

Exercise 21.3 Consider a parallel DBMS in which each relation is stored by horizontally partitioning its tuples across all disks.

```
Employees(eid: integer, did: integer, sal: real)
Departments(did: integer, mgrid: integer, budget: integer)
```

The *mgrid* field of Departments is the *eid* of the manager. Each relation contains 20-byte tuples, and the *sal* and *budget* fields both contain uniformly distributed values in the range 0 to 1,000,000. The Employees relation contains 100,000 pages, the Departments relation contains 5,000 pages, and each processor has 100 buffer pages of 4,000 bytes each. The cost of one page I/O is t_d , and the cost of shipping one page is t_s ; tuples are shipped in units of one page by waiting for a page to be filled before sending a message from processor i to processor j . There are no indexes, and all joins that are local to a processor are carried out using a sort-merge join. Assume that the relations are initially partitioned using a round-robin algorithm and that there are 10 processors.

For each of the following queries, describe the evaluation plan briefly and give its cost in terms of t_d and t_s . You should compute the total cost across all sites as well as the ‘elapsed time’ cost (i.e., if several operations are carried out concurrently, the time taken is the maximum over these operations).

1. Find the highest paid employee.
2. Find the highest paid employee in the department with *did* 55.
3. Find the highest paid employee over all departments with *budget* less than 100,000.
4. Find the highest paid employee over all departments with *budget* less than 300,000.
5. Find the average salary over all departments with *budget* less than 300,000.
6. Find the salaries of all managers.
7. Find the salaries of all managers who manage a department with a budget less than 300,000 and earn more than 100,000.
8. Print the *eids* of all employees, ordered by increasing salaries. Each processor is connected to a separate printer, and the answer can appear as several sorted lists, each printed by a different processor, as long as we can obtain a fully sorted list by concatenating the printed lists (in some order).

Answer 21.3 The round-robin partitioning implies that every tuple has a equal probability of residing at each processor. Moreover, since the *sal* field of Employees and *budget* field of Departments are uniformly distributed on 0 to 1,000,000, each processor must also have a uniform distribution on this range. Also note that processing a partial page incurs the same cost as processing an entire page and the cost of writing out the result is uniformly ignored. Finally, recall that elapsed time is the maximum time taken for any one processor to complete its task.

1. Find the highest paid employee.
Plan: Conduct a complete linear scan of the Employees relation at each processor retaining only the tuple with the highest value in *sal* field. All processors except one then

send their result to a chosen processor which selects the tuple with the highest value of *sal*.

$$\begin{aligned}
 \text{Total Cost} &= (\# \text{ CPU}s) * (\text{Emp pg} / \text{CPU}) * (I/O \text{ cost}) \\
 &+ (\# \text{ CPU}s - 1) * (\text{send cost}) \\
 &= (10 * 10,000 * t_d) + (9 * t_s) \\
 &= 100,000 * t_d + 9 * t_s
 \end{aligned}$$

$$\text{Elapsed Time} = 10,000 * t_d + t_s$$

2. Find the highest paid employee in the department with *did* 55.

Plan: Conduct a complete linear scan of the Employees relation at each processor retaining only the tuple with the highest value in *sal* field and a *did* field equal to 55. All processors except one then send their result to a chosen processor which selects the tuple with the highest value of *sal*.

Total Cost: The answer is the same as for part 1 above. Even if no qualifying tuples are found at a given processor, a page should still be sent from nine processors to a chosen tenth. The page will either contain a real tuple or if a processor fails to find any tuple with *did* equal to 55, a generated tuple with *sal* equal to -1 will suffice. Note that the chosen processor must also account for the case where no tuple qualifies, simply by ignoring any tuple with *sal* equal to -1 in its final selection.

Elapsed Time: The elapsed time is also the same as for part 1 above.

3. Find the highest paid employee over all departments with *budget* less than 100,000.

Plan: First, conduct a complete linear scan of the Departments relation at each processor retaining only the *did* fields from tuples with *budget* less than 100,000. Recall that Departments is uniformly distributed on the *budget* field from 0 to 1,000,000, thus each processor will retain only 10% of its 500 Departments pages. Since the *did* field is 1/3 of a Departments tuple, the scan will result in approximately 16.7 pages which rounds up to 17.

