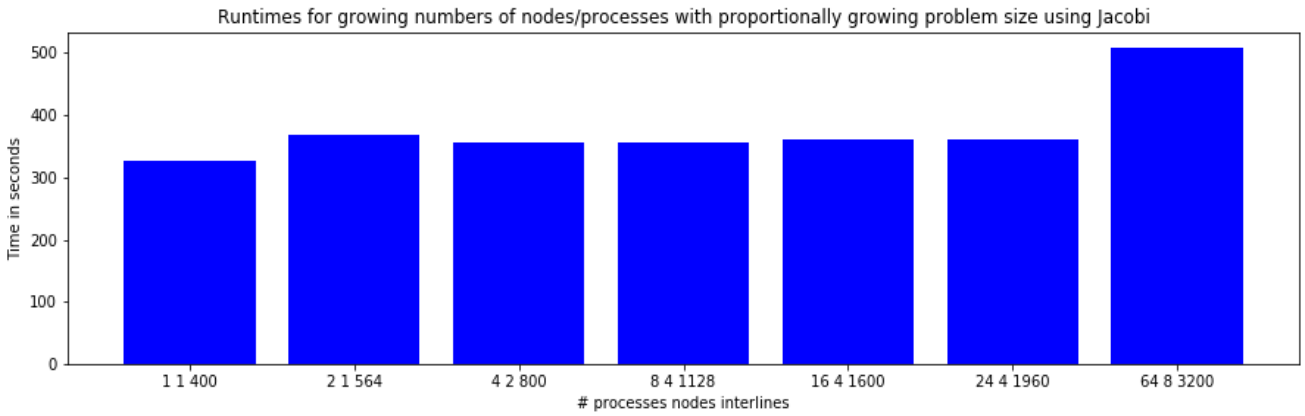
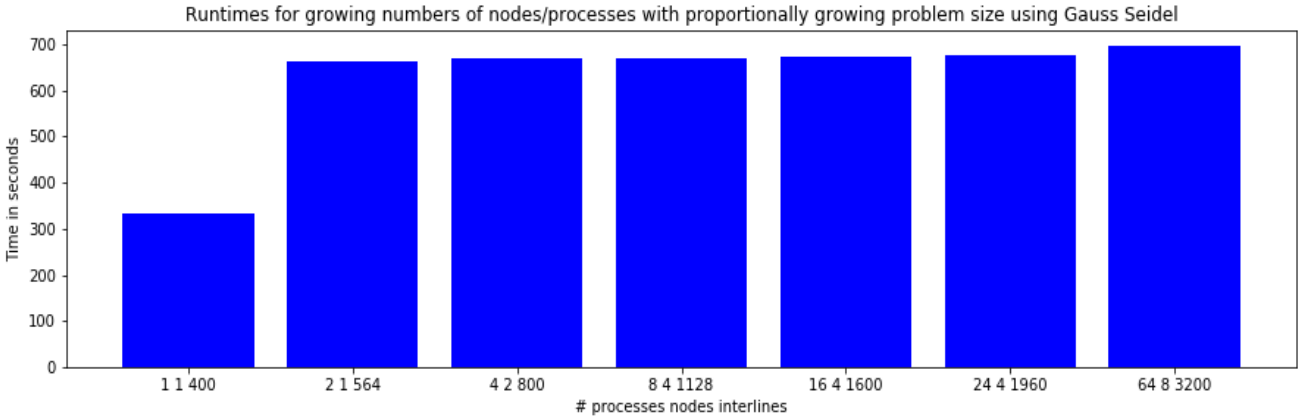
# Weak Scaling




Zugehörige Tabelle:

# NPROCS NNODES ILINES, TIME
1 1 400, 327.9332
2 1 564, 368.1801
4 2 800, 357.3738
8 4 1128, 356.7986
16 4 1600, 360.5349
24 4 1960, 361.9641
64 8 3200, 508.7017

