# Out-of-Core Computation of the QR Factorization on Multi-Core Processors

Mercedes Marqués
**Enrique S. Quintana-Ortí**
Gregorio Quintana-Ortí

Universidad Jaime I de Castellón (España)

Robert A. van de Geijn

The University of Texas at Austin

Euro-Par – August, 2009

# Motivation

Linear least-quares problems

$$\min_x \|Ax = b\|,$$

with $A \rightarrow m \times n$, $m > n$, dense and $n \rightarrow O(10,000 - 100,000)$ appear in the estimation of the Earth gravitational field

- Only two years ago, these problems were considered "large" and a cluster was employed for their solution
- Current multi-core processors provide sufficient *computational power* to tackle them...
  - Computational cost is $O(n^3) \rightarrow$ be patient
  - Storage cost is $O(n^2) \rightarrow$ use disk

# Motivation

## Conventional wisdom against use of disk

- Disk is too slow to feed the processor
- OS can deal with data on disk, but careful design is required to reduce I/O $\rightarrow$ Out-of-Core (OOC) algorithms
- Programming OOC is cumbersome (e.g., asynchronous I/O)

Not true for linear algebra problems!

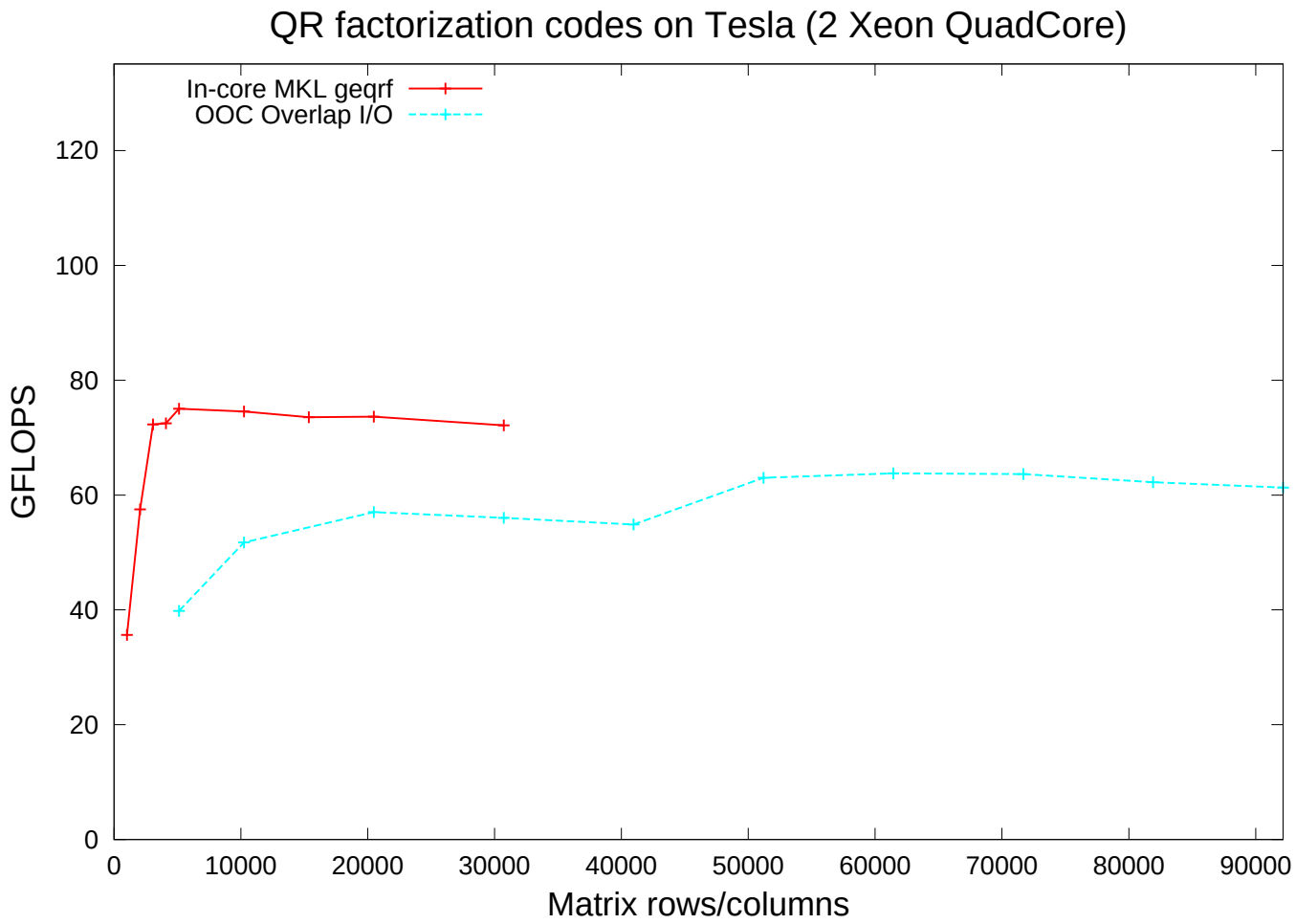
# Motivation

## Conventional wisdom against use of disk

- Disk is too slow to feed the processor
- OS can deal with data on disk, but careful design is required to reduce I/O $\rightarrow$ Out-of-Core (OOC) algorithms
- Programming OOC is cumbersome (e.g., asynchronous I/O)

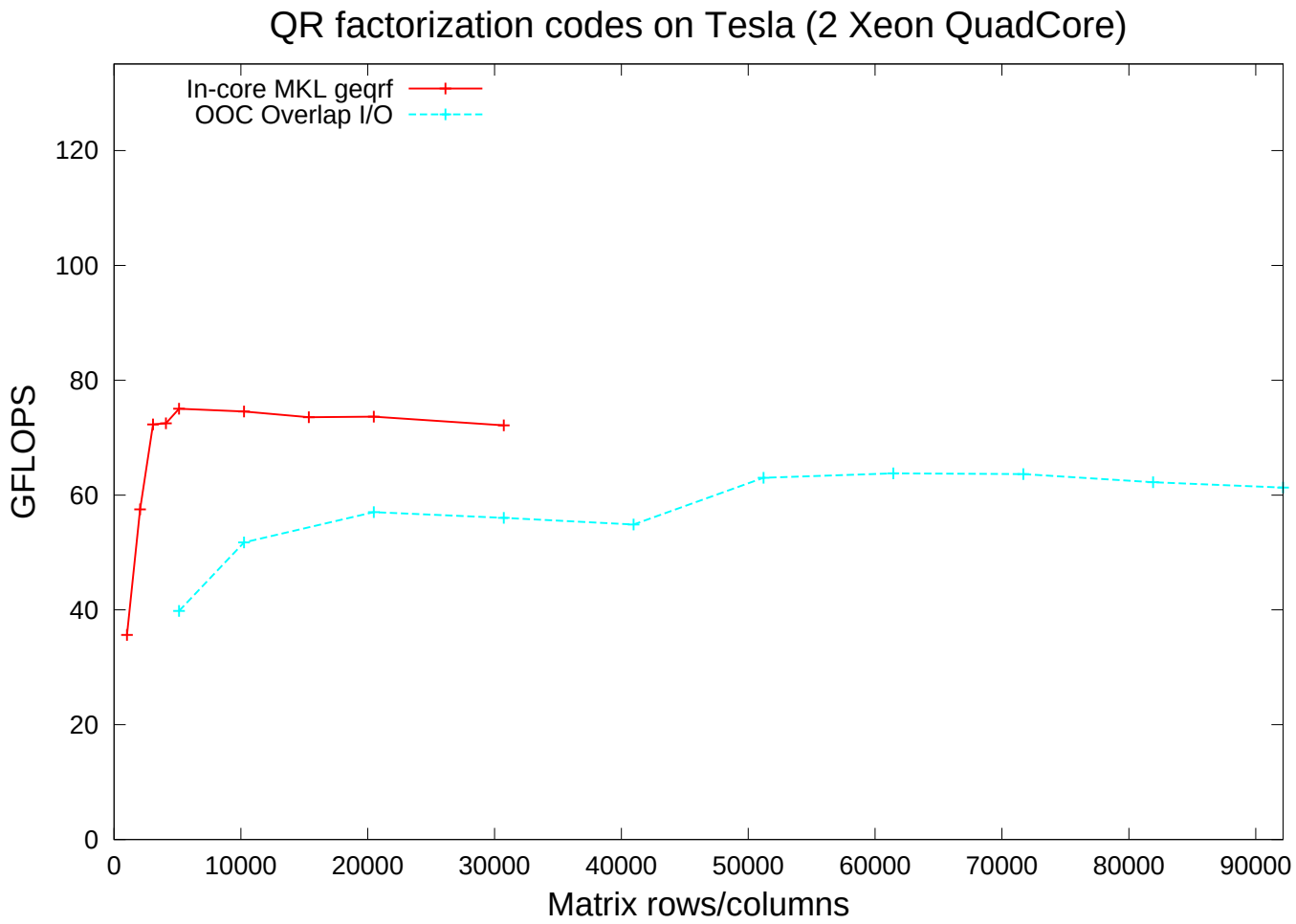
Not true for linear algebra problems!

