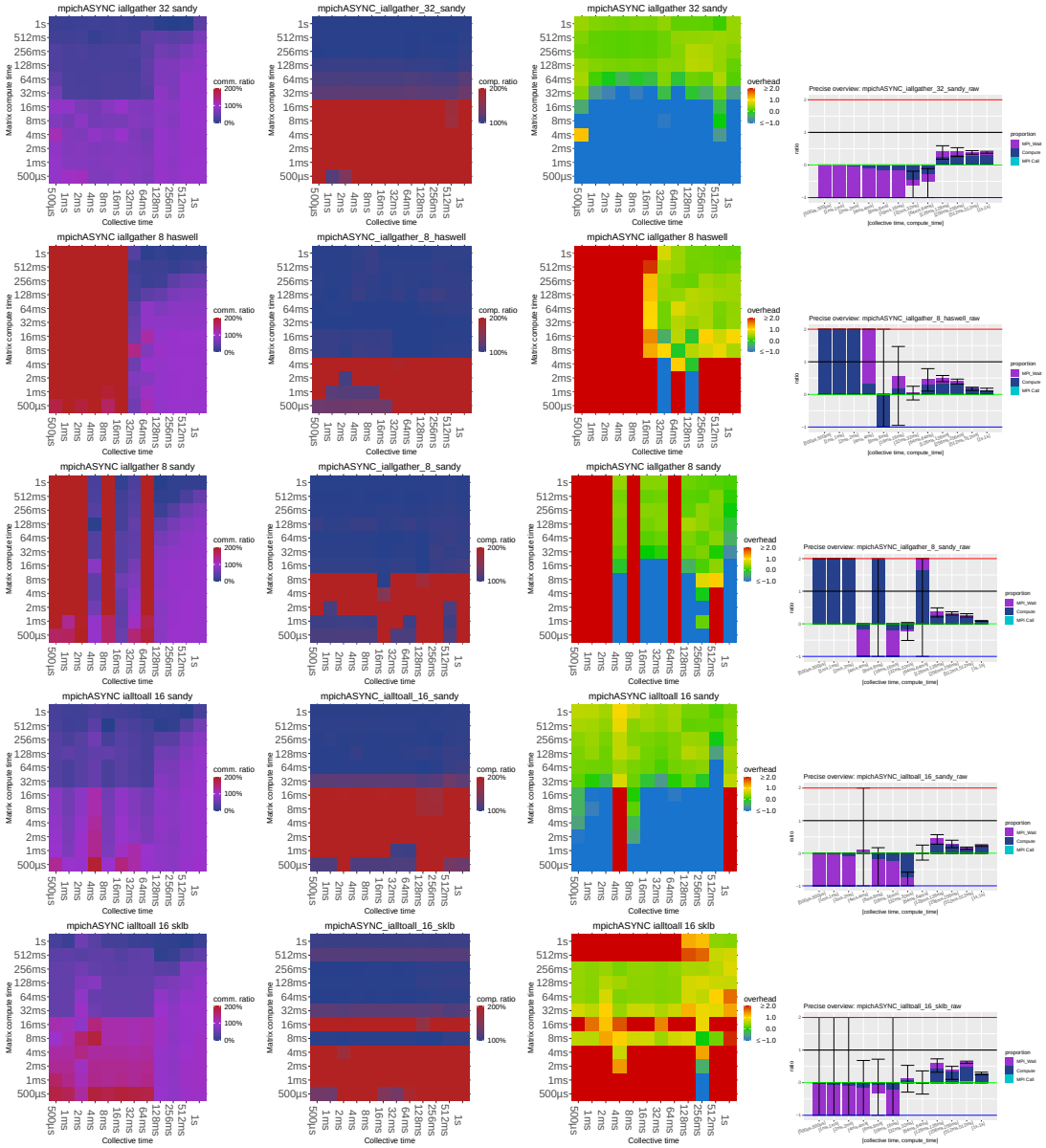

mpich ialltoall 16 sandy
mpich_ialltoall_16_sandy
mpich ialltoall 16 sandy
Precise overview: mpich_ialltoall_16_sandy_raw
mpich ialltoall 16 sklb
mpich_ialltoall_16_sklb
mpich ialltoall 16 sklb
Precise overview: mpich_ialltoall_16_sklb_raw
mpich ialltoall 32 sandy
mpich_ialltoall_32_sandy
mpich ialltoall 32 sandy
Precise overview: mpich_ialltoall_32_sandy_raw
mpich ialltoall 64 sandy
mpich_ialltoall_64_sandy
mpich ialltoall 64 sandy
Precise overview: mpich_ialltoall_64_sandy_raw
mpich ialltoall 8 haswell
mpich_ialltoall_8_haswell
mpich ialltoall 8 haswell
Precise overview: mpich_ialltoall_8_haswell_raw
mpich ialltoall 8 sandy
mpich_ialltoall_8_sandy
mpich ialltoall 8 sandy
Precise overview: mpich_ialltoall_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]

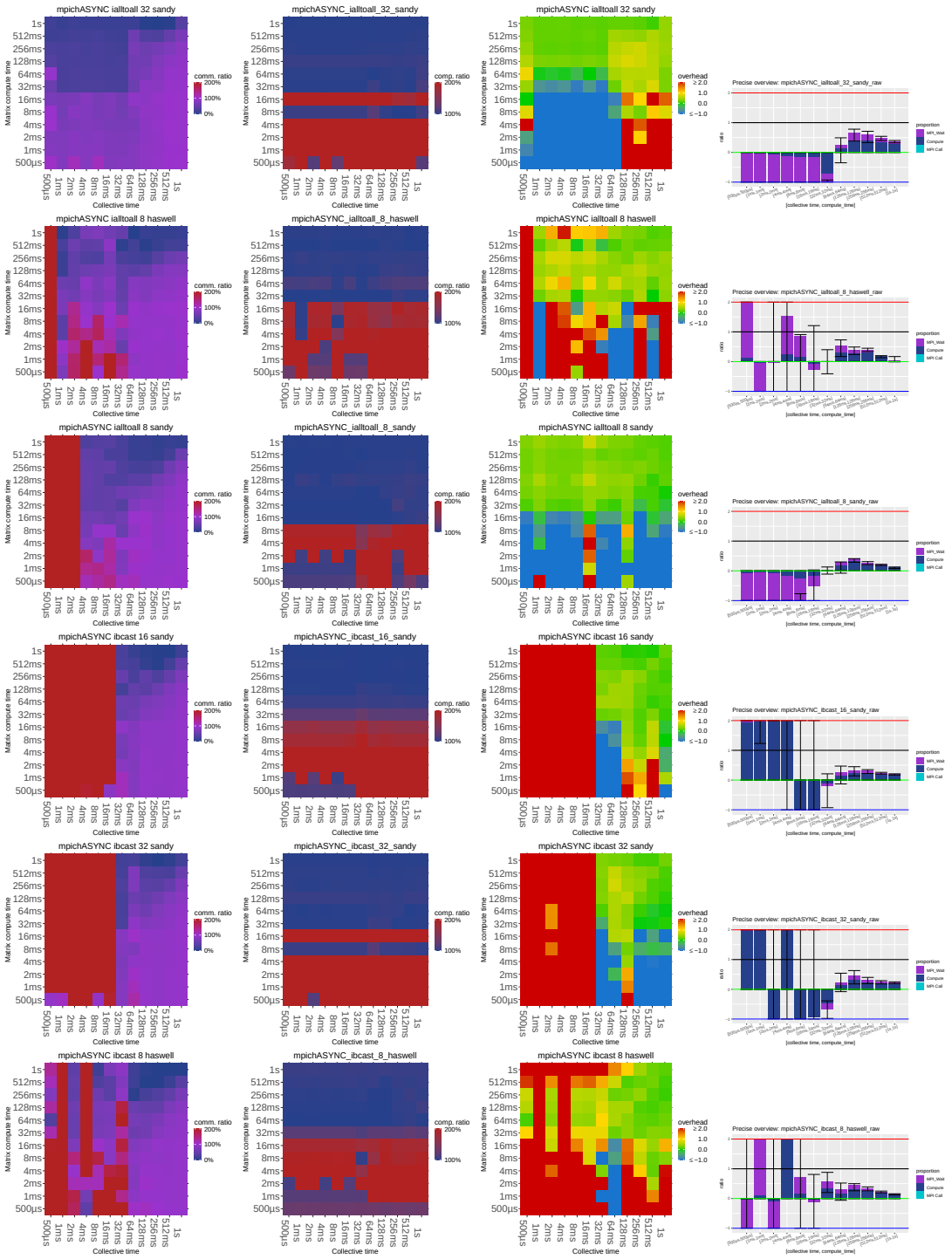

mpich ibcast 16 sandy
mpich_ibcast_16_sandy
mpich ibcast 16 sandy
Precise overview: mpich_ibcast_16_sandy_raw
mpich ibcast 32 sandy
mpich_ibcast_32_sandy
mpich ibcast 32 sandy
Precise overview: mpich_ibcast_32_sandy_raw
mpich ibcast 64 sandy
mpich_ibcast_64_sandy
mpich ibcast 64 sandy
Precise overview: mpich_ibcast_64_sandy_raw
mpich ibcast 8 haswell
mpich_ibcast_8_haswell
mpich ibcast 8 haswell
Precise overview: mpich_ibcast_8_haswell_raw
mpich ibcast 8 sandy
mpich_ibcast_8_sandy
mpich ibcast 8 sandy
Precise overview: mpich_ibcast_8_sandy_raw
mpich ireduce 16 sandy
mpich_ireduce_16_sandy
mpich ireduce 16 sandy
Precise overview: mpich_ireduce_16_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
MPI_Wait
Compute
MPI Call
[collective time, compute_time]