Second, each processor sends its 17 pages of retained *did* field tuples to every other processor which subsequently stores them. 10 processors send 17 pages to 9 other processors for a total of 1,530 sends. After sending, each processor has 170 (partially filled) pages of Departments tuples.

Third, each processor joins the *did* field tuples with the Employees relation retaining only the joined tuple with the highest value in the *sal* field. Let $M = 170$ represent the number of Departments pages and $N = 10,000$ represent the number of Employees pages at each processor. Since the number of buffer pages, $100 \geq \sqrt{N}$, the refined Sort-Merge may be used for a join cost of 30,510 at each processor.

Fourth, all processors except one then send their result to a chosen processor which selects the tuple with the highest value of *sal*.

$$\begin{aligned}
 \text{Total Cost} &= \text{scan Dept for tuples with budget} < 100,000 \\
 &+ \text{sending did field tuples from 10 processors to 9 others}
 \end{aligned}$$

$$\begin{aligned}
& + \text{ storing did field tuples at each processor} \\
& + \text{ joining with Emp and selecting max(sal) tuple} \\
& + \text{ sending local results to the chosen processor} \\
\\
& = (\# \text{ CPU scanning}) * (\text{Dept pgs/CPU}) * (I/O \text{ cost}) \\
& \quad 10 * 500 * t_d \\
& \quad 5,000 * t_d \\
& + (\# \text{ CPU sending}) * (\# \text{ CPU receiving}) * (17 \text{ did pgs}) * t_s \\
& \quad 10 * 9 * 17 * t_s \\
& \quad 1,530 * t_s \\
& + (\# \text{ CPU storing}) * (170 \text{ did pgs}) * (I/O \text{ cost}) \\
& \quad 10 * 170 * t_d \\
& \quad 1,700 * t_d \\
& + (\# \text{ CPU joining}) * (\text{join cost}) \\
& \quad 10 * (3 * (170 + 10,000) * t_d) \\
& \quad 10 * 30,510 * t_d \\
& \quad 305,100 * t_d \\
& + (\# \text{ CPUs} - 1) * (\text{send cost}) \\
& \quad 9 * t_s \\
& = 5,000 * t_d + 1,530 * t_s + 1,700 * t_d + 305,100 * t_d + 9 * t_s \\
& = 311,800 * t_d + 1,539 * t_s
\end{aligned}$$

$$\begin{aligned}
\text{Elapsed Time} & = 500 * t_d + 153 * t_s + 170 * t_d + 30,510 * t_d + t_s \\
& = 31,180 * t_d + 154 * t_s
\end{aligned}$$

4. Find the highest paid employee over all departments with *budget* less than 300,000.

Plan: The evaluation of this query is identical to that in part 3 except that the probability of a Departments tuple's *budget* field being selected in step one is multiplied by three. There are then 50 pages retained by each processor and sent to every other processor for joins and maximum selection.

$$\begin{aligned}
\text{Total Cost} & = \text{scan Dept for tuples with budget} < 300,000 \\
& + \text{ sending did field tuples from 10 processors to 9 others} \\
& + \text{ storing did field tuples at each processor} \\
& + \text{ joining with Emp and selecting max(sal) tuple} \\
& + \text{ sending local results to the chosen processor} \\
& = (\# \text{ CPU scanning}) * (\text{Dept pgs/CPU}) * (I/O \text{ cost}) \\
& \quad 10 * 500 * t_d \\
& \quad 5,000 * t_d
\end{aligned}$$

$$\begin{aligned}
& + (\# \text{ CPU sending}) * (\# \text{ CPU receiving}) * (50 \text{ did pgs}) * t_s \\
& \quad 10 * 9 * 50 * t_s \\
& \quad 4,500 * t_s \\
& + (\# \text{ CPU storing}) * (500 \text{ did pgs}) * (I/O \text{ cost}) \\
& \quad 10 * 500 * t_d \\
& \quad 5,000 * t_d \\
& + (\# \text{ CPU joining}) * (\text{join cost}) \\
& \quad 10 * (3 * (500 + 10,000) * t_d) \\
& \quad 10 * 31,500 * t_d \\
& \quad 315,000 * t_d \\
& + (\# \text{ CPUs} - 1) * (\text{send cost}) \\
& \quad 9 * t_s \\
& = 5,000 * t_d + 4,500 * t_s + 5,000 * t_d + 315,000 * t_d + 9 * t_s \\
& = 325,000 * t_d + 4,509 * t_s
\end{aligned}$$

$$\begin{aligned}
\text{Elapsed Time} & = 500 * t_d + 450 * t_s + 500 * t_d + 31,500 * t_d + t_s \\
& = 32,500 * t_d + 451 * t_s
\end{aligned}$$

5. Find the average salary over all departments with *budget* less than 300,000.

Plan: The first two steps in evaluating this query are identical to part 4. Steps three and four differ in that the desired result is an average instead of a maximum.

