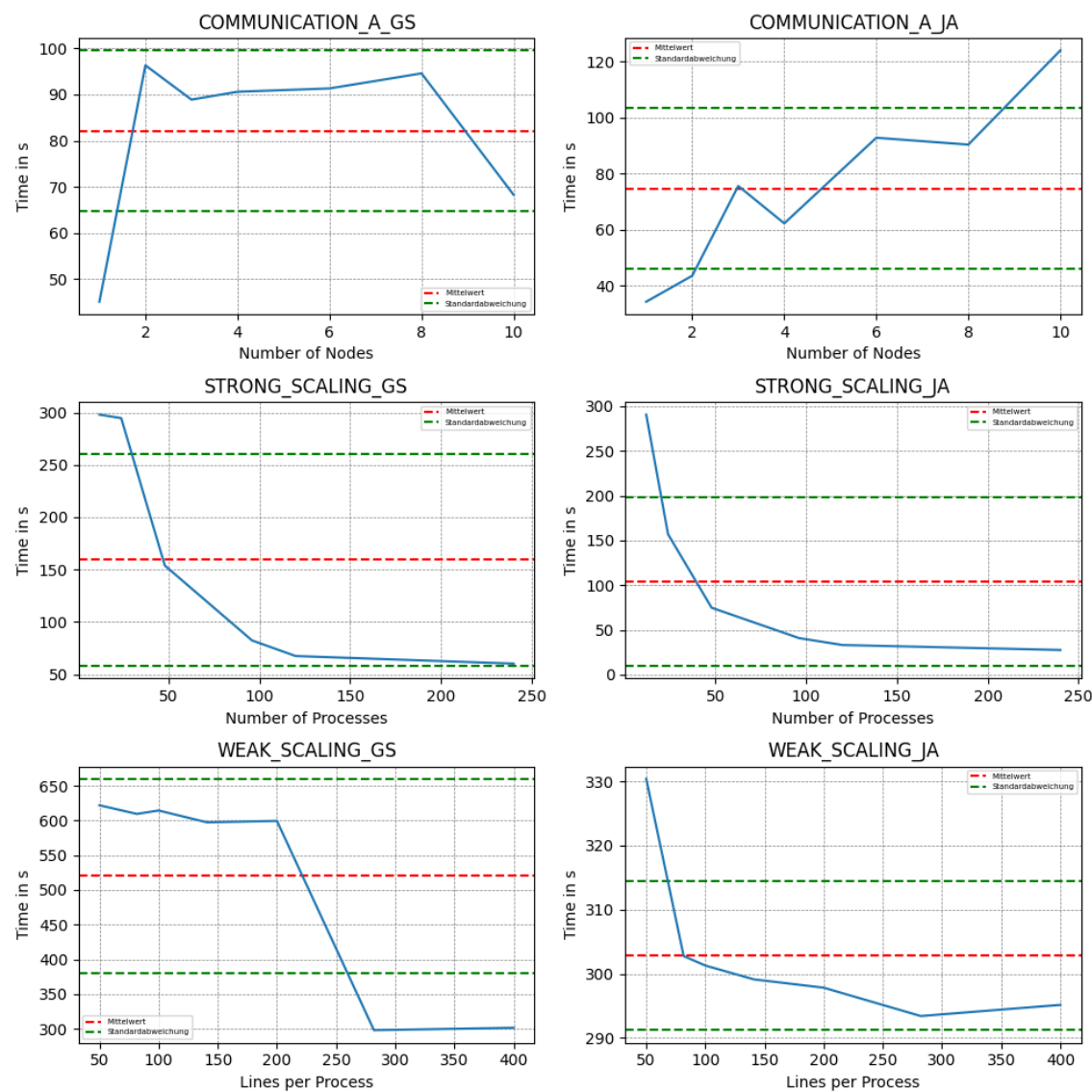


HLR #11



Weak Scaling

Vorab: Die Wahl der Interlines bezogen auf die Prozesszahl soll eine immer kleinere Anzahl von zu bearbeitenden Zeilen pro Prozess liefern.

Man kann im Weak Scaling mit GS erkennen, dass im Durchschnitt mit steigender Anzahl von Zeilen pro Prozess die Rechendauer sinkt. Dies gilt auch für Jacobi. Bei beiden Implementationen, ab einem gewissen Punkt (bei beiden ca. 300 Lines per Process), ändert sich nichts mehr (GS) oder verlängert sich die Rechendauer sogar wieder (JA). Dies lässt sich daran begründen, dass bei mehr Zeilen ein Prozess länger rechnet und weniger auf seinen Kommunikationspartner warten muss. Es gilt nämlich, dass die Konfiguration mit den meisten Interlines die wenigsten Prozesse hat und das Senden und aufeinander Warten zum Empfangen sehr viel Zeit kostet.

Strong Scaling

Hier wird geschaut, ob mit gleicher Menge an zu bearbeitenden Zeilen mit steigenden Prozessen die Dauer verkürzt wird, und hier passen beide Implementationen zur Erwartung, dass mehr Prozesse, trotz aufeinander wartende und verlangsamende Kommunikation, durch (semi)parallele Berechnung auf der Matrix durchschnittlich weniger Zeit braucht. Die gewonnene Zeit wird mit steigenden Prozessen immer weniger, weil Prozesse immer weniger Zeilen anderen Prozessen vergeben.

Kommunikation und Teilnutzung der Knoten

Diese Werte sind von der Gruppe von Patrick ausgeliehen, weil unsere Implementation bei Kommunikation zwischen Knoten viele Probleme hat.

Hier sollte im Durchschnitt mit mehr Knoten die Rechendauer länger brauchen. Der Berechnungsvorteil, den man von steigender Knotenanzahl gewinnen würde, wird vom Kommunikationsoverhead ausgeglichen. Einstürze in der Messung sind Messfehler.