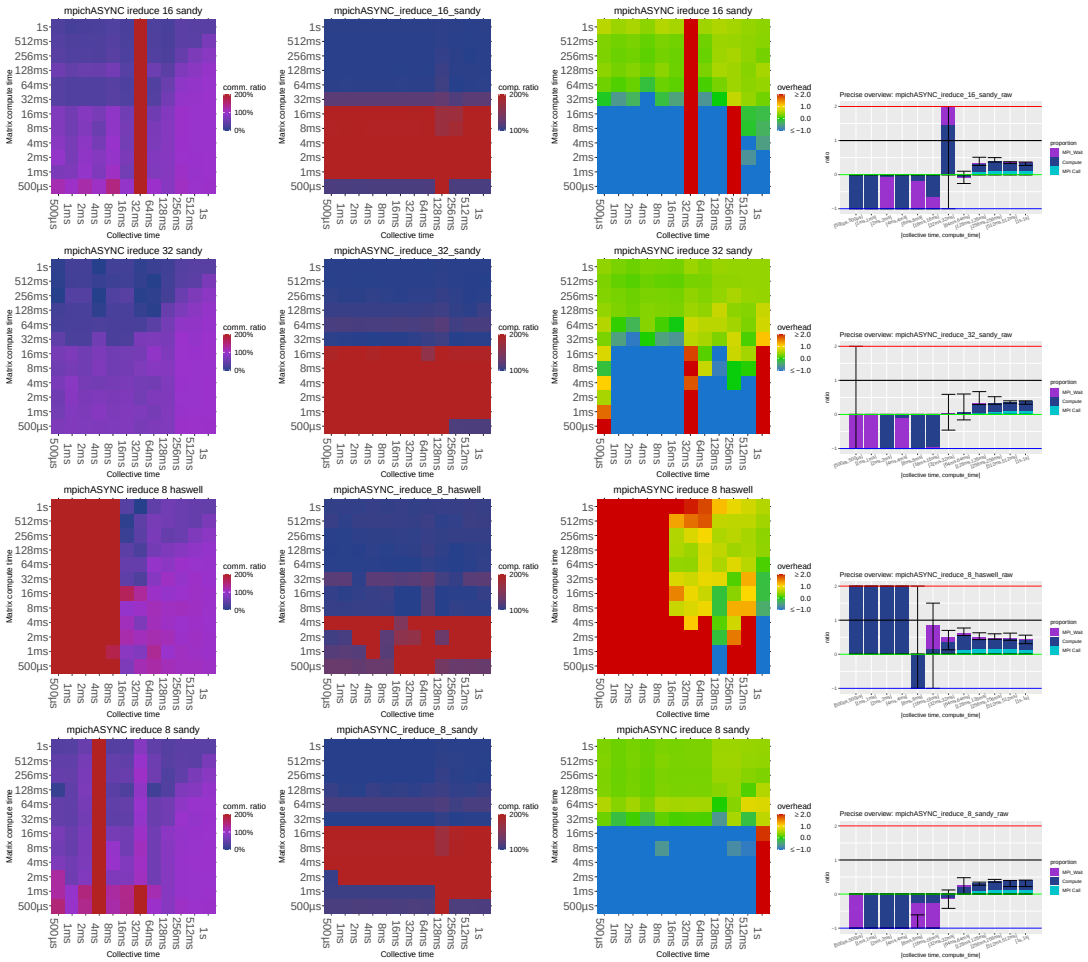

mpich ireduce 16 sklb
mpich_ireduce_16_sklb
mpich ireduce 16 sklb
Precise overview: mpich_ireduce_16_sklb_raw
mpich ireduce 32 sandy
mpich_ireduce_32_sandy
mpich ireduce 32 sandy
Precise overview: mpich_ireduce_32_sandy_raw
mpich ireduce 64 sandy
mpich_ireduce_64_sandy
mpich ireduce 64 sandy
Precise overview: mpich_ireduce_64_sandy_raw
mpich ireduce 8 haswell
mpich_ireduce_8_haswell
mpich ireduce 8 haswell
Precise overview: mpich_ireduce_8_haswell_raw
mpich ireduce 8 sandy
mpich_ireduce_8_sandy
mpich ireduce 8 sandy
Precise overview: mpich_ireduce_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]

5 MPICH with progress thread

The same version (MPICH 3.3) and same network(Infiniband) is used here. We enabled the progress thread using MPICH_ASYNC_PROGRESS=1. No core is dedicated to the progress thread.

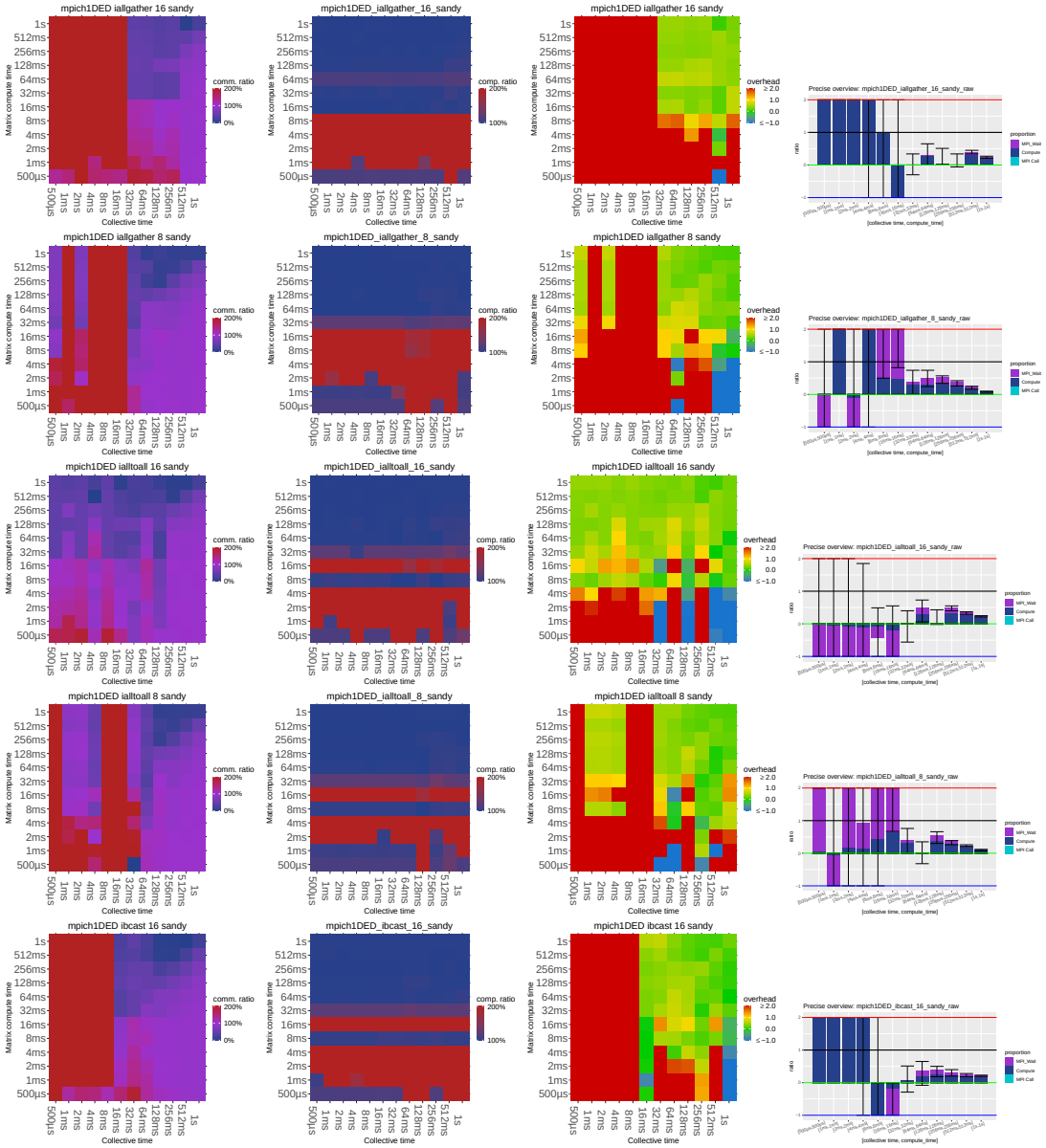


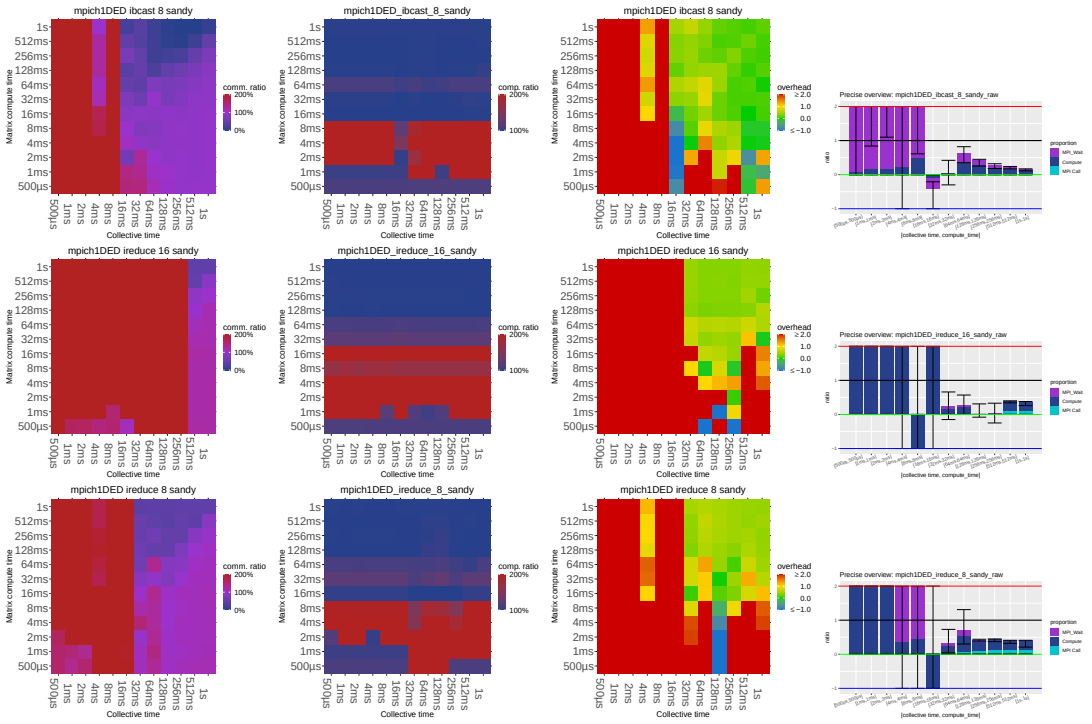

mpichASYNC ialltoall 32 sandy
mpichASYNC_ialltoall_32_sandy
mpichASYNC ialltoall 32 sandy
Precise overview: mpichASYNC_ialltoall_32_sandy_raw
mpichASYNC ialltoall 8 haswell
mpichASYNC_ialltoall_8_haswell
mpichASYNC ialltoall 8 haswell
Precise overview: mpichASYNC_ialltoall_8_haswell_raw
mpichASYNC ialltoall 8 sandy
mpichASYNC_ialltoall_8_sandy
mpichASYNC ialltoall 8 sandy
Precise overview: mpichASYNC_ialltoall_8_sandy_raw
mpichASYNC ibcast 16 sandy
mpichASYNC_ibcast_16_sandy
mpichASYNC ibcast 16 sandy
Precise overview: mpichASYNC_ibcast_16_sandy_raw
mpichASYNC ibcast 32 sandy
mpichASYNC_ibcast_32_sandy
mpichASYNC ibcast 32 sandy
Precise overview: mpichASYNC_ibcast_32_sandy_raw
mpichASYNC ibcast 8 haswell
mpichASYNC_ibcast_8_haswell
mpichASYNC ibcast 8 haswell
Precise overview: mpichASYNC_ibcast_8_haswell_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]


mpichASYNC ireduce 16 sandy
mpichASYNC_ireduce_16_sandy
mpichASYNC ireduce 16 sandy
Precise overview: mpichASYNC_ireduce_16_sandy_raw
mpichASYNC ireduce 32 sandy
mpichASYNC_ireduce_32_sandy
mpichASYNC ireduce 32 sandy
Precise overview: mpichASYNC_ireduce_32_sandy_raw
mpichASYNC ireduce 8 haswell
mpichASYNC_ireduce_8_haswell
mpichASYNC ireduce 8 haswell
Precise overview: mpichASYNC_ireduce_8_haswell_raw
mpichASYNC ireduce 8 sandy
mpichASYNC_ireduce_8_sandy
mpichASYNC ireduce 8 sandy
Precise overview: mpichASYNC_ireduce_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
MPI_Wait
Compute
MPI Call
[collective time, compute_time]