First, each processor conducts a complete linear scan of the Departments relation retaining only the *did* field from tuples with a *budget* field less than 300,000. Second, each processor sends its result pages to every other processor. Third, each processor joins the *did* field tuples with the Employees relation and retains a running sum and count of the *sal* field. Fourth, each processor except one sends its sum and count to a chosen processor which divides the total sum by the total count to obtain the average. The cost is identical to part 4 above.

6. Find the salaries of all managers.

Plan: First, conduct a complete linear scan of the Departments relation at each processor retaining only the *mgrid* field for all tuples. Since the *mgrid* field is 1/3 of each tuple, there will be 167 (rounded up) resulting pages. Second, each processor sends its result pages to every other processor which subsequently stores them. Third, each processor joins the *mgrid* field tuples with Employees thus obtaining the salaries of all managers.

$$\begin{aligned}
\text{Total Cost} & = \text{scan Dept for mgrid fields} \\
& + \text{sending mgrid field tuples from 10 processors to 9 others} \\
& + \text{storing mgrid field tuples at each processor} \\
& + \text{joining with Emp} \\
& = (\# \text{ CPU scanning}) * (\text{Dept pgs/CPU}) * (I/O \text{ cost}) \\
& \quad 10 * 500 * t_d
\end{aligned}$$

$$\begin{aligned}
& 5,000 * t_d \\
+ & (\# \text{ CPU sending}) * (\# \text{ CPU receiving}) * (167 \text{ mgrid pgs}) * t_s \\
& 10 * 9 * 167 * t_s \\
& 15,030 * t_s \\
+ & (\# \text{ CPU storing}) * (1,670 \text{ mgrid pgs}) * (I/O \text{ cost}) \\
& 10 * 1,670 * t_d \\
& 16,700 * t_d \\
+ & (\# \text{ CPU joining}) * (\text{join cost}) \\
& 10 * (3 * (1,670 + 10,000) * t_d) \\
& 10 * 35,010 * t_d \\
& 350,100 * t_d \\
= & 5,000 * t_d + 15,030 * t_s + 16,700 * t_d + 350,100 * t_d \\
= & 386,830 * t_d + 15,030 * t_s
\end{aligned}$$

$$\begin{aligned}
\text{Elapsed Time} &= 500 * t_d + 1,503 * t_s + 1,670 * t_d + 35,010 * t_d \\
&= 38,683 * t_d + 1,503 * t_s
\end{aligned}$$

7. Find the salaries of all managers who manage a department with a budget less than 300,000 and earn more than 100,000.

Plan: The evaluation of this query is similar to that of part 6. The additional selection condition on the *budget* field is applied in step one and serves to reduce the number of pages sent and joined in steps two and three. The additional selection condition on the *sal* field is applied during the join in step three and has no effect on the final cost.

First, conduct a complete linear scan of the Departments relation at each processor retaining only the *mgrid* field for all tuples. Since the *mgrid* field is 1/3 of each tuple and there are 150 qualifying Departments pages at each processor, there will be 50 resulting pages. Second, each processor sends its result pages to every other processor which subsequently stores them. Third, each processor joins the *mgrid* field tuples with Employees thus obtaining the salaries of all managers.

