

Parallelisierungsschema Jakobi/Gauß-Seidel

Datenaufteilung:

Die Daten der Matrix werden zeilenweise auf die einzelnen Prozesse aufgeteilt.

Parallelisierungsschema Jakobi:

Die Berechnung der Werte erfolgt spaltenweise.

Jedem Prozess ist dabei bekannt, ob er einen Prozess über oder unter sich hat.

Vor der Berechnung sendet der jeweilige Prozess den Zelleninhalt seiner untersten Zeile an den darunterliegenden Prozess und erhält den Zelleninhalt der untersten Zeile vom darüberliegenden.

Selbiges Verfahren gilt dabei für die Kommunikation von oben nach unten.

Nun hat jeder Prozess die erforderlichen Daten, um den neuen Inhalt der jetzigen Spalte zu berechnen.

Nach der Rechnung werden die Prozesse durch eine Barriere synchronisiert, woraufhin in die nächste Spalte gewechselt wird.

Parallelisierungsschema Gauß-Seidel:

In diesem Verfahren laufen wir iterativ von der oberen linken in die untere rechte Ecke der Matrix. Dadurch ergibt sich das Problem, dass n procs am Start nur ein mögliches Feld berechnen kann, woraus sich eine Verzögerung des Startzeitpunkts der Berechnung der einzelnen Prozesse ergibt, weshalb die Ressourcen am Anfang und Ende der Matrix nicht voll ausgenutzt werden können.

Nachdem der erste Prozess seine erste Zelle berechnet hat, versendet er diese an den nächsten Prozess, woraufhin dieser mit dem Rechnen anfangen kann, während der erste Prozess die Rechnung bei der zweiten Spalte fortsetzt usw.

Ist der erste Prozess am Ende seiner Zeile angekommen, so rechnet er wieder von vorne los ohne auf den letzten Prozess zu warten.

0	1	2	3	4	5	6	7	8	9
1	11	12	13	14	15	16	17	18	19
2	21	22	23	24	25	26	27	28	29
3	31	32	33	34	35	36	37	38	39
4	41	42	43	44	45	46	47	48	49
5	51	52	53	54	55	56	57	58	59
6	61	62	63	64	65	66	67	68	69
7	71	72	73	74	75	76	77	78	79
8	81	82	83	84	85	86	87	88	89
9	91	92	93	94	95	96	97	98	99

Beispiel: In der gezeigten Grafik ist der erste Prozess durch die Farbe Rot, der zweite durch Gelb und der dritte durch Grün dargestellt. Zuerst berechnet der erste Prozess die Zelle M_{11} , bevor er das Ergebnis dieser Rechnung an den zweiten Prozess, der daraufhin mit der Berechnung von M_{21} anfängt, sendet und mit der Berechnung der Zelle M_{12} fortfährt.

Abbruchbedingung:

Anzahl der Iterationen:

Jeder Prozess zählt die Anzahl der Iterationen und weiß somit, wann terminiert werden muss. Ist ein Prozess am Ende seiner Zeilen angekommen, so wird die Anzahl inkrementiert. Bevor er mit der Berechnung des nächsten Durchgangs anfängt, prüft er, ob die gespeicherte Anzahl größer der angegebenen Anzahl ist.

Genauigkeit Jacobi:

Jeder Prozess speichert das jeweilige maxresiduum. Nach einer kompletten Iteration teilt jeder Prozess dem Master-Prozess sein maxresiduum mit (Reduce). Allein der Master-Prozess überprüft nun, ob maxresiduum unterhalb der Abbruchbedingung liegt und teilt allen Prozessen mit, ob eine weitere Iteration durchgeführt werden muss.

Genauigkeit Gauß-Seidel:

Nur der letzte Prozess kann die Überprüfung auf Genauigkeit vornehmen. Um nicht an Performance zu verlieren, rechnen die anderen Prozesse ohne Pause direkt nach Ende der Iteration weiter. Daraus folgt jedoch, dass sich die Prozesse mit niedrigeren Rängen die Daten der vorherigen Iteration merken müssen. Die läuft naturgemäß auf einen höheren Speicherverbrauch hinaus.

Hat der letzte Prozess die erforderliche Genauigkeit erreicht, so teilt er dies allen anderen mit und kümmert sich um die Ausgabe der Daten, die er von den anderen Prozessen aus der entsprechenden Iteration erhält.