6 MPICH with dedicated core

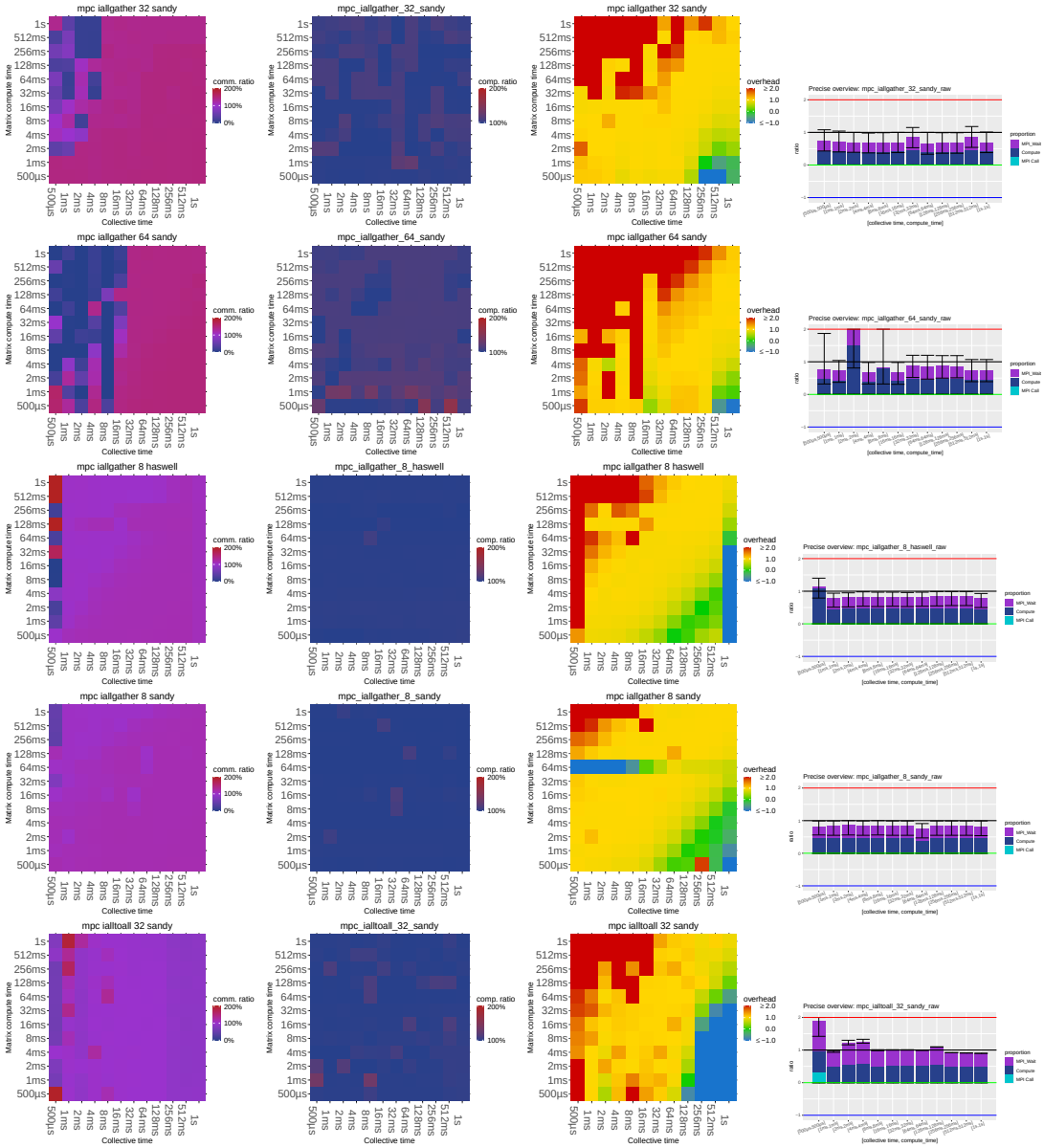
The same version (MPICH 3.3) and same network(Infiniband) is used here. We enabled the progress thread using MPICH_ASYNC_PROGRESS=1. We dedicate a core for the progression. The progress thread placement is done by MPICH.