$$\begin{aligned}
\text{Total Cost} &= \text{scan Dept for mgrid fields} \\
+ & \text{sending mgrid field tuples from 10 processors to 9 others} \\
+ & \text{storing mgrid field tuples at each processor} \\
+ & \text{joining with Emp} \\
= & (\# \text{ CPU scanning}) * (\text{Dept pgs/CPU}) * (I/O \text{ cost}) \\
& 10 * 500 * t_d \\
& 5,000 * t_d \\
+ & (\# \text{ CPU sending}) * (\# \text{ CPU receiving}) * (50 \text{ mgrid pgs}) * t_s \\
& 10 * 9 * 50 * t_s \\
& 4,500 * t_s
\end{aligned}$$

$$\begin{aligned}
& + (\# \text{ CPU storing}) * (500 \text{ mgrid pgs}) * (I/O \text{ cost}) \\
& \quad 10 * 500 * t_d \\
& \quad 5,000 * t_d \\
& + (\# \text{ CPU joining}) * (\text{join cost}) \\
& \quad 10 * (3 * (500 + 10,000) * t_d) \\
& \quad 10 * 31,500 * t_d \\
& \quad 315,000 * t_d \\
& = 5,000 * t_d + 4,500 * t_s + 5,000 * t_d + 315,000 * t_d \\
& = 325,000 * t_s + 4,500 * t_s
\end{aligned}$$

$$\begin{aligned}
\text{Elapsed Time} & = 500 * t_d + 450 * t_s + 500 * t_d + 31,500 * t_d \\
& = 32,500 * t_d + 450 * t_s
\end{aligned}$$

8. Print the *eids* of all employees, ordered by increasing salaries. Each processor is connected to a separate printer, and it is acceptable to have the answer in the form of several sorted lists, each printed by a different processor, as long as we can obtain a fully sorted list by concatenating the printed lists (in some order).

Plan: At each processor, sort the Employees relation by the *sal* field and print the result. Note that the refined Sort-Merge join may be applied without the on-the-fly merge to sort at a cost of $3 * M * t_d$.

$$\begin{aligned}
\text{Total Cost} & = (\# \text{ CPU sorting}) * (\text{sort cost}) \\
& = 10 * (3 * 10,000 * t_d) \\
& = 300,000 * t_d
\end{aligned}$$

$$\text{Elapsed Time} = 30,000 * t_d$$

Exercise 21.4 Consider the same scenario as in Exercise 21.3, except that the relations are originally partitioned using range partitioning on the *sal* and *budget* fields.

Answer 21.4 Range partitioning places tuples with either a *sal* or *budget* field between 0 and 100,000 at processor 1, between 100,001 and 200,000 at the processor 2, etc. The uniform distribution of values in *sal* and *budget* implies that there are equal numbers of tuples at each processor. Then, as is Exercise 21.3, there are 10,000 Departments pages and 500 Employees pages at each processor.

1. Find the highest paid employee.

Plan: The tuple with the highest *sal* value is located at processor 10. Conduct a complete linear scan of the Employees relation there retaining the tuple with the highest value in the *sal* field.

$$\begin{aligned}
 \text{Total Cost} &= (\# \text{ of Emp pgs at CPU } 10) * (I/O \text{ cost}) \\
 &= 10,000 * t_d
 \end{aligned}$$

$$\text{Elapsed Time} = 10,000 * t_d$$

2. Find the highest paid employee in the department with *did* 55.

Plan: Since there is no guarantee that such a tuple might exist at any given processor, conduct a complete linear scan of all Employees tuples at each processor retaining the one with the highest *sal* value and *did* 55. Each processor except one should then send their result to a chosen processor which selects the tuple with the highest value in the *sal* field.

$$\begin{aligned}
 \text{Total Cost} &= (\# \text{ CPU scanning}) * (\# \text{ of Emp pgs/CPU}) * (I/O \text{ cost}) \\
 &+ (\# \text{ CPUs} - 1) * (\text{send cost}) \\
 &= 10 * 10,000 * t_d \\
 &+ 9 * t_s \\
 &= 100,000 * t_d + 9 * t_s
 \end{aligned}$$

$$\text{Elapsed Time} = 10,000 * t_d + t_s$$

3. Find the highest paid employee over all departments with *budget* less than 100,000.

Plan: Department tuples with a *budget* field less than 100,000 must be located at processor 1. The highest paid employees are located at the higher numbered processors, however; as in the part 2 above, there is no guarantee that any processor has an Employees tuple with a particular *did* field value. So, processor 1 must conduct a complete linear scan of Departments retaining only the *did* field. The results are then sent to all other processors which store and join them with the Employees relation retaining only the join tuple with the highest *sal* value. Finally, each processor except one sends the result to a chosen processor who selects the Employees tuple with the highest *sal* value.