Die Laufzeiten sind für die Durchläufe 2-6 ungefähr gleich, was bei weak scaling zu erwarten ist, da die Problemgröße mit der Anzahl der Prozessoren wächst. Nur der erste und der letzte Durchlauf fallen aus der Reihe.
Der Grund für die Unregelmäßigkeit beim ersten Lauf ist vermutlich die Tatsache, dass nur ein Prozessor benutzt wird, weshalb die sequentielle Variante des Programms ausgeführt wird. Dadurch fällt der Kommunikationsoverhead weg, weshalb die Laufzeit bei dieser Problemgröße kleiner ist als bei allen anderen.
Die Unregelmäßigkeit beim letzten Durchlauf kommt vermutlich durch den erhöhten Kommunikationsaufwand im Vergleich zu den anderen Durchläufen.

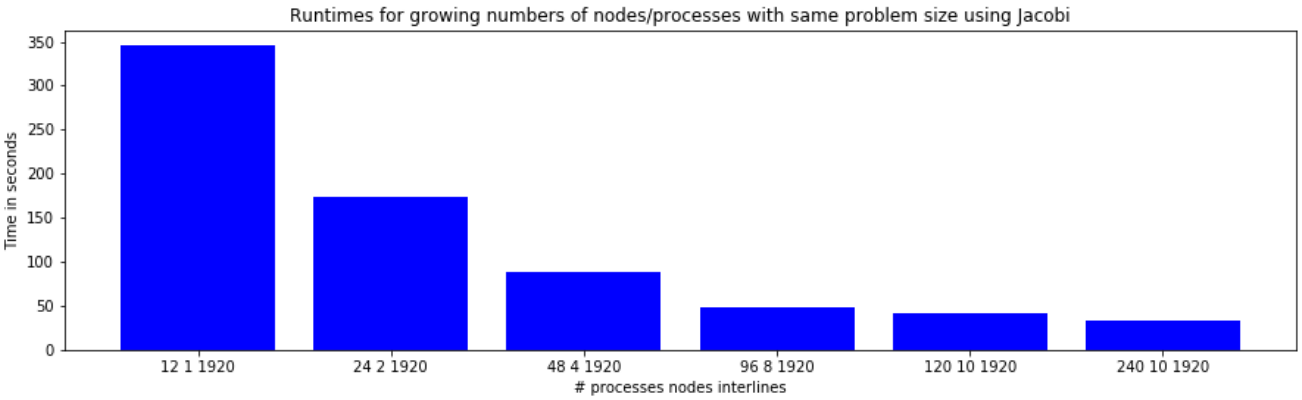



Zugehörige Tabelle:

# NPROCS NNODES ILINES, TIME
1 1 400, 334.2272
2 1 564, 661.5958
4 2 800, 669.3090
8 4 1128, 668.1684
16 4 1600, 673.9965
24 4 1960, 675.2247
64 8 3200, 696.2339

Die Laufzeiten für alle Durchläufe bis auf den ersten sind in etwa gleich, allerdings sind die Laufzeiten im Vergleich zu Jacobi deutlich schlechter, da das Kommunikationsmuster der Gauss-Seidel-Variante etwas komplizierter/ineffizienter ist.
Der erste Durchlauf ist deutlich kürzer, weil dieser nur einen Prozessor benutzt, weshalb die sequentielle Variante ausgeführt wird. Es fällt also auch hier der Kommunikationsoverhead weg.