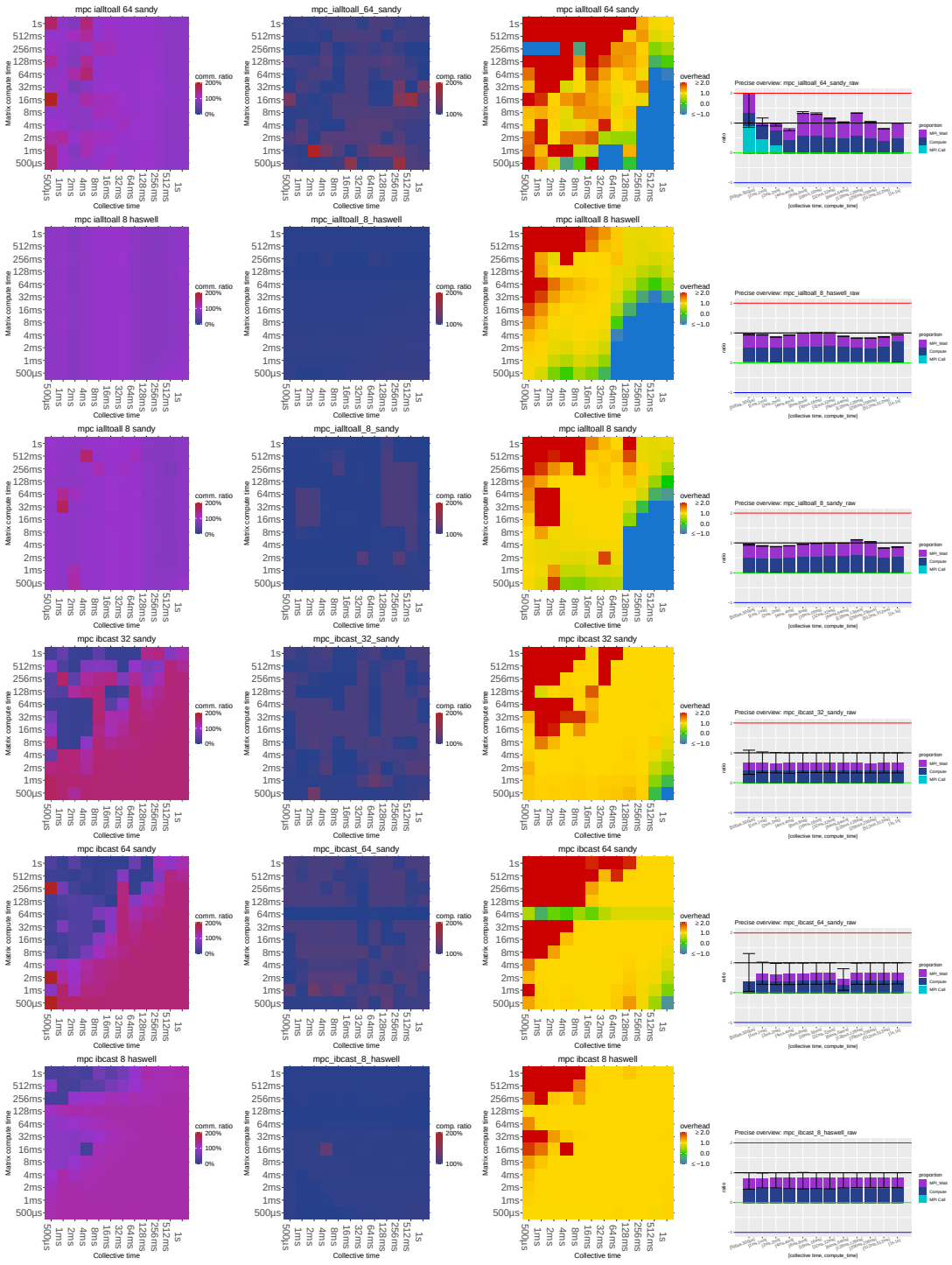


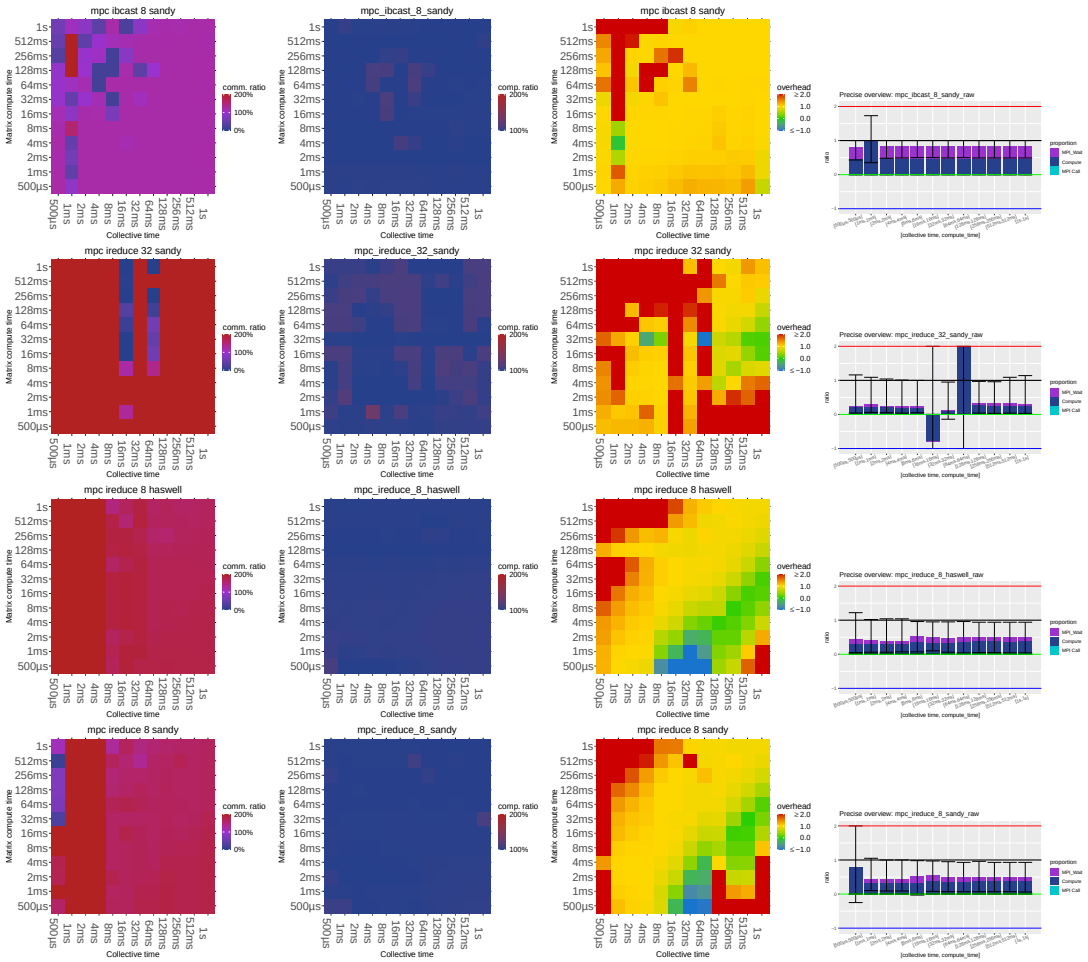

mpich1DED ibcast 8 sandy
mpich1DED_ibcast_8_sandy
mpich1DED ibcast 8 sandy
Precise overview: mpich1DED_ibcast_8_sandy_raw
mpich1DED ireduce 16 sandy
mpich1DED_ireduce_16_sandy
mpich1DED ireduce 16 sandy
Precise overview: mpich1DED_ireduce_16_sandy_raw
mpich1DED ireduce 8 sandy
mpich1DED_ireduce_8_sandy
mpich1DED ireduce 8 sandy
Precise overview: mpich1DED_ireduce_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
MPI_Wait
Compute
MPI Call
[collective time, compute_time]

7 MPC

These results are done using MPC on an Infiniband network. We used the default configuration.

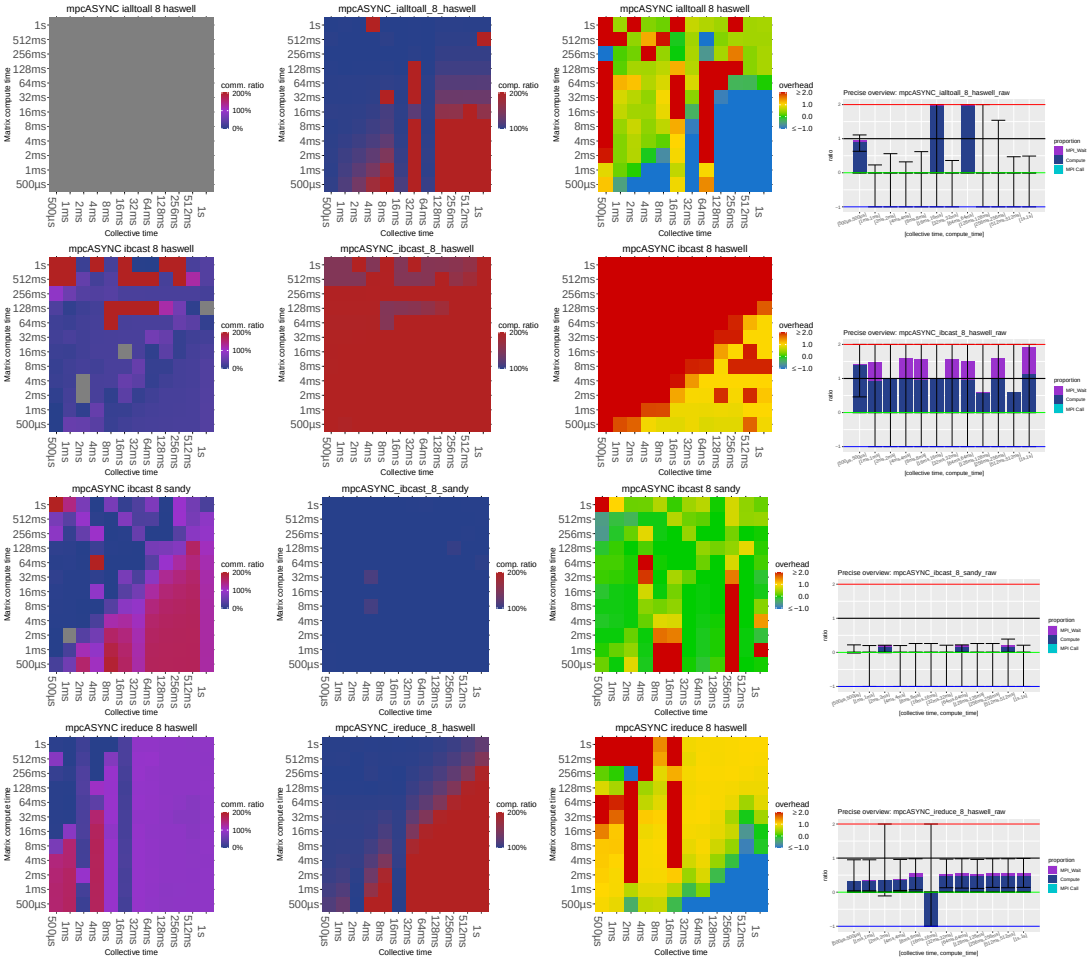



mpc ialltoall 64 sandy
mpc_ialltoall_64_sandy
mpc ialltoall 64 sandy
Precise overview: mpc_ialltoall_64_sandy_raw
mpc ialltoall 8 haswell
mpc_ialltoall_8_haswell
mpc ialltoall 8 haswell
Precise overview: mpc_ialltoall_8_haswell_raw
mpc ialltoall 8 sandy
mpc_ialltoall_8_sandy
mpc ialltoall 8 sandy
Precise overview: mpc_ialltoall_8_sandy_raw
mpc ibcast 32 sandy
mpc_ibcast_32_sandy
mpc ibcast 32 sandy
Precise overview: mpc_ibcast_32_sandy_raw
mpc ibcast 64 sandy
mpc_ibcast_64_sandy
mpc ibcast 64 sandy
Precise overview: mpc_ibcast_64_sandy_raw
mpc ibcast 8 haswell
mpc_ibcast_8_haswell
mpc ibcast 8 haswell
Precise overview: mpc_ibcast_8_haswell_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
ratio
proportion
MPI_Wait
Compute
MPI Call
[collective time, compute_time]


mpc ibcast 8 sandy
mpc_ibcast_8_sandy
mpc ibcast 8 sandy
Precise overview: mpc_ibcast_8_sandy_raw
mpc ireduce 32 sandy
mpc_ireduce_32_sandy
mpc ireduce 32 sandy
Precise overview: mpc_ireduce_32_sandy_raw
mpc ireduce 8 haswell
mpc_ireduce_8_haswell
mpc ireduce 8 haswell
Precise overview: mpc_ireduce_8_haswell_raw
mpc ireduce 8 sandy
mpc_ireduce_8_sandy
mpc ireduce 8 sandy
Precise overview: mpc_ireduce_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]

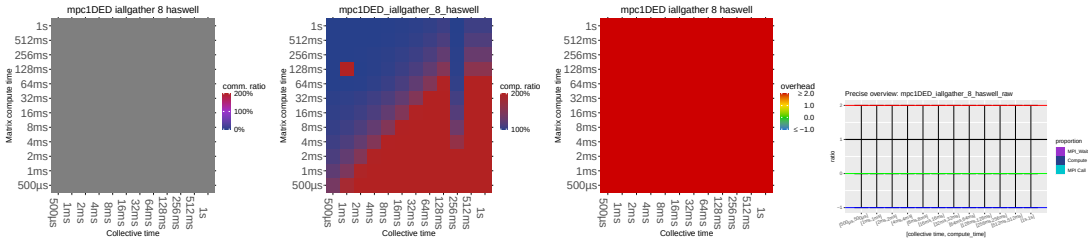
8 MPC with progress thread

These results are done using MPC on an Infiniband network. We enabled the progress thread. We didn't dedicate core for the progress thread.