$$\begin{aligned}
 \text{Total Cost} &= \text{scan Dept for did fields at first CPU} \\
 &+ \text{sending did field tuples from 1 CPU to 9 CPUs} \\
 &+ \text{storing did field tuples at each processor} \\
 &+ \text{joining did field tuples with Emp} \\
 &+ \text{sending local results to chosen processor} \\
 &= (\# \text{ CPUs w/budget} < 100,000) * (\# \text{ Dept pgs}) * (I/O \text{ cost}) \\
 &\quad 1 * 500 * t_d \\
 &\quad 500 * t_d \\
 &+ (\# \text{ CPU sending}) * (\# \text{ CPU receiving}) * (167 \text{ did pgs}) * t_s \\
 &\quad 1 * 9 * 167 * t_s
 \end{aligned}$$

$$\begin{aligned}
& 10 * 1,503 * t_s \\
& 15,030 * t_s \\
+ & (\# \text{ CPU storing}) * (167 \text{ did pgs}) * (I/O \text{ cost}) \\
& 10 * 167 * t_d \\
& 1,670 * t_d \\
+ & (\# \text{ CPU joining}) * (\text{join cost}) \\
& 10 * (3 * (167 + 1,000) * t_d) \\
& 10 * 3,501 * t_d \\
& 35,010 * t_d \\
+ & (\# \text{ CPUs} - 1) * (\text{send cost}) \\
& 9 * t_s \\
= & 500 * t_d + 15,030 * t_s + 1,670 * t_d + 35,010 * t_d + 9 * t_s \\
= & 37,180 * t_d + 15,039 * t_s
\end{aligned}$$

$$\begin{aligned}
\text{Elapsed Time} &= 50 * t_d + 1,503 * t_s + 167 * t_d + 3,501 * t_d + t_s \\
&= 3,718 * t_d + 1,504 * t_s
\end{aligned}$$

4. Find the highest paid employee over all departments with *budget* less than 300,000.
Plan: The plan is identical to that for part 3 above except that now the first 3 processors must create relations of *did* fields and send them to all other processors.

$$\begin{aligned}
\text{Total Cost} &= \text{scan Dept for did fields at first three CPUs} \\
+ & \text{sending did field tuples from 3 CPUs to 9 CPUs} \\
+ & \text{storing did field tuples at each processor} \\
+ & \text{joining did field tuples with Emp} \\
+ & \text{sending local results to chosen processor} \\
= & (\# \text{ CPUs w/budget} < 300,000) * (\# \text{ Dept pgs}) * (I/O \text{ cost}) \\
& 3 * 500 * t_d \\
& 1,500 * t_d \\
+ & (\# \text{ CPU sending}) * (\# \text{ CPU receiving}) * (170 \text{ did pgs}) * t_s \\
& 3 * 9 * 170 * t_s \\
& 3 * 1,530 * t_s \\
& 4,590 * t_s \\
+ & (\# \text{ CPU storing}) * (510 \text{ did pgs}) * (I/O \text{ cost}) \\
& 10 * 510 * t_d \\
& 5,100 * t_d \\
+ & (\# \text{ CPU joining}) * (\text{join cost}) \\
& 10 * (3 * (510 * 1,000) * t_d) \\
& 10 * 4,530 * t_d
\end{aligned}$$

$$\begin{aligned}
& 45,030 * t_d \\
& + (\# \text{ CPUs} - 1) * (\text{send cost}) \\
& 9 * t_s \\
& = 1,500 * t_d + 4,590 * t_s + 5,100 * t_d + 45,030 * t_d + 9 * t_s \\
& = 51,630 * t_d + 4,599 * t_s
\end{aligned}$$

$$\begin{aligned}
\text{Elapsed Time} &= 500 * t_d + 1,530 * t_s + 510 * t_d + 4,530 * t_d + t_s \\
&= 5,540 * t_d + 1,531 * t_s
\end{aligned}$$

5. Find the average salary over all departments with *budget* less than 300,000.

Plan: This query is similar to part 4 above. The difference is that instead of selecting the highest salary during the join and reporting to a chosen processor, each processor retains a running sum of the *sal* field and count of joined tuples. The chosen processor then computes the total sum and divides by the total count to obtain the average. Note that the costs are identical to part 4.