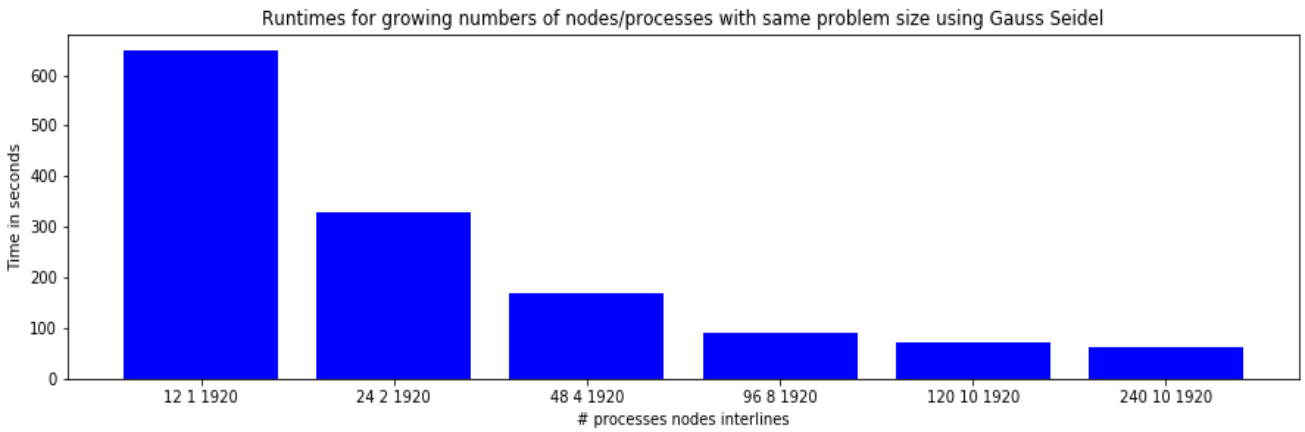
## Strong Scaling




Zugehörige Tabelle:

# NPROCS NNODES ILINES, TIME
12 1 1920, 346.0330
24 2 1920, 173.3743
48 4 1920, 87.9076
96 8 1920, 49.0396
120 10 1920, 41.7916
240 10 1920, 33.2469

Es ist zu beobachten, dass die Laufzeit mit steigender Knoten-/Prozessorzahl abnimmt, was bei strong scaling auch zu erwarten ist. Vergleicht man z.B. den ersten Lauf mit dem letzten fällt auf, dass der letzte Durchlauf ca. 10-11 mal so schnell ist wie der erste. Das ist bei 10-facher Knotenanzahl auch verständlich, allerdings wurden die Prozesse um das 20-fache angehoben, weshalb wir eigentlich von einem stärkeren Speedup ausgegangen wären. Dies ist offensichtlich nicht der Fall, jedoch ist nicht klar, ob diese Tatsache auf einen Mangel an unserem Programm zurückzuführen ist oder nicht.

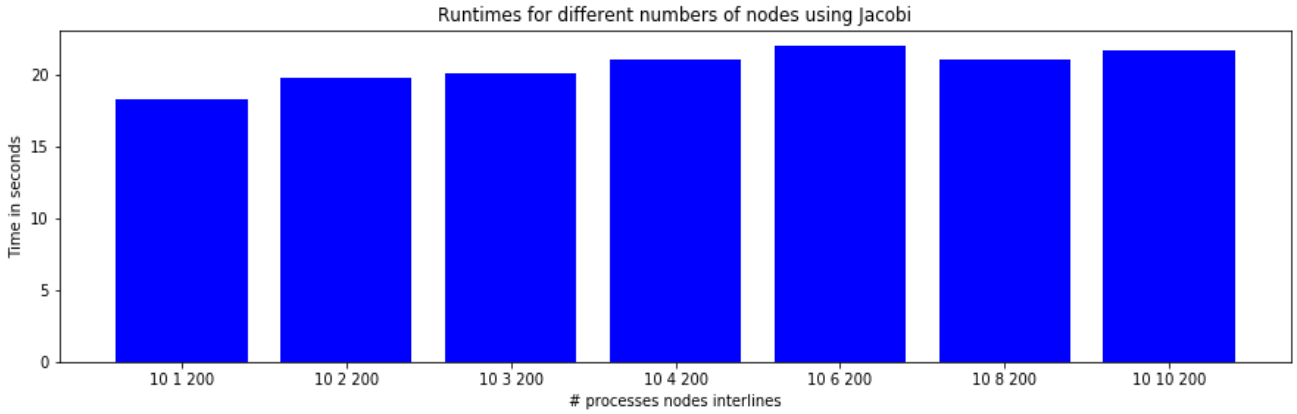



Zugehörige Tabelle:

# NPROCS NNODES ILINES, TIME
12 1 1920, 648.9297
24 2 1920, 328.9762
48 4 1920, 168.8006
96 8 1920, 89.7932
120 10 1920, 72.9899
240 10 1920, 61.1567

Es ist zu beobachten, dass die Laufzeit mit steigender Knoten-/Prozessorzahl abnimmt, was bei strong scaling auch zu erwarten ist, allerdings ist die Laufzeit auch hier wieder schlechter im Vergleich zu Jacobi aus den bereits genannten Gründen.
Ansonsten lassen sich auch hier dieselben Beobachtungen wie bei der vorangegangenen Graphik treffen.

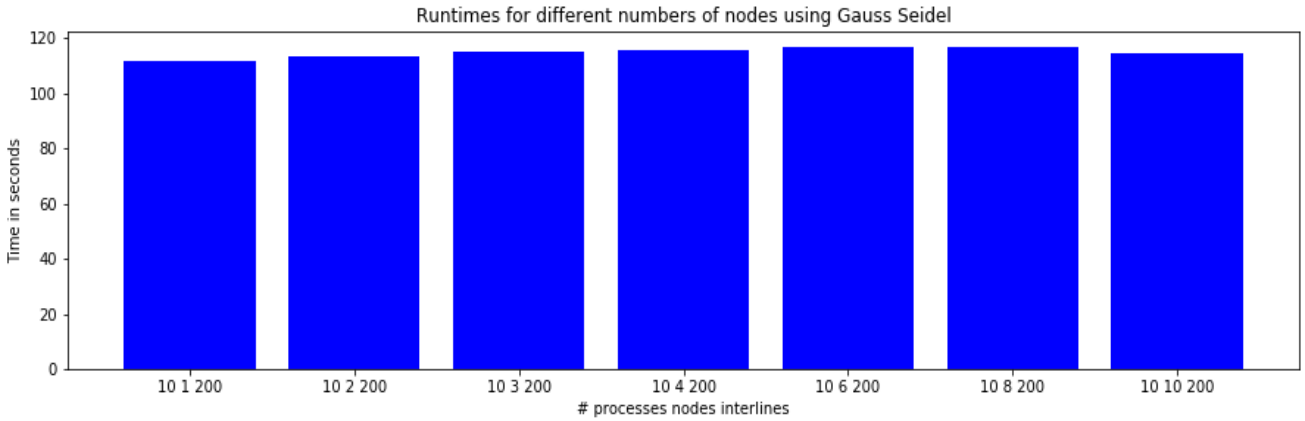
# Communication




Zugehörige Tabelle:

# NPROCS NNODES ILINES, TIME
10 1 200, 18.2703
10 2 200, 19.8351
10 3 200, 20.1341
10 4 200, 21.0638
10 6 200, 22.0612
10 8 200, 21.0842
10 10 200, 21.7438

Für diese Messungen wurde jeweils die Knotenzahl erhöht, sowohl Prozessanzahl als auch Interlines blieben gleich. Es ist zu beobachten, dass die Laufzeit mit steigender Knotenanzahl fast immer zunahm. Das ist vermutlich darauf zurückzuführen, dass die Kommunikation zwischen verschiedenen Knoten aufwendiger ist als die Kommunikation zwischen einzelnen Prozessen auf einem einzigen Knoten.




Zugehörige Tabelle:

# NPROCS NNODES ILINES, TIME
10 1 200, 111.9643
10 2 200, 113.4848
10 3 200, 115.1446
10 4 200, 115.6873
10 6 200, 116.8097
10 8 200, 116.7105
10 10 200, 114.4558

Auch hier kann eine ähnliche Beobachtung gemacht werden wie bei der vorherigen Graphik. Allerdings sind die Laufzeiten erneut schlechter, die Gründe hierfür wurden bereits genannt.