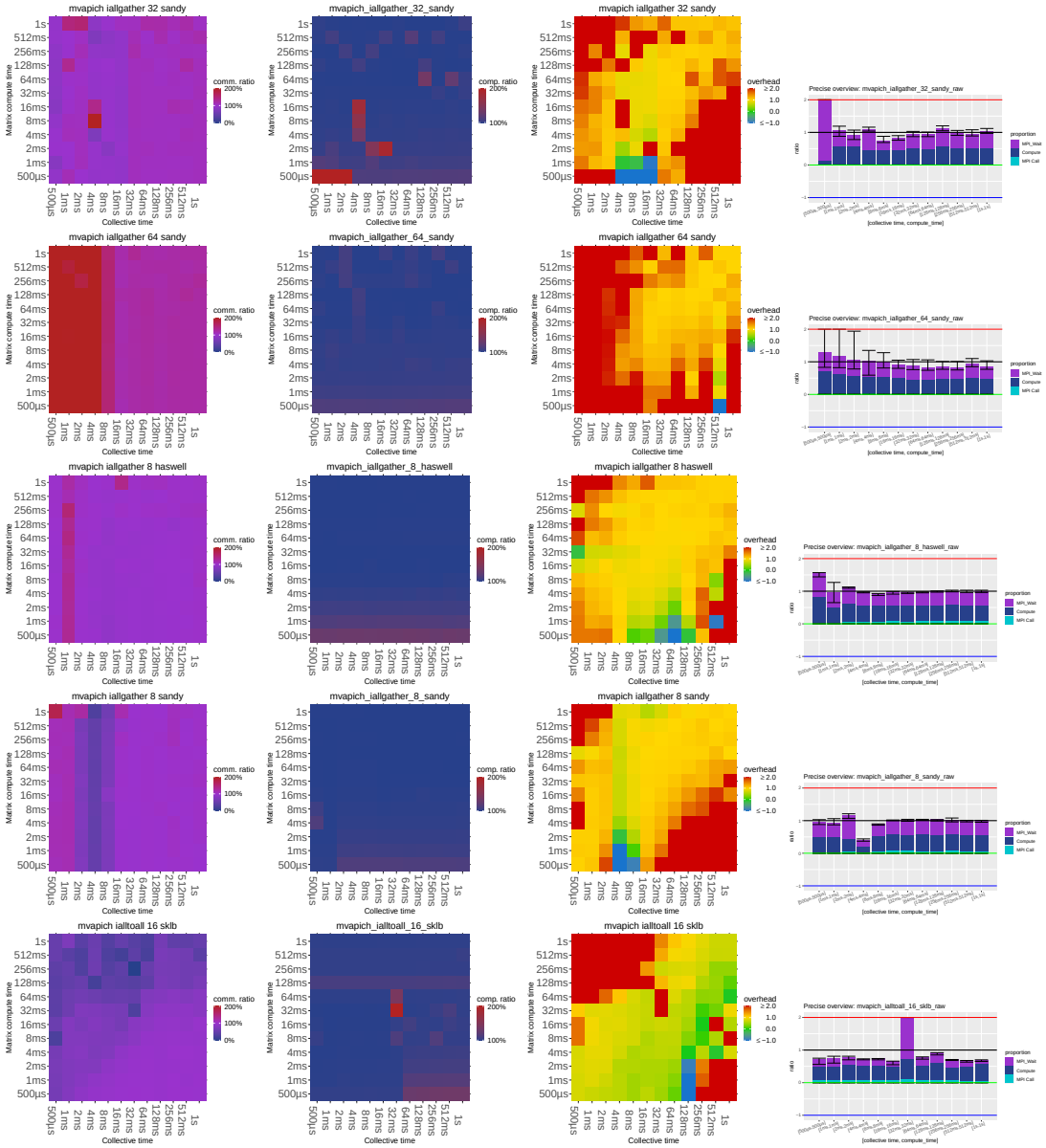
9 MPC with dedicated core

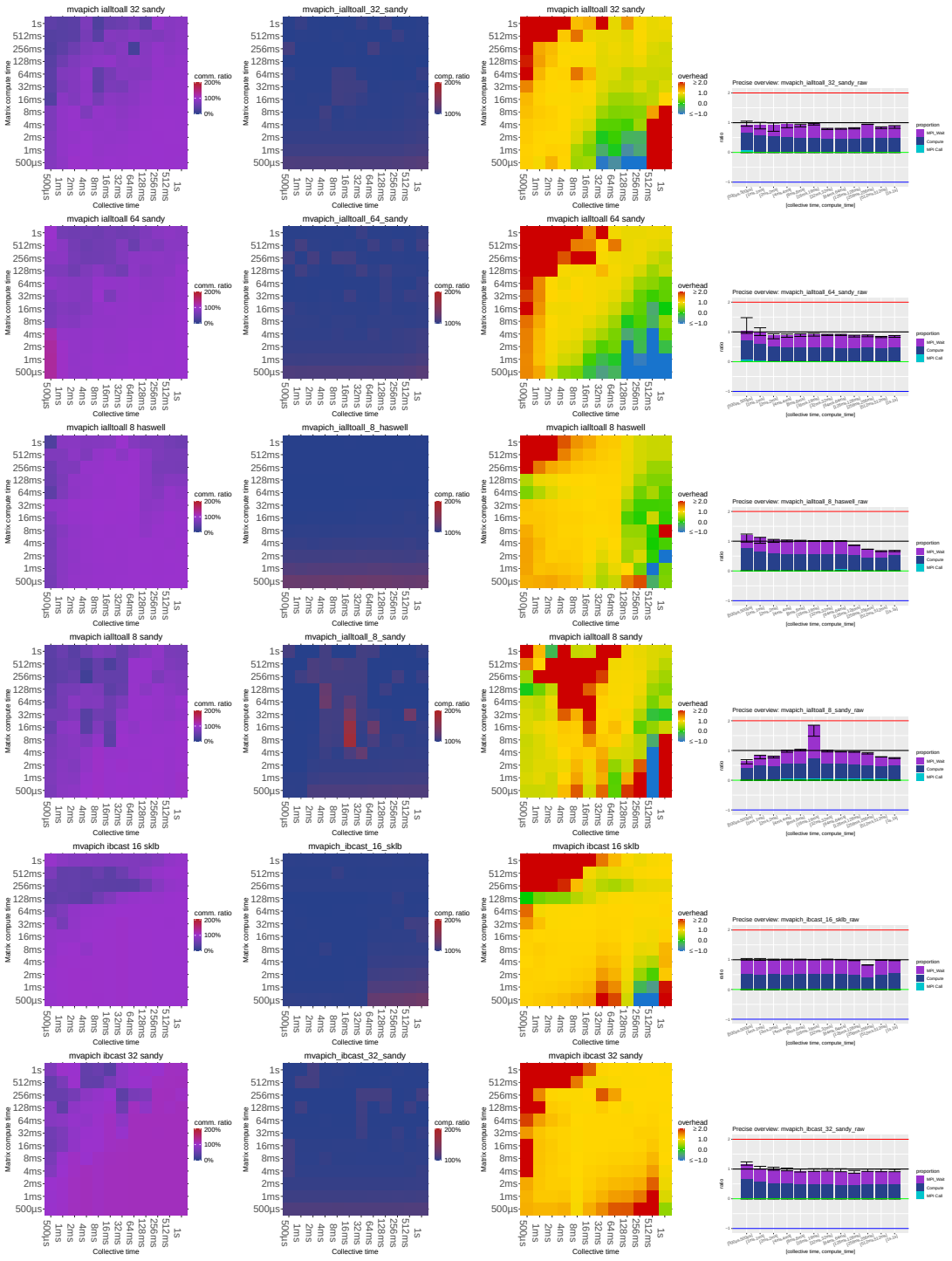
These results are done using MPC on an Infiniband network. We enabled the progress thread. A core is dedicated for the progress thread. The thread is manually bound on this core.

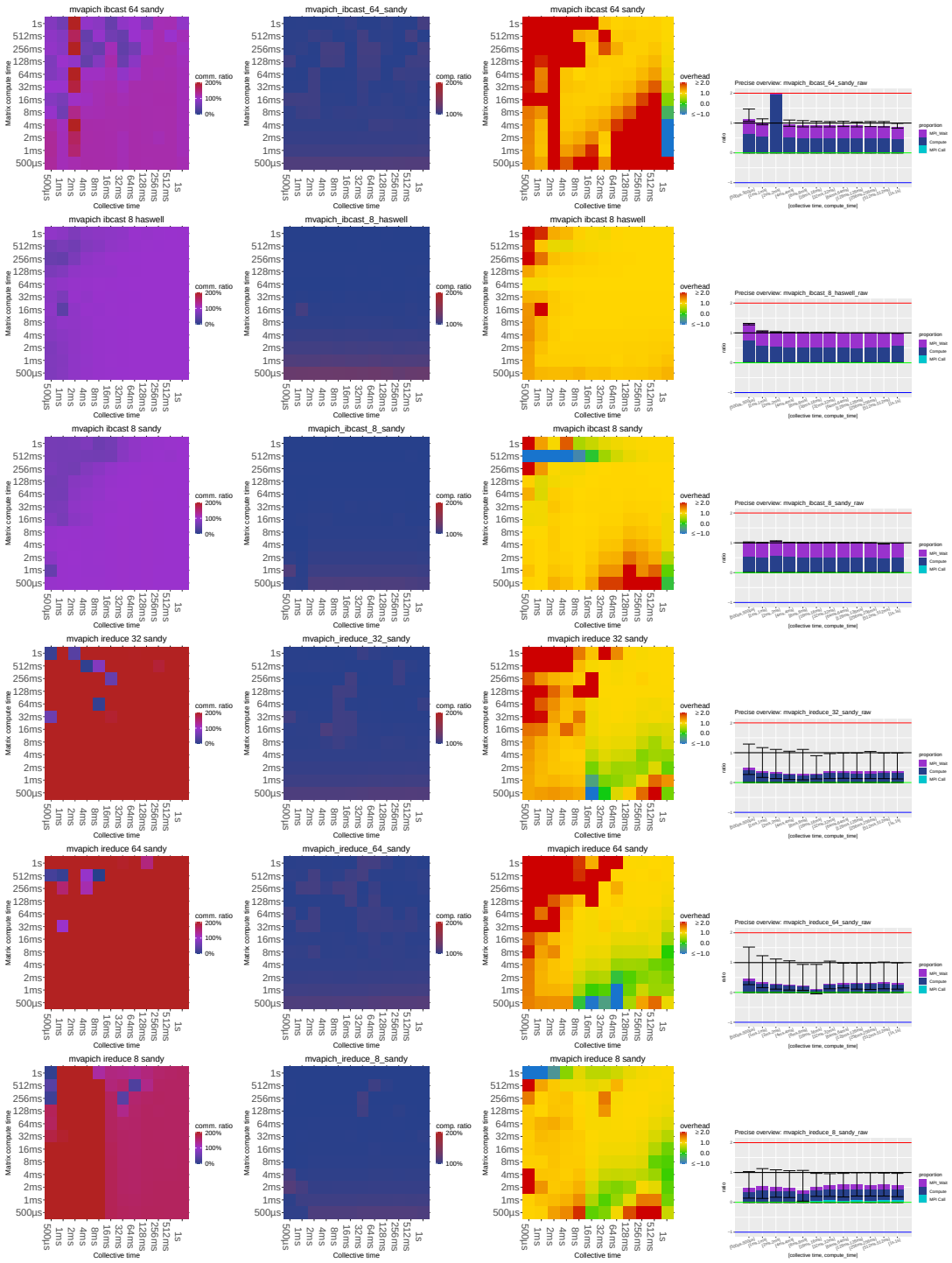


10 MVAPICH

These experiments have been done with MVAPICH 3.2.1 on an Infiniband network. We used the default configuration.