6. Find the salaries of all managers.

Plan: Employees tuples with an *eid* field equal to a *mgrid* field of a Departments relation may be stored anywhere. Each processor should conduct a complete linear scan of its Departments tuples retaining only the *mgrid* field. Then, each processor sends the result to all others who subsequently store the *mgrid* relation. Next, each processor joins the *mgrid* relation with Employees retaining only the *sal* field of joined tuples.

$$\begin{aligned}
\text{Total Cost} &= \text{scan Dept for mgrid fields} \\
&+ \text{sending did field tuples from 10 CPUs to 9 CPUs} \\
&+ \text{storing did field tuples at each processor} \\
&+ \text{joining mgrid field tuples with Emp} \\
&+ \text{sending local results to chosen processor} \\
&= (\# \text{ CPUs scanning}) * (\# \text{ Dept pgs}) * (I/O cost) \\
&10 * 500 * t_d \\
&5,000 * t_d \\
&+ (\# \text{ CPU sending}) * (\# \text{ CPU receiving}) * (170 \text{ mgrid pgs}) * t_s \\
&10 * 9 * 170 * t_s \\
&10 * 1,530 * t_s \\
&15,300 * t_s \\
&+ (\# \text{ CPU storing}) * (1,700 \text{ did pgs}) * (I/O cost) \\
&10 * 1,700 * t_d \\
&17,000 * t_d \\
&+ (\# \text{ CPU joining}) * (\text{join cost}) \\
&10 * (3 * (1,700 * 1,000) * t_d) \\
&81,000 * t_d
\end{aligned}$$

$$\begin{aligned}
& + (\# \text{ CPUs} - 1) * (\text{send cost}) \\
& \quad 9 * t_s \\
& = 5,000 * t_d + 15,300 * t_s + 17,000 * t_d + 81,000 * t_d + 9 * t_s \\
& = 103,000 * t_d + 15,309 * t_s
\end{aligned}$$

$$\begin{aligned}
\text{Elapsed Time} & = 50 * t_d + 1,530 * t_s + 1,700 * t_d + 8,100 * t_d + t_s \\
& = 9,850 * t_d + 1,531 * t_s
\end{aligned}$$

7. Find the salaries of all managers who manage a department with a budget less than 300,000 and earn more than 100,000.

Plan: Department tuples with *budget* less than 300,000 are located at the first three processors. Employees tuples with a *sal* fields greater than 100,000 are located at processors 2 through 10. Conduct a complete linear scan of all Department tuples retaining only the *mgrid* field of tuples with a *budget* field less than 300,000. Send the new *mgrid* relation to processors 2 through 10 which subsequently store them. Next, processors 2 through 10 join the new *mgrid* relation with Employees to obtain the desired result.

$$\begin{aligned}
\text{Total Cost} & = \text{scan Dept for mgrid fields at first thirty CPUs} \\
& + \text{sending mgrid field tuples from 1 CPU to 9 CPUs} \\
& + \text{sending mgrid field tuples from 2 CPUs to 8 CPUs} \\
& + \text{storing mgrid field tuples at 9 CPUs} \\
& + \text{joining mgrid field tuples with Emp in 9 CPUs} \\
& = (\# \text{ CPUs scanning}) * (\# \text{ Dept pgs/CPU}) * (I/O cost) \\
& \quad 3 * 500 * t_d \\
& \quad 1,500 * t_d \\
& + (\# \text{ CPU sending}) * (\# \text{ CPU receiving}) * (170 \text{ mgrid pgs}) * t_s \\
& \quad 1 * 9 * 170 * t_s \\
& \quad 1 * 1,530 * t_s \\
& \quad 1,530 * t_s \\
& + (\# \text{ CPU sending}) * (\# \text{ CPU receiving}) * (170 \text{ mgrid pgs}) * t_s \\
& \quad 2 * 8 * 170 * t_s \\
& \quad 2 * 1,360 * t_s \\
& \quad 2,270 * t_s \\
& + (\# \text{ CPU storing}) * (510 \text{ mgrid pgs}) * (I/O cost) \\
& \quad 9 * 510 * t_d \\
& \quad 4,590 * t_d \\
& + (\# \text{ CPU joining}) * (\text{join cost}) \\
& \quad 9 * (3 * (510 + 1,000)) * t_d \\
& \quad 9 * 4,530 * t_d \\
& \quad 40,770 * t_d
\end{aligned}$$