# A preview...




Solution of a 100,000×100,000 linear least squares problem in 6 hours and 10 minutes!

# A preview. . .




Solution of a 100,000×100,000 linear least squares problem in 6 hours and 10 minutes!

# Outline

# The QR factorization

**Definition**

Given $A \rightarrow m \times n$, $m > n$, compute

$$A = Q \cdot R,$$

with $Q \rightarrow m \times m$ orthogonal and $R \rightarrow m \times n$ upper triangular

Algorithms should ideally be represented in a way that captures how we reason about them

# The QR factorization

**Definition**

Given $A \rightarrow m \times n$, $m > n$, compute

$$A = Q \cdot R,$$

with $Q \rightarrow m \times m$ orthogonal and $R \rightarrow m \times n$ upper triangular

Algorithms should ideally be represented in a way that captures how we reason about them

# The QR factorization

**Algorithm:** $[A] := \mathrm{QR\_BLK}(A)$

**Partition** $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right)$

**where** $A_{TL}$ **is** $0 \times 0$

**while** $n(A_{BR}) \neq 0$ **do**

**Determine block size** $b$

**Repartition**

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

**where** $A_{11}$ **is** $b \times b$

---

$$\left(\begin{array}{c} A_{11} \\ A_{21} \end{array}\right) := \mathrm{QR\_UNB}\left(\begin{array}{c} A_{11} \\ A_{21} \end{array}\right)$$

$$\left(\begin{array}{c} A_{12} \\ A_{22} \end{array}\right) := \mathrm{APPLY\_QR}\left(\left(\begin{array}{c} A_{11} \\ A_{21} \end{array}\right), \left(\begin{array}{c} A_{12} \\ A_{22} \end{array}\right)\right)$$

---

**Continue with**

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

**endwhile**

# Algorithms-by-tiles and algorithms-by-blocks

## Matrix of tiles/blocks

$$A \rightarrow \begin{pmatrix} \bar{A}_{0,0} & \bar{A}_{0,1} & \bar{A}_{0,1} & \cdots & \bar{A}_{0,1} \\ \bar{A}_{1,0} & \bar{A}_{1,1} & \bar{A}_{0,1} & \cdots & \bar{A}_{0,1} \\ \bar{A}_{2,0} & \bar{A}_{2,1} & \bar{A}_{2,2} & \cdots & \bar{A}_{0,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{A}_{N-1,0} & \bar{A}_{N-1,1} & \bar{A}_{N-1,2} & \cdots & \bar{A}_{N-1,N-1} \end{pmatrix}$$

## Hierarchical organization

- Each tile is a matrix of blocks
- The tile is the unit of storage on disk
- The block is the unit of computation in the processor

## Tiled OOC QR factorization (Gunter and van de Geijn, 2005):

- Left-looking, tiled variant
- Four basic building blocks:
  - $\mathrm{QR}$, $\textsc{Apply\_QR}$ as in standard QR factorization
  - New $\mathrm{QR}_{\mathrm{TD}}$, $\textsc{Apply\_QR}_{\mathrm{TD}}$
- Careful exploitation of structure in the latter two yields procedure with the same cost as standard QR factorization
- Insert explicit I/O calls to move data between RAM and disk
- Use large tiles to occupy most of RAM
- Asynchronous I/O?

# Transparent OOC

## First stage: build list of *pending tasks*

| | | | |
|---|---|---|---|
| $\bar{A}_{0,0}$ | $\bar{A}_{0,1}$ | $\bar{A}_{0,2}$ | $\cdots$ |
| $\bar{A}_{1,0}$ | $\bar{A}_{1,1}$ | $\bar{A}_{1,2}$ | $\cdots$ |
| $\bar{A}_{2,0}$ | $\bar{A}_{2,1}$ | $\bar{A}_{2,2}$ | $\cdots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ |

$\longrightarrow$ at runtime

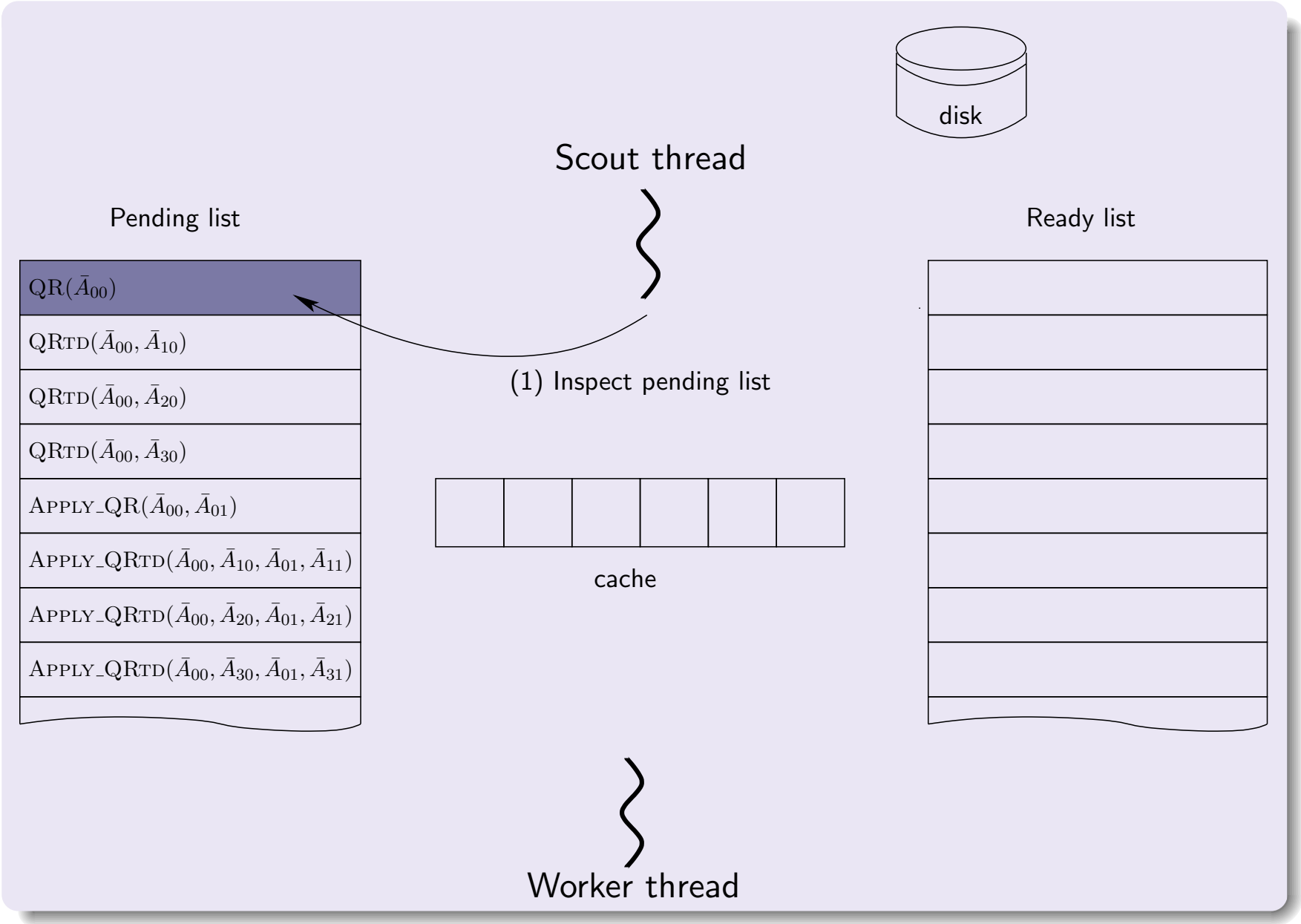
1. $\mathrm{QR}(\bar{A}_{0,0})$
2. $\mathrm{QR}_{\mathrm{TD}}(\bar{A}_{0,0}, \bar{A}_{1,0})$
3. $\mathrm{QR}_{\mathrm{TD}}(\bar{A}_{0,0}, \bar{A}_{2,0})$
4. $\vdots$
5. $\textsc{Apply\_QR}(\bar{A}_{0,0}, \bar{A}_{0,1})$
6. $\vdots$