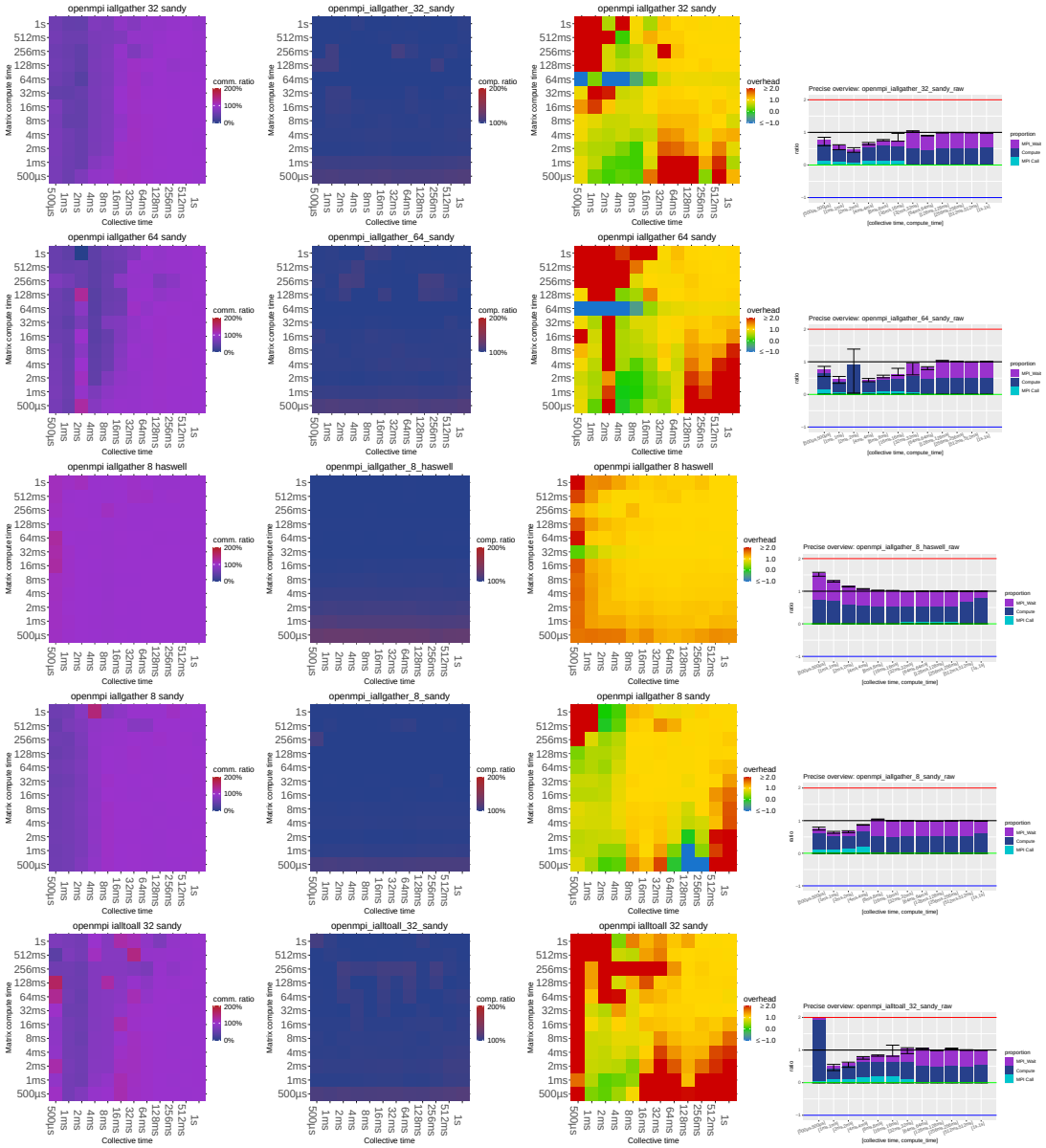


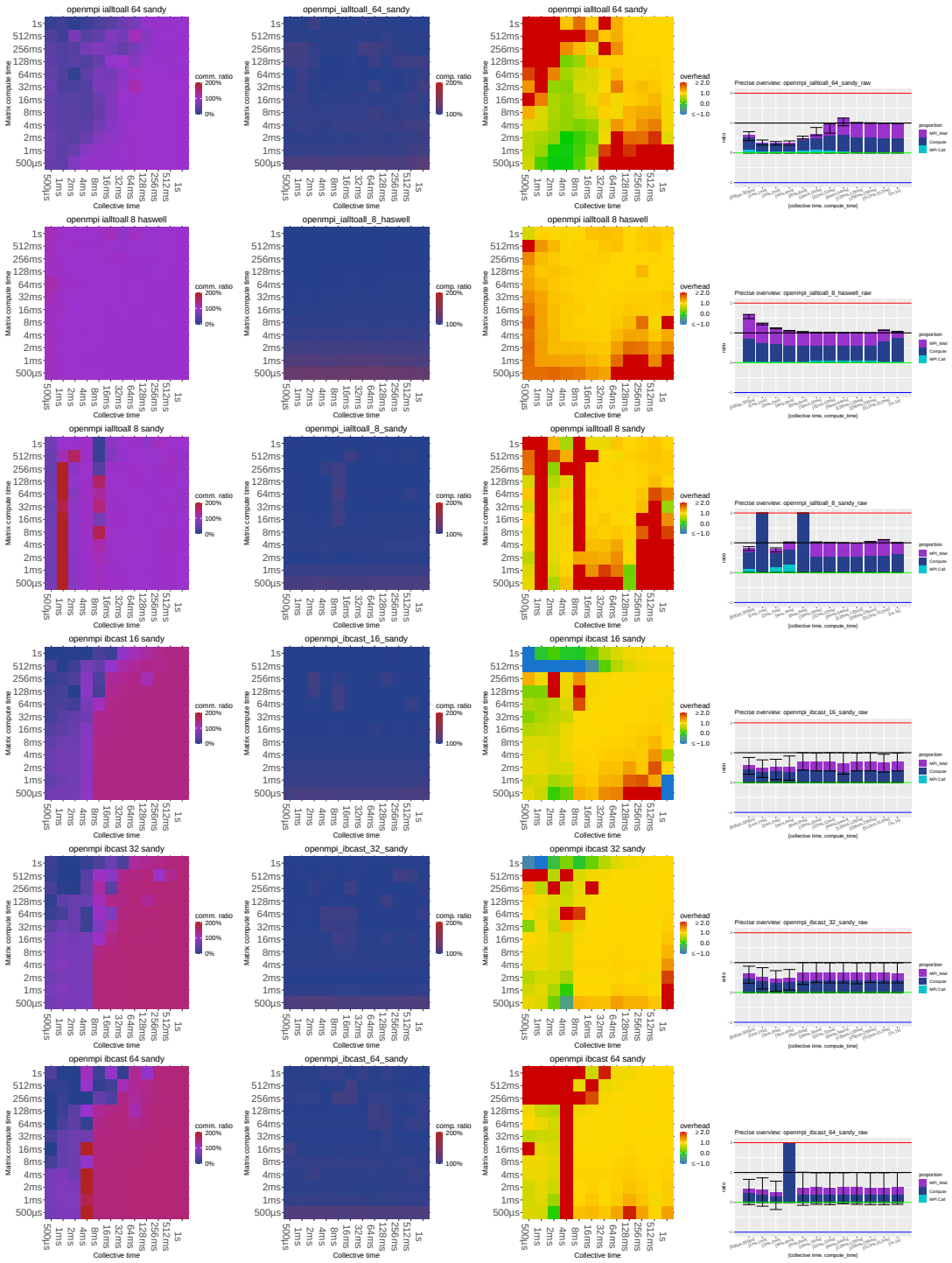

mvapich ialltoall 32 sandy
mvapich_ialltoall_32_sandy
mvapich ialltoall 32 sandy
Precise overview: mvapich_ialltoall_32_sandy_raw
mvapich ialltoall 64 sandy
mvapich_ialltoall_64_sandy
mvapich ialltoall 64 sandy
Precise overview: mvapich_ialltoall_64_sandy_raw
mvapich ialltoall 8 haswell
mvapich_ialltoall_8_haswell
mvapich ialltoall 8 haswell
Precise overview: mvapich_ialltoall_8_haswell_raw
mvapich ialltoall 8 sandy
mvapich_ialltoall_8_sandy
mvapich ialltoall 8 sandy
Precise overview: mvapich_ialltoall_8_sandy_raw
mvapich ibcast 16 sklb
mvapich_ibcast_16_sklb
mvapich ibcast 16 sklb
Precise overview: mvapich_ibcast_16_sklb_raw
mvapich ibcast 32 sandy
mvapich_ibcast_32_sandy
mvapich ibcast 32 sandy
Precise overview: mvapich_ibcast_32_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]

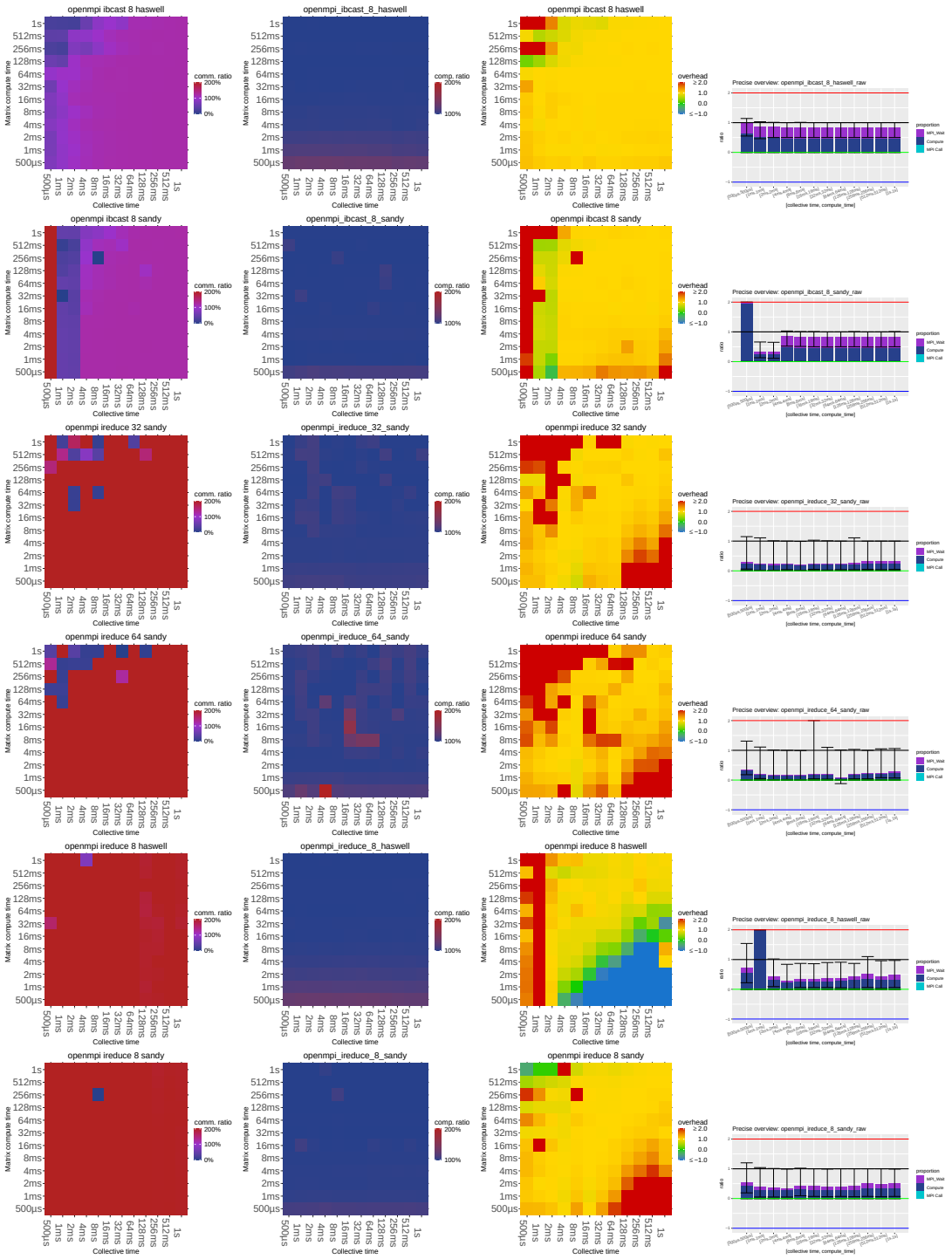

mvapich ibcast 64 sandy
mvapich_ibcast_64_sandy
mvapich ibcast 64 sandy
Precise overview: mvapich_ibcast_64_sandy_raw
mvapich ibcast 8 haswell
mvapich_ibcast_8_haswell
mvapich ibcast 8 haswell
Precise overview: mvapich_ibcast_8_haswell_raw
mvapich ibcast 8 sandy
mvapich_ibcast_8_sandy
mvapich ibcast 8 sandy
Precise overview: mvapich_ibcast_8_sandy_raw
mvapich ireduce 32 sandy
mvapich_ireduce_32_sandy
mvapich ireduce 32 sandy
Precise overview: mvapich_ireduce_32_sandy_raw
mvapich ireduce 64 sandy
mvapich_ireduce_64_sandy
mvapich ireduce 64 sandy
Precise overview: mvapich_ireduce_64_sandy_raw
mvapich ireduce 8 sandy
mvapich_ireduce_8_sandy
mvapich ireduce 8 sandy
Precise overview: mvapich_ireduce_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
ratio
proportion
[collective time, compute_time]

11 Open MPI

The MPI runtime used here is Open MPI 4.0.2. We runned the bnechmark on an Infiniband network. We used the default configuration.

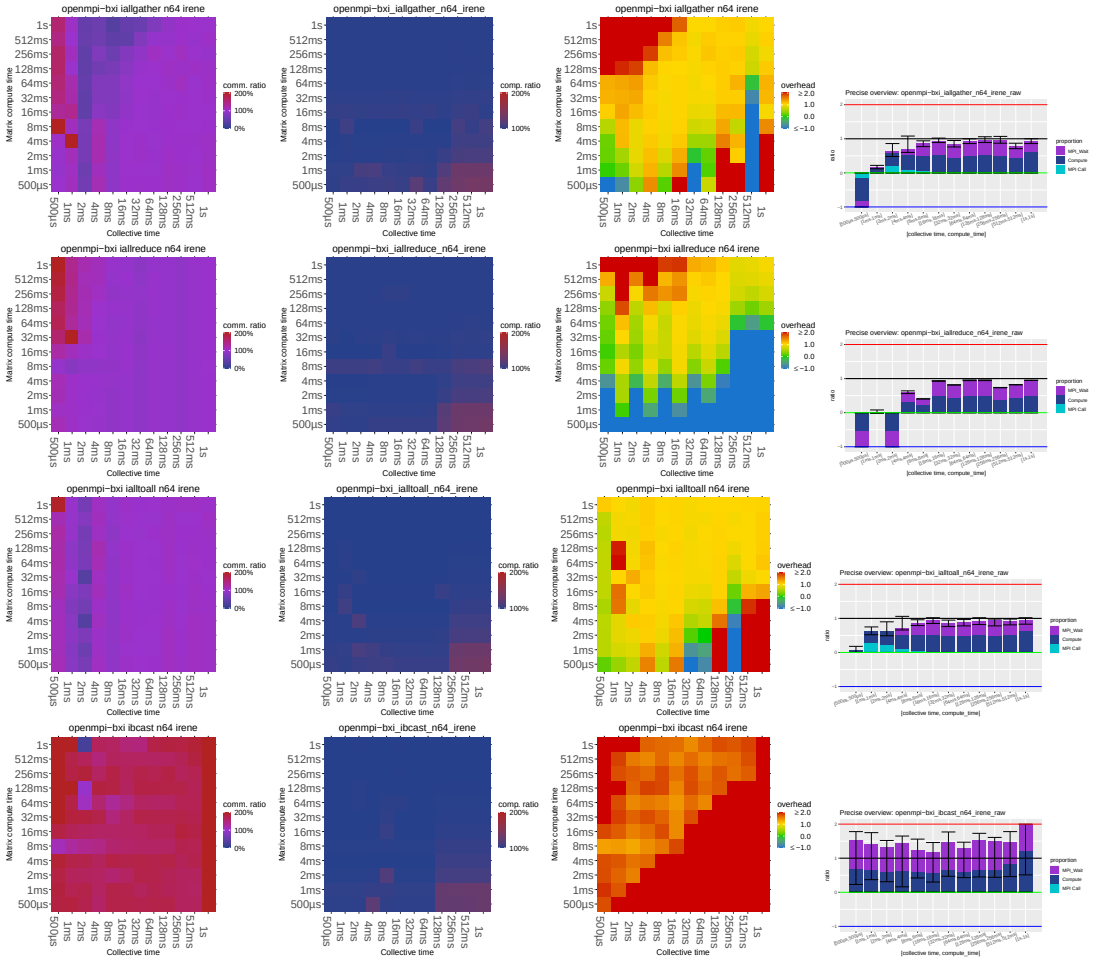



openmpi ialltoall 64 sandy
openmpi_ialltoall_64_sandy
openmpi ialltoall 64 sandy
Precise overview: openmpi_ialltoall_64_sandy_raw
openmpi ialltoall 8 haswell
openmpi_ialltoall_8_haswell
openmpi ialltoall 8 haswell
Precise overview: openmpi_ialltoall_8_haswell_raw
openmpi ialltoall 8 sandy
openmpi_ialltoall_8_sandy
openmpi ialltoall 8 sandy
Precise overview: openmpi_ialltoall_8_sandy_raw
openmpi ibcast 16 sandy
openmpi_ibcast_16_sandy
openmpi ibcast 16 sandy
Precise overview: openmpi_ibcast_16_sandy_raw
openmpi ibcast 32 sandy
openmpi_ibcast_32_sandy
openmpi ibcast 32 sandy
Precise overview: openmpi_ibcast_32_sandy_raw
openmpi ibcast 64 sandy
openmpi_ibcast_64_sandy
openmpi ibcast 64 sandy
Precise overview: openmpi_ibcast_64_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
MPI_Wait
Compute
MPI Call
[collective time, compute_time]


openmpi ibcast 8 haswell
openmpi_ibcast_8_haswell
openmpi ibcast 8 haswell
Precise overview: openmpi_ibcast_8_haswell_raw
openmpi ibcast 8 sandy
openmpi_ibcast_8_sandy
openmpi ibcast 8 sandy
Precise overview: openmpi_ibcast_8_sandy_raw
openmpi ireduce 32 sandy
openmpi_ireduce_32_sandy
openmpi ireduce 32 sandy
Precise overview: openmpi_ireduce_32_sandy_raw
openmpi ireduce 64 sandy
openmpi_ireduce_64_sandy
openmpi ireduce 64 sandy
Precise overview: openmpi_ireduce_64_sandy_raw
openmpi ireduce 8 haswell
openmpi_ireduce_8_haswell
openmpi ireduce 8 haswell
Precise overview: openmpi_ireduce_8_haswell_raw
openmpi ireduce 8 sandy
openmpi_ireduce_8_sandy
openmpi ireduce 8 sandy
Precise overview: openmpi_ireduce_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]

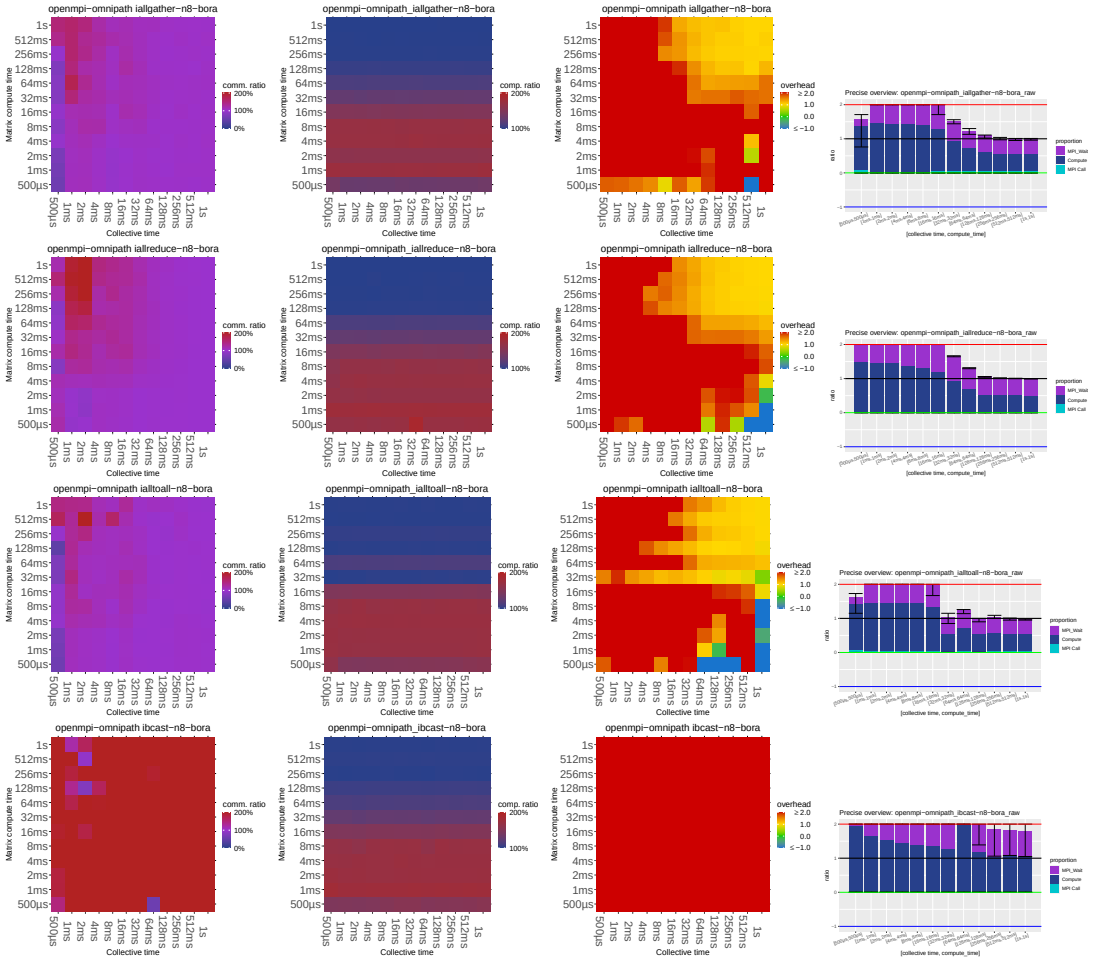
12 Open MPI with BXI adapter

We used the same configuration as previous section. The network adapter is a Bull BXI v1.2.