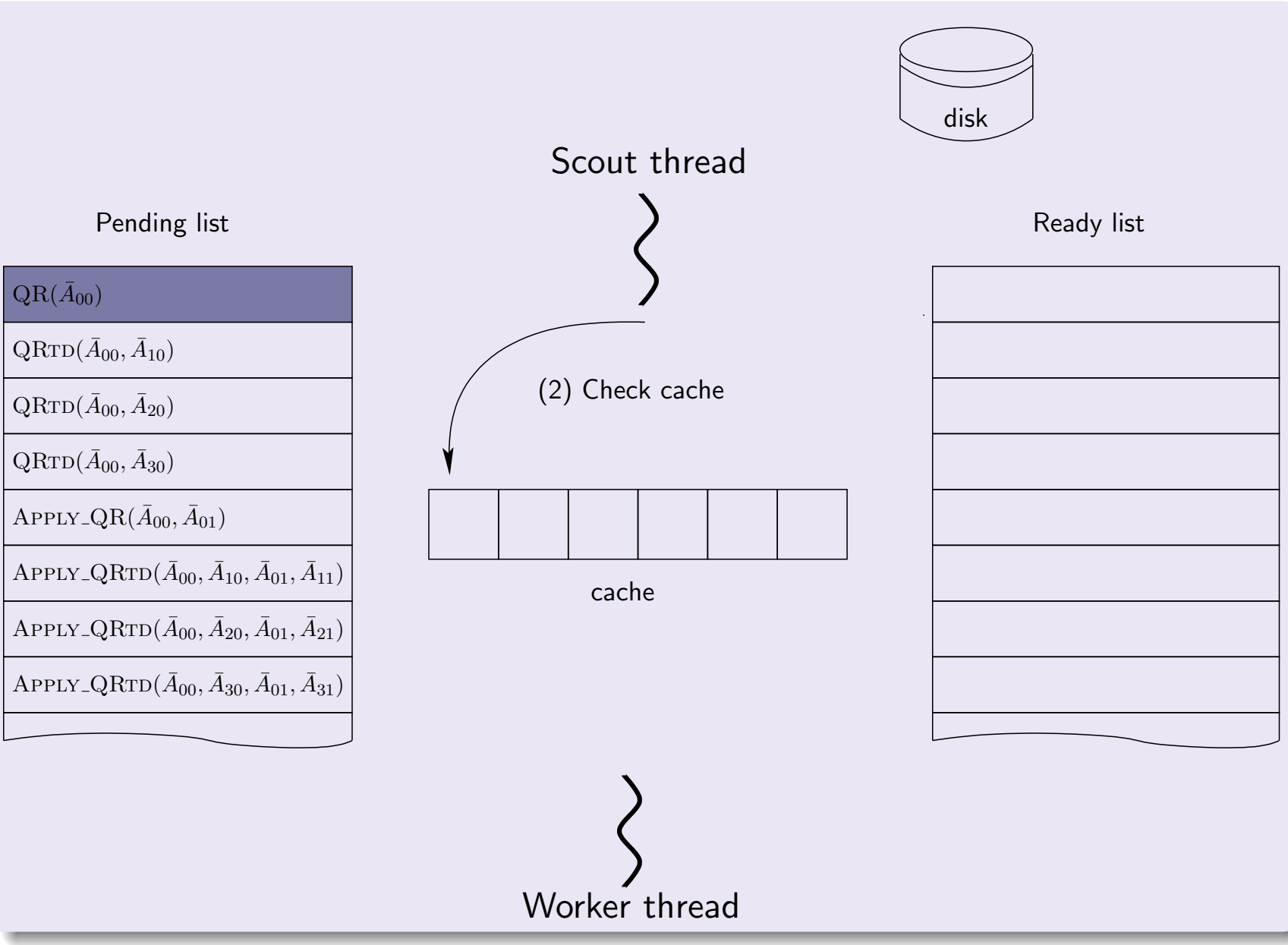
## Second stage: runtime dictates the execution

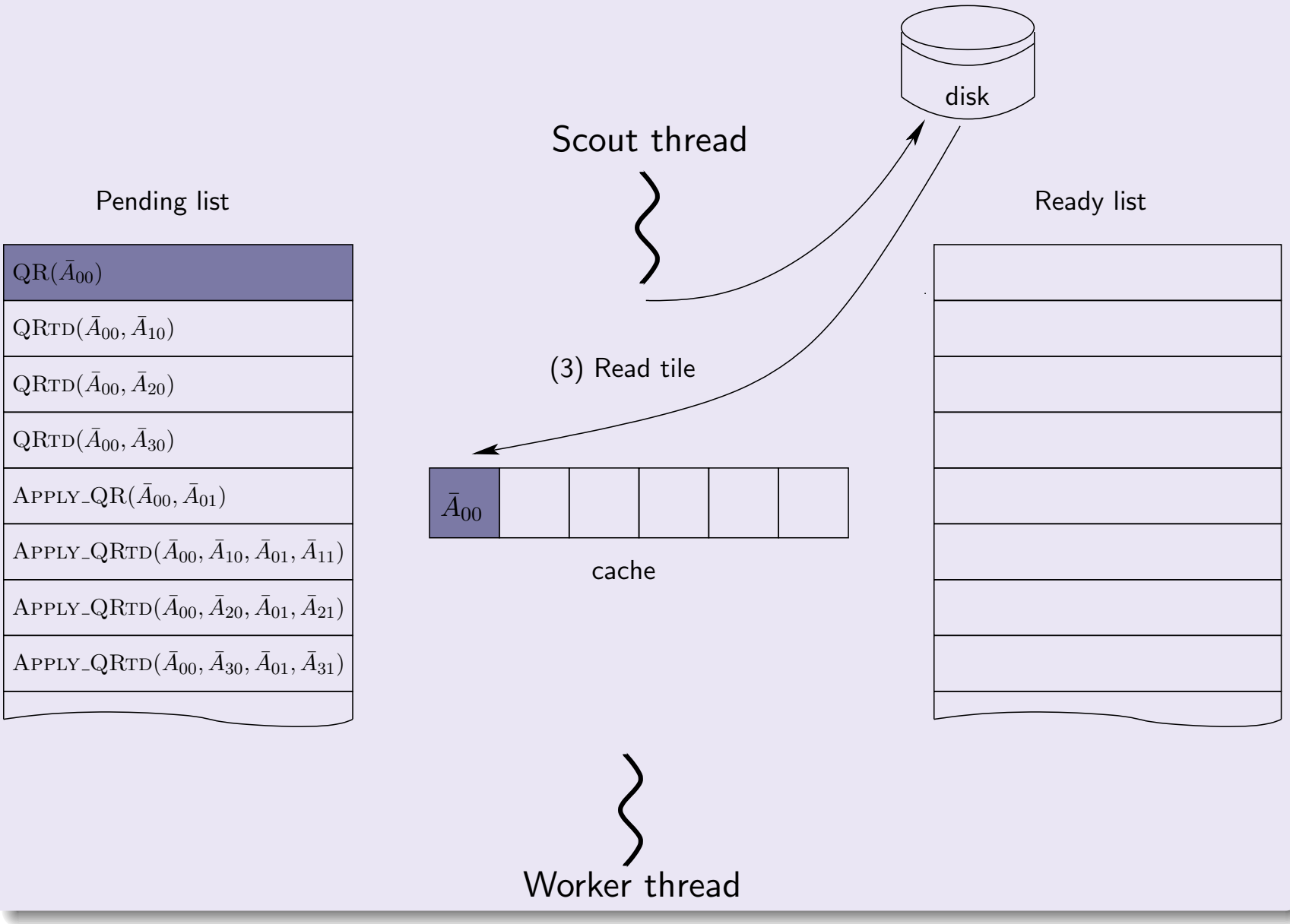
1. RAM is a software cache for data on disk
2. Scout and worker threads deliver transparent asynchronous I/O

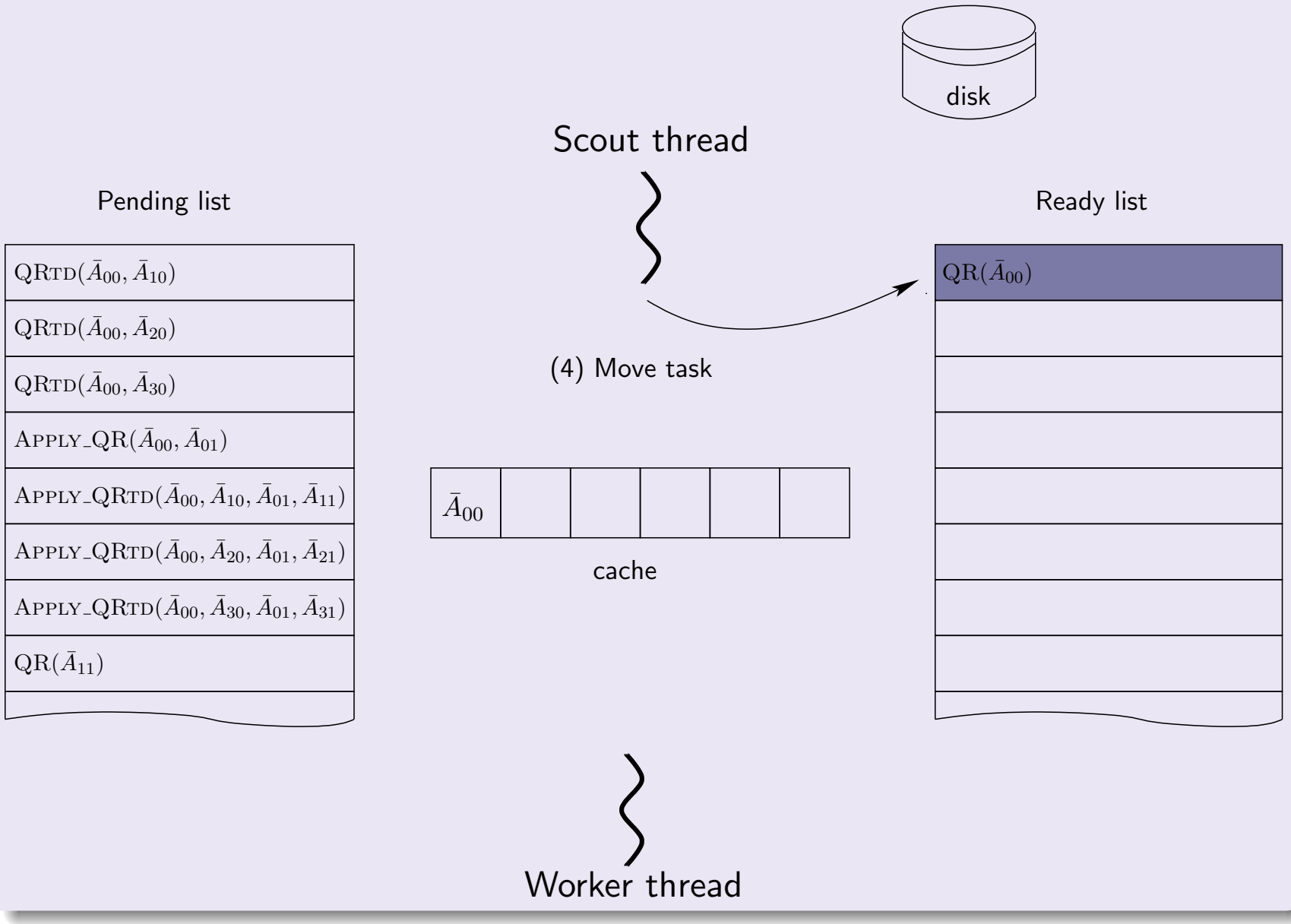
# OOC QR factorization of matrix $A$

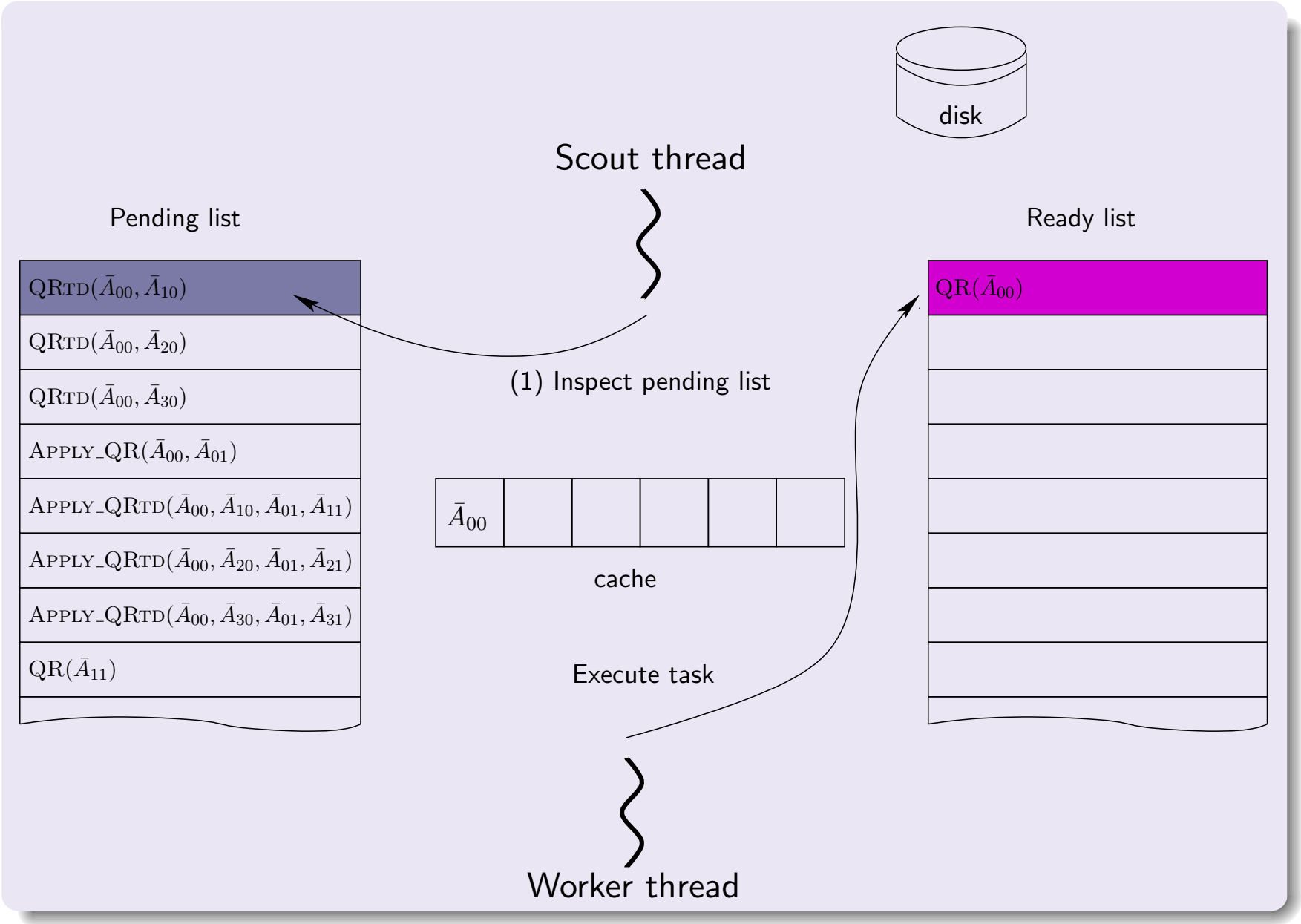
$$A = \begin{pmatrix} \bar{A}_{00} & \bar{A}_{01} & \bar{A}_{02} \\ \bar{A}_{10} & \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{20} & \bar{A}_{21} & \bar{A}_{22} \\ \bar{A}_{30} & \bar{A}_{31} & \bar{A}_{32} \end{pmatrix}$$

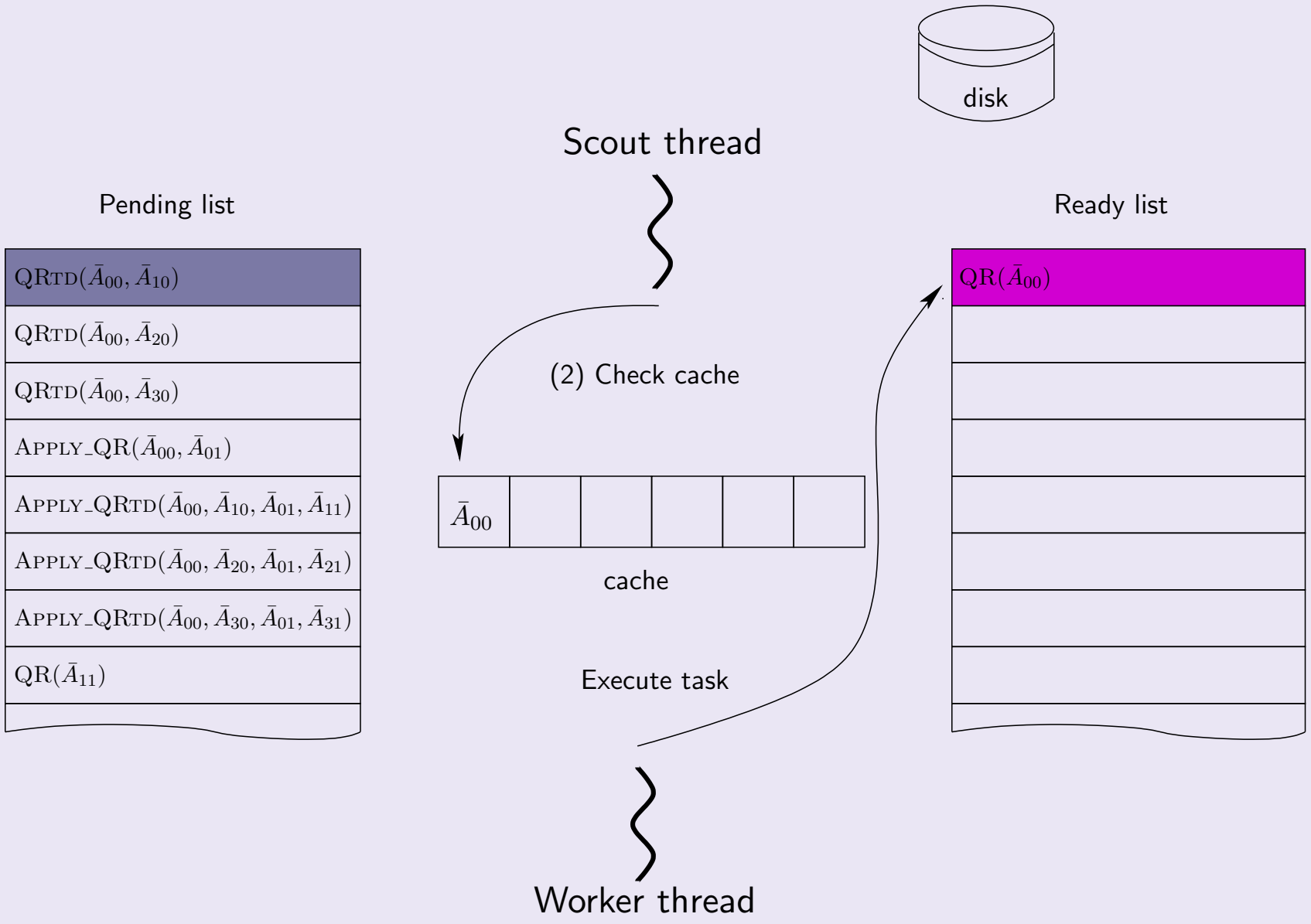

disk
Scout thread
Pending list
Ready list
$\mathrm{QR}(\bar{A}_{00})$
$\mathrm{QRTD}(\bar{A}_{00}, \bar{A}_{10})$
$\mathrm{QRTD}(\bar{A}_{00}, \bar{A}_{20})$
$\mathrm{QRTD}(\bar{A}_{00}, \bar{A}_{30})$
$\mathrm{APPLY\_QR}(\bar{A}_{00}, \bar{A}_{01})$
$\mathrm{APPLY\_QRTD}(\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11})$
$\mathrm{APPLY\_QRTD}(\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21})$
$\mathrm{APPLY\_QRTD}(\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31})$
(1) Inspect pending list
cache
Worker thread

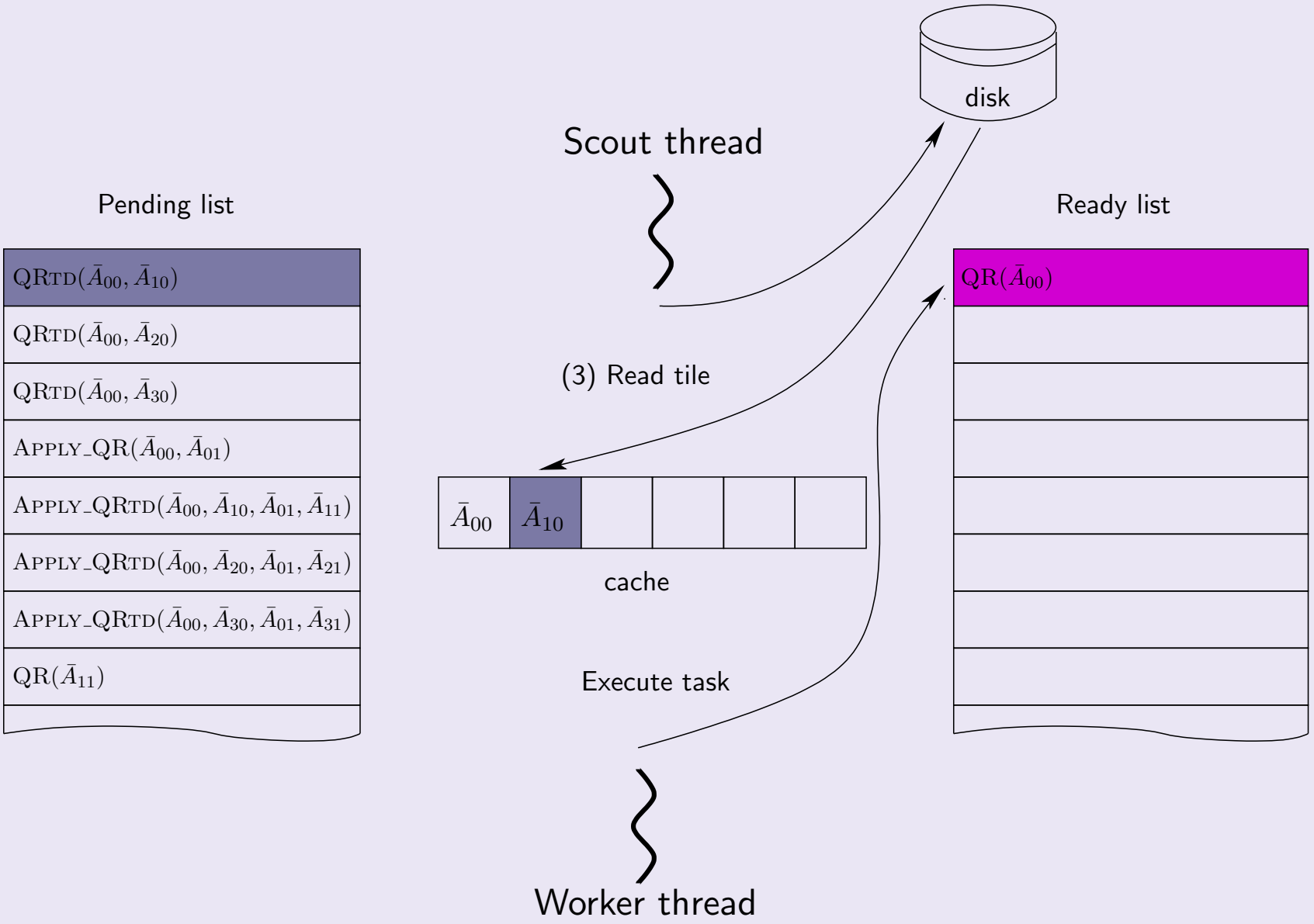

disk
Scout thread
Pending list
Ready list
$\mathrm{QR}(\bar{A}_{00})$
$\mathrm{QRTD}(\bar{A}_{00}, \bar{A}_{10})$
$\mathrm{QRTD}(\bar{A}_{00}, \bar{A}_{20})$
$\mathrm{QRTD}(\bar{A}_{00}, \bar{A}_{30})$
$\mathrm{APPLY\_QR}(\bar{A}_{00}, \bar{A}_{01})$
$\mathrm{APPLY\_QRTD}(\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11})$
$\mathrm{APPLY\_QRTD}(\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21})$
$\mathrm{APPLY\_QRTD}(\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31})$
(2) Check cache
cache
Worker thread

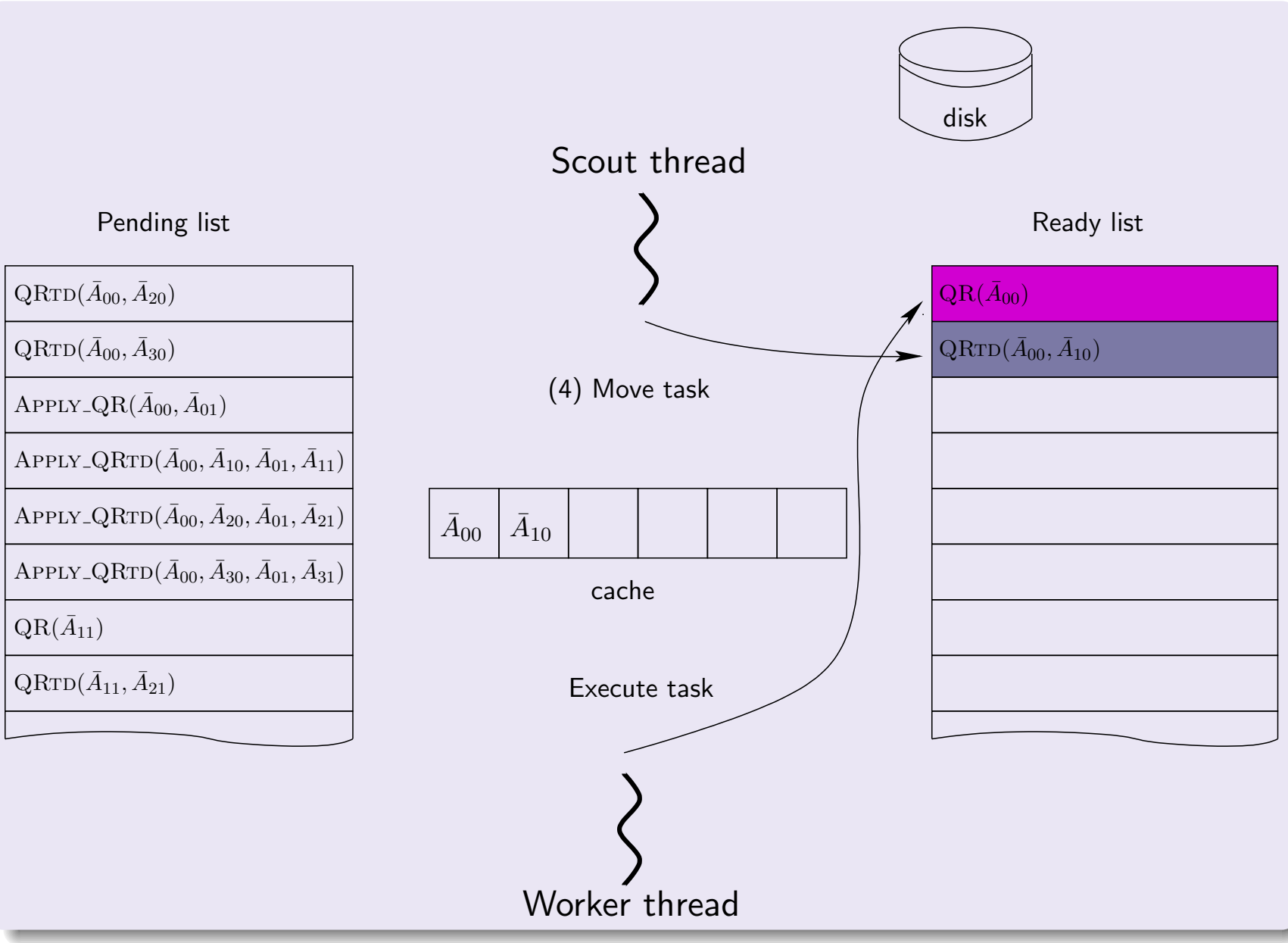

disk
Scout thread
Pending list
Ready list
$\mathrm{QR}(\bar{A}_{00})$
$\mathrm{QRTD}(\bar{A}_{00}, \bar{A}_{10})$
$\mathrm{QRTD}(\bar{A}_{00}, \bar{A}_{20})$
$\mathrm{QRTD}(\bar{A}_{00}, \bar{A}_{30})$
$\mathrm{APPLY\_QR}(\bar{A}_{00}, \bar{A}_{01})$
$\mathrm{APPLY\_QRTD}(\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11})$
$\mathrm{APPLY\_QRTD}(\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21})$
$\mathrm{APPLY\_QRTD}(\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31})$
(3) Read tile
$\bar{A}_{00}$
cache
Worker thread

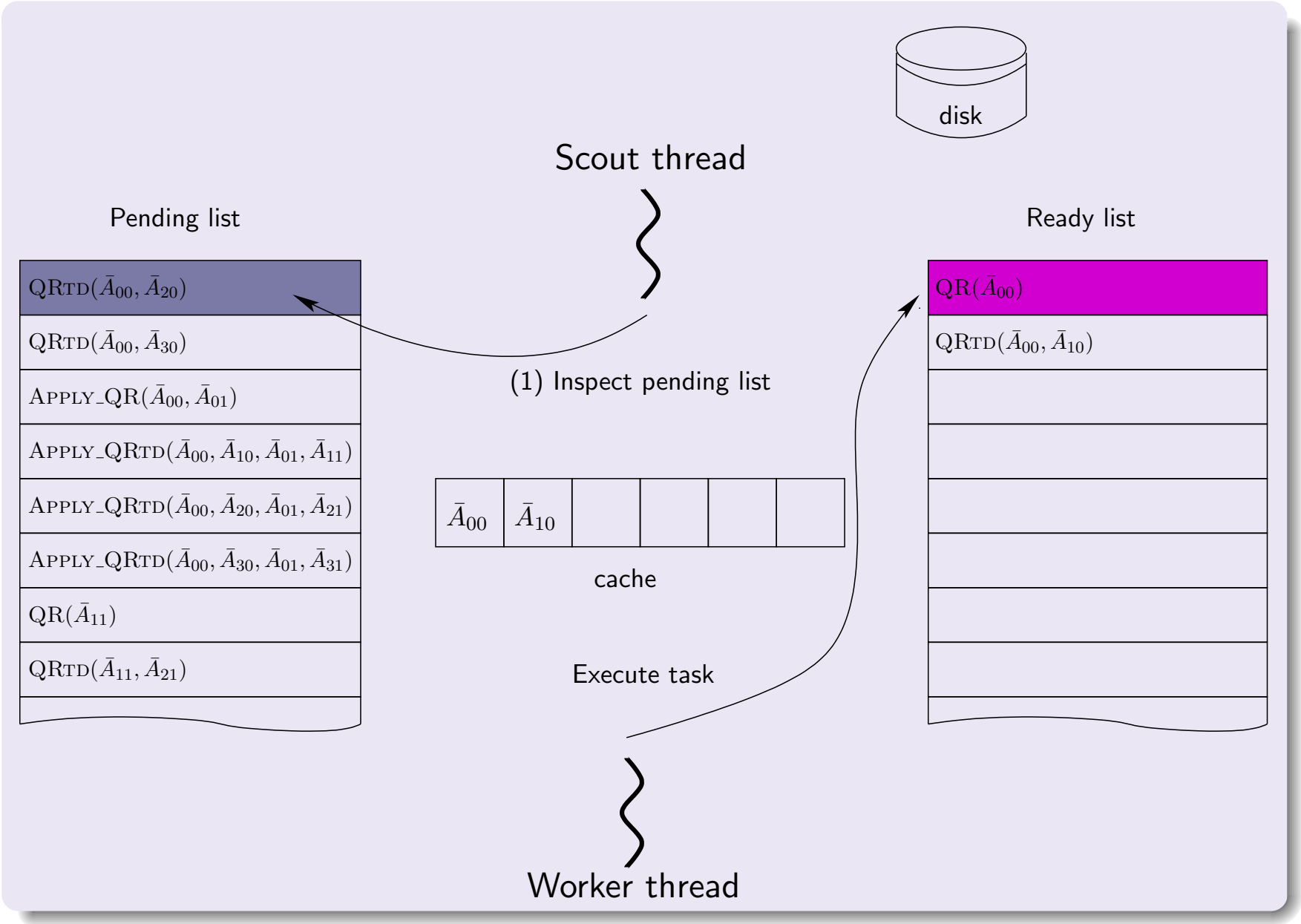

disk
Scout thread
Pending list
Ready list
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QR($\bar{A}_{00}$)
(4) Move task
$\bar{A}_{00}$
cache
Worker thread

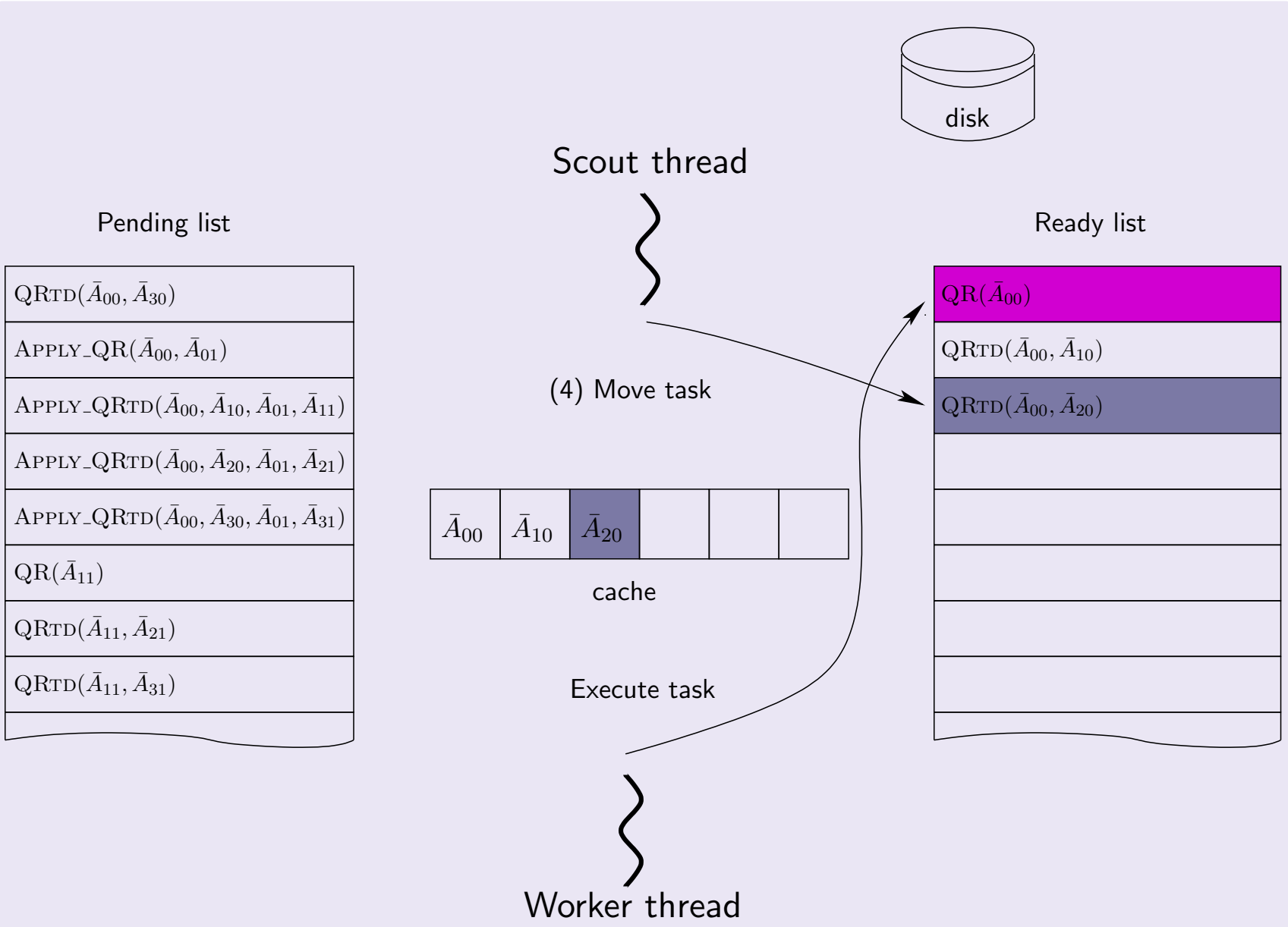

disk
Scout thread
Pending list
Ready list
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QR($\bar{A}_{00}$)
(1) Inspect pending list
$\bar{A}_{00}$
cache
Execute task
Worker thread

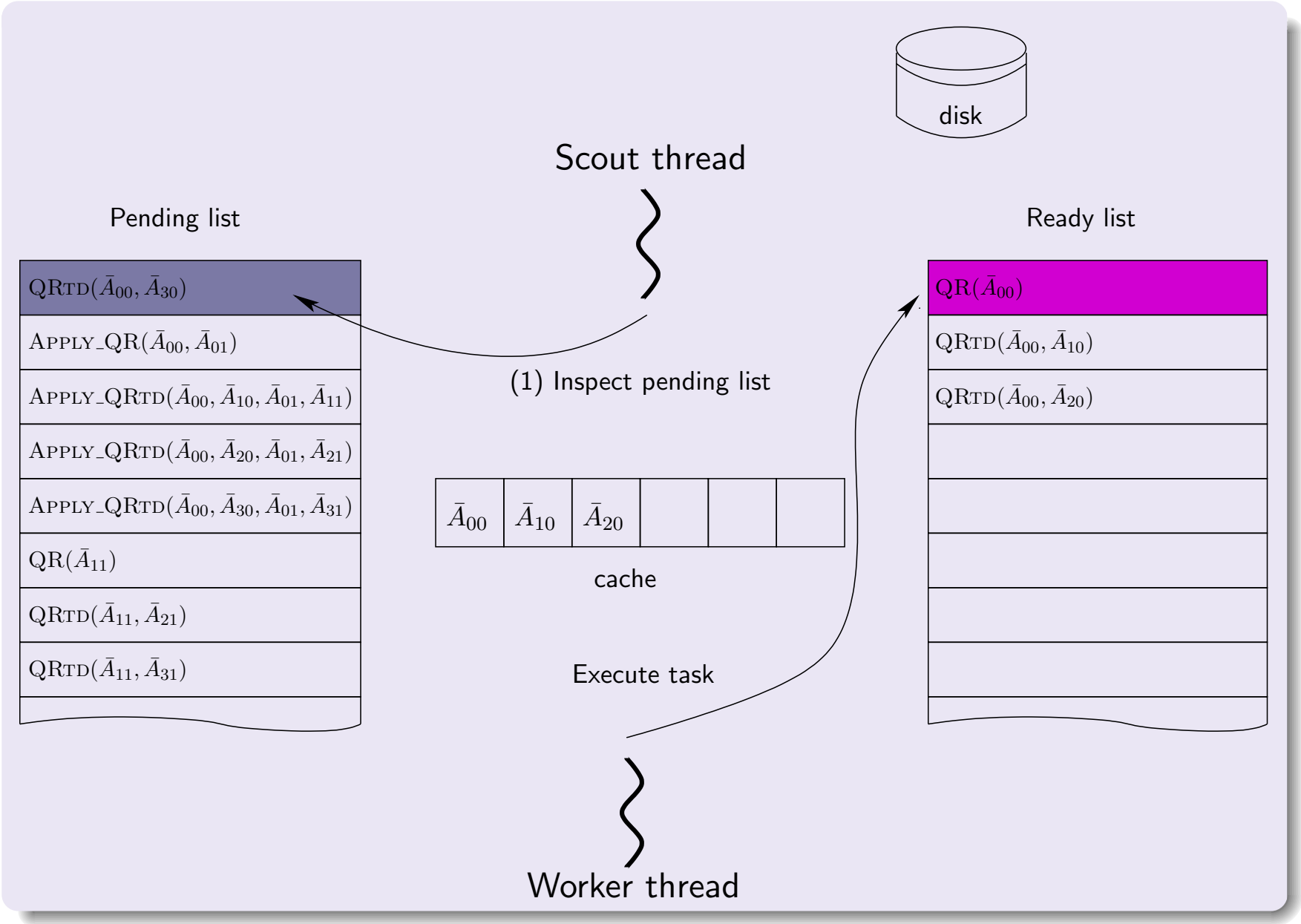

disk
Scout thread
Pending list
Ready list
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QR($\bar{A}_{00}$)
(2) Check cache
$\bar{A}_{00}$
cache
Execute task
Worker thread

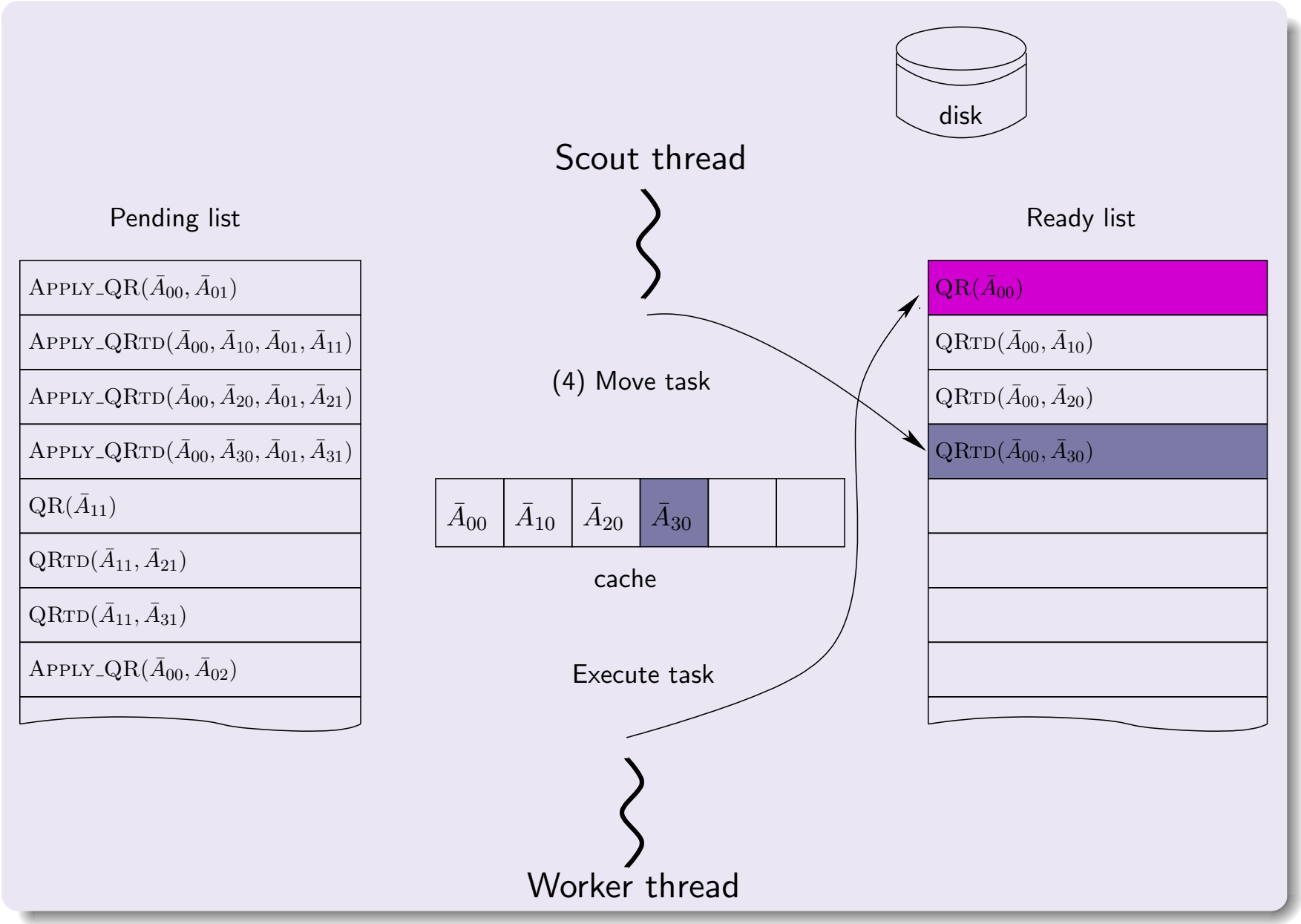

disk
Scout thread
Pending list
Ready list
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QR($\bar{A}_{00}$)
(3) Read tile
$\bar{A}_{00}$
$\bar{A}_{10}$
cache
Execute task
Worker thread

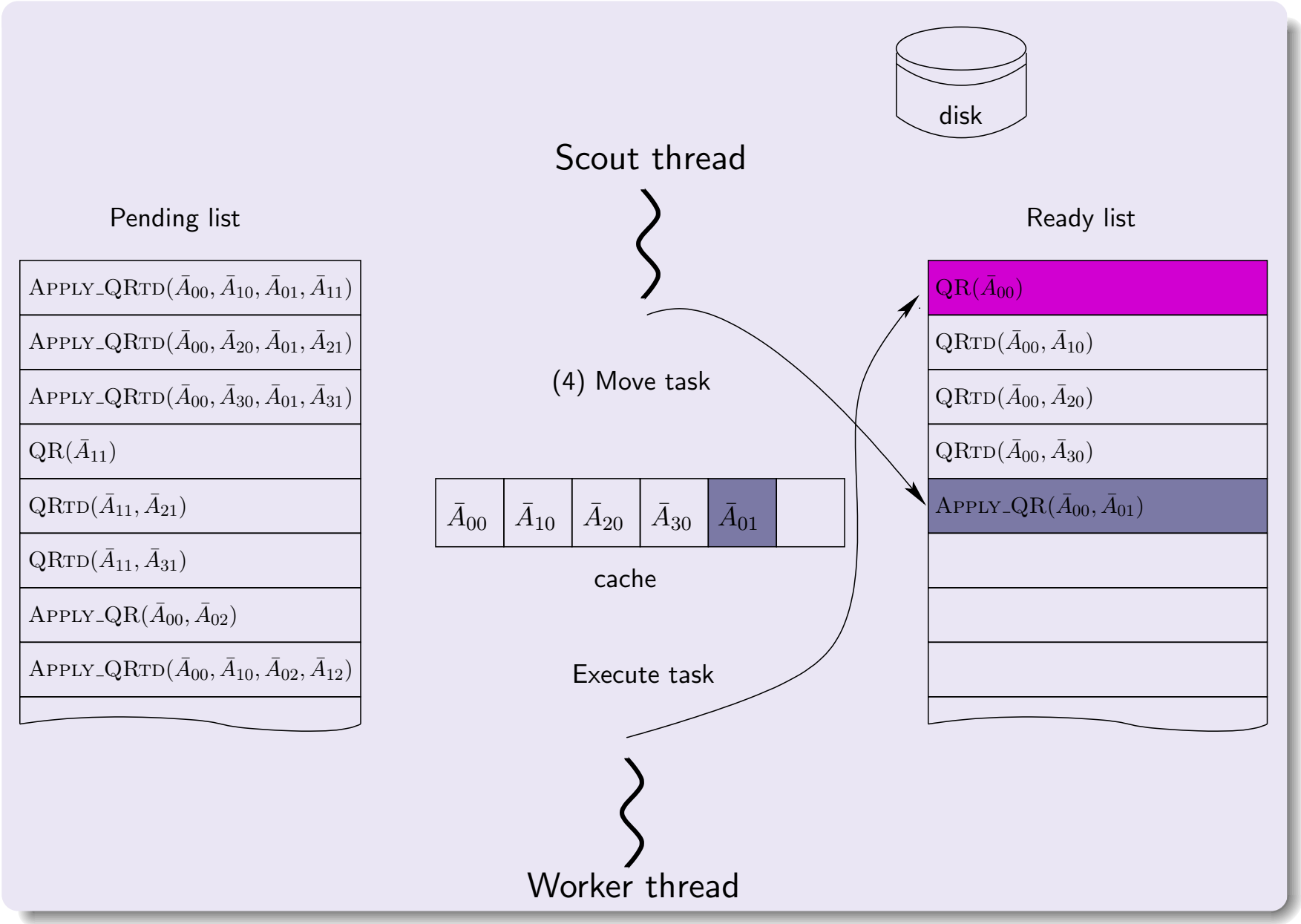

disk
Scout thread
Pending list
Ready list
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QR($\bar{A}_{00}$)
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
(4) Move task
$\bar{A}_{00}$
$\bar{A}_{10}$
cache
Execute task
Worker thread

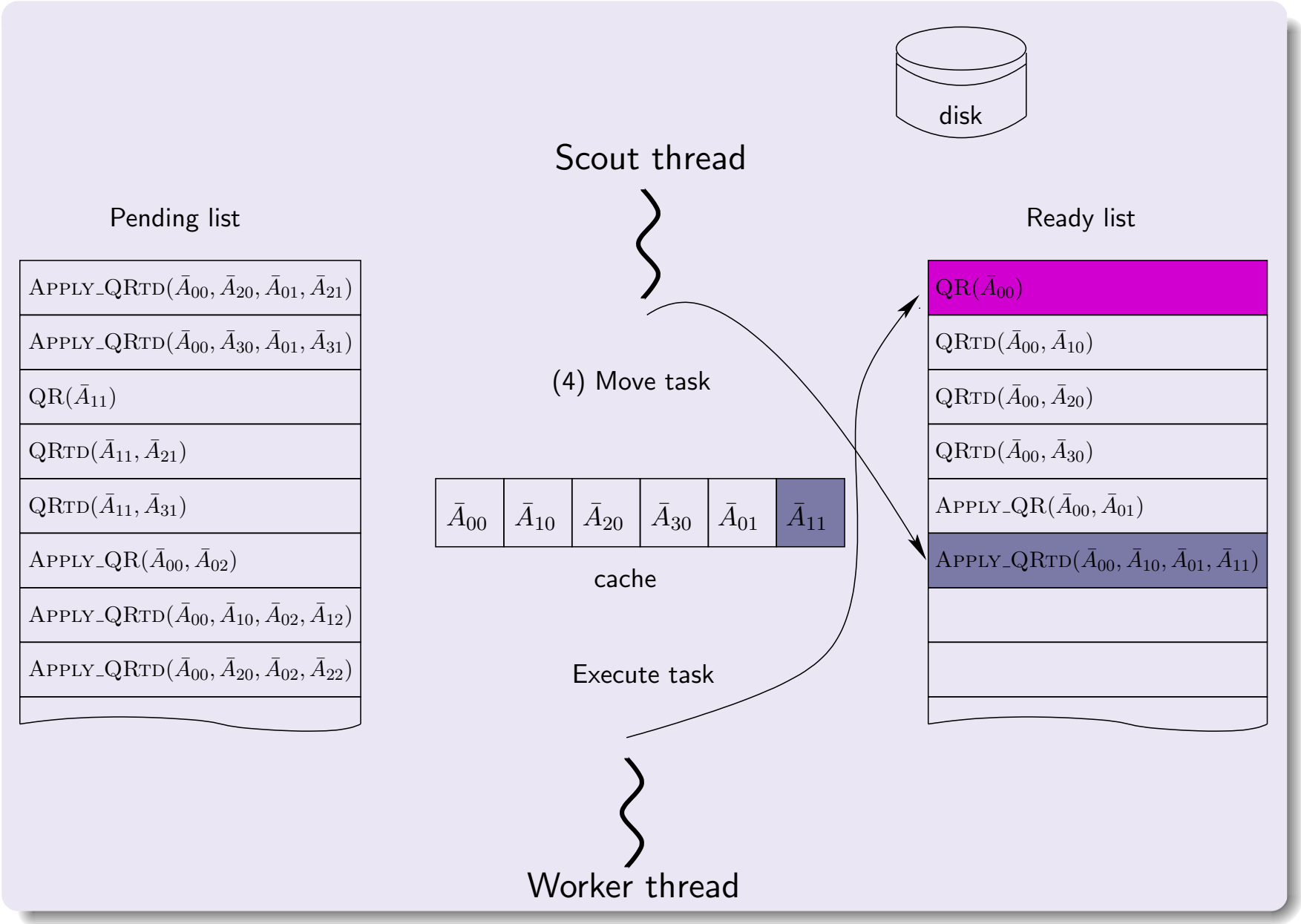

disk
Scout thread
Pending list
Ready list
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QR($\bar{A}_{00}$)
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
(1) Inspect pending