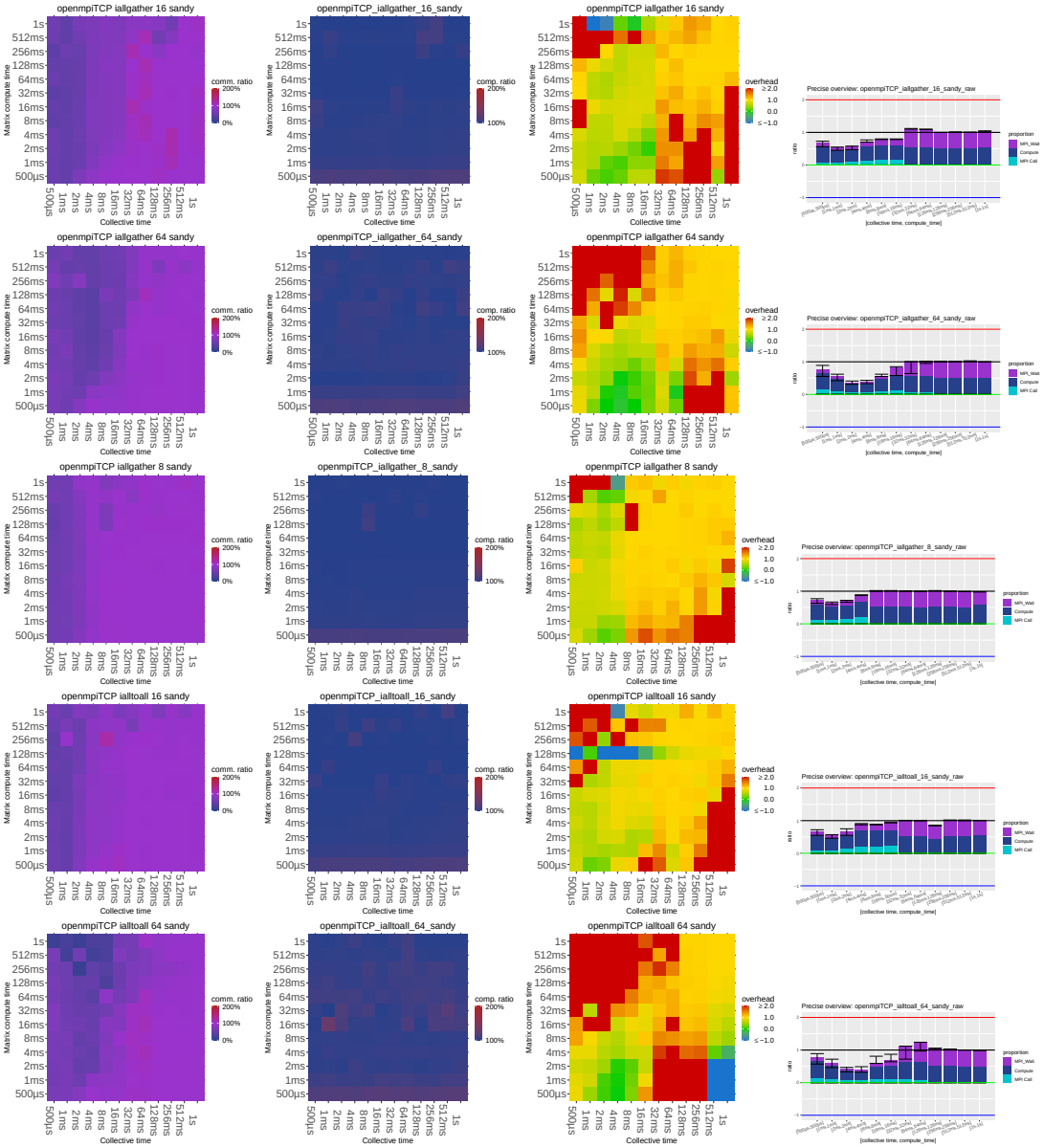
13 Open MPI on Omnipath network

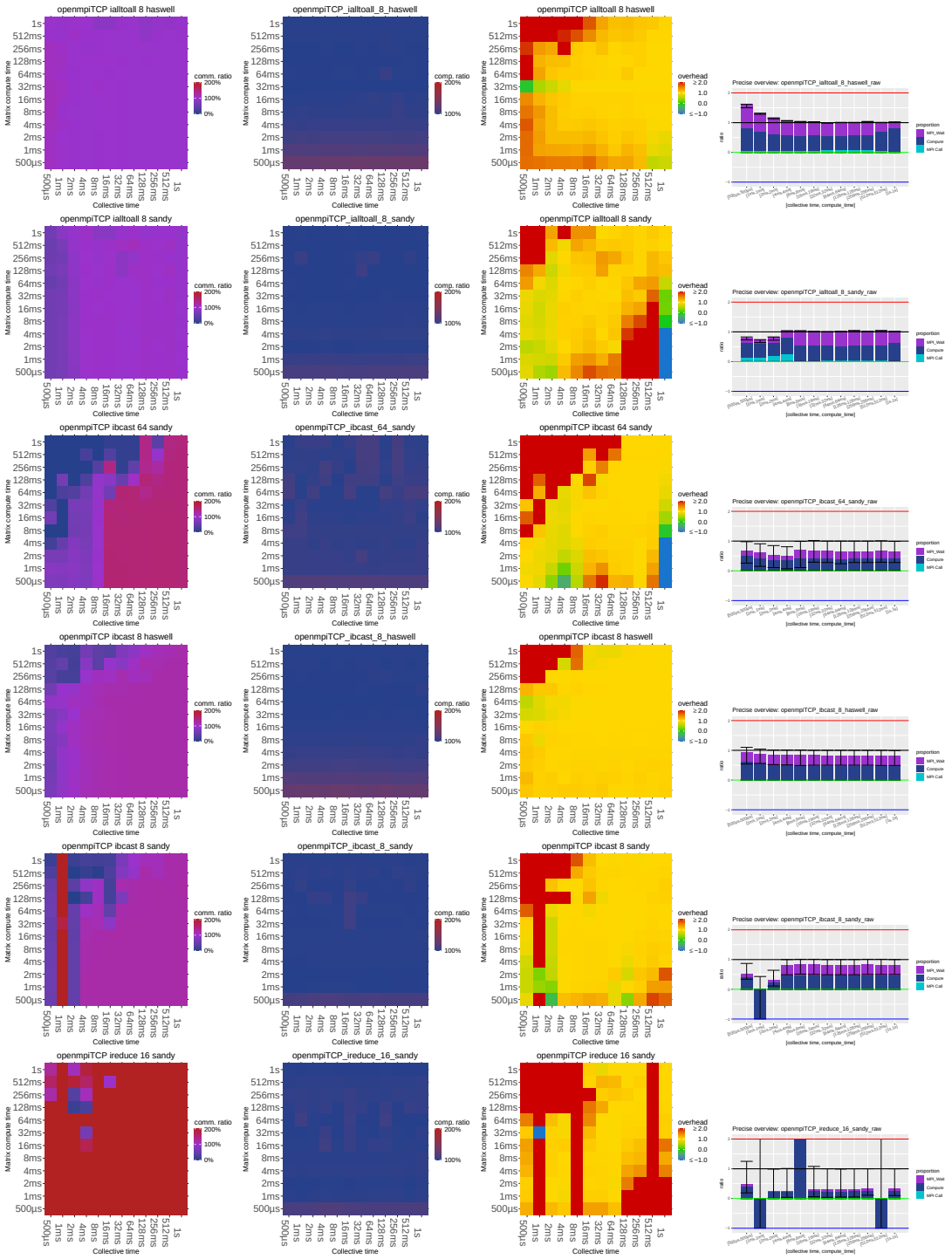
We used the same configuration as both previous sections but on an Omnipath network.

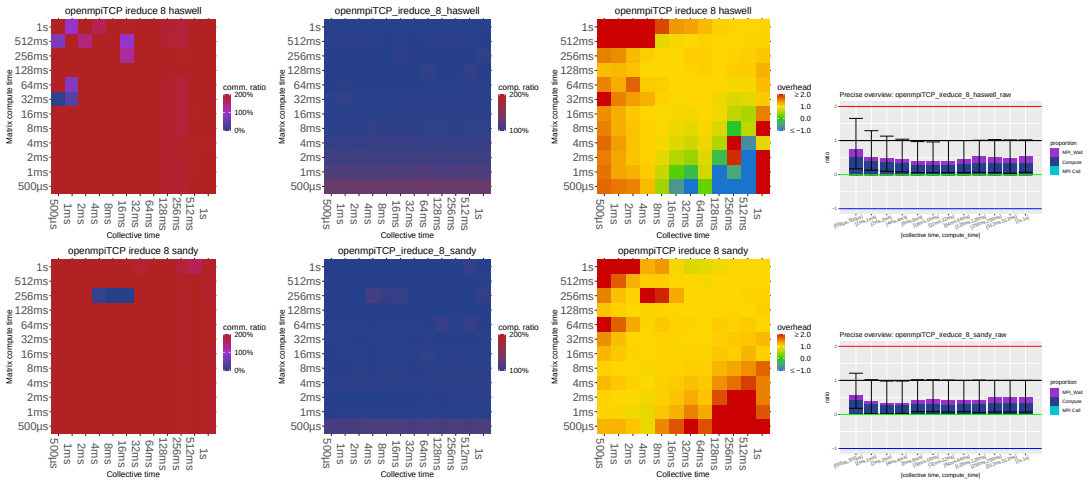


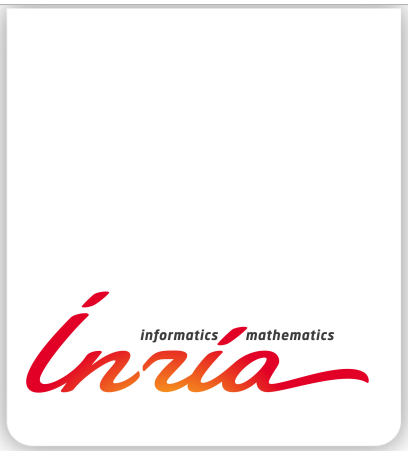
14 Open MPI on TCP network

We used the same configuration as three previous sections but on an TCP network.




openmpiTCP ialltoall 8 haswell
openmpiTCP_ialltoall_8_haswell
openmpiTCP ialltoall 8 haswell
Precise overview: openmpiTCP_ialltoall_8_haswell_raw
openmpiTCP ialltoall 8 sandy
openmpiTCP_ialltoall_8_sandy
openmpiTCP ialltoall 8 sandy
Precise overview: openmpiTCP_ialltoall_8_sandy_raw
openmpiTCP ibcast 64 sandy
openmpiTCP_ibcast_64_sandy
openmpiTCP ibcast 64 sandy
Precise overview: openmpiTCP_ibcast_64_sandy_raw
openmpiTCP ibcast 8 haswell
openmpiTCP_ibcast_8_haswell
openmpiTCP ibcast 8 haswell
Precise overview: openmpiTCP_ibcast_8_haswell_raw
openmpiTCP ibcast 8 sandy
openmpiTCP_ibcast_8_sandy
openmpiTCP ibcast 8 sandy
Precise overview: openmpiTCP_ibcast_8_sandy_raw
openmpiTCP ireduce 16 sandy
openmpiTCP_ireduce_16_sandy
openmpiTCP ireduce 16 sandy
Precise overview: openmpiTCP_ireduce_16_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
[collective time, compute_time]


openmpiTCP ireduce 8 haswell
openmpiTCP_ireduce_8_haswell
openmpiTCP ireduce 8 haswell
Precise overview: openmpiTCP_ireduce_8_haswell_raw
openmpiTCP ireduce 8 sandy
openmpiTCP_ireduce_8_sandy
openmpiTCP ireduce 8 sandy
Precise overview: openmpiTCP_ireduce_8_sandy_raw
Matrix compute time
Collective time
comm. ratio
comp. ratio
overhead
proportion
MPI_Wait
Compute
MPI Call
[collective time, compute_time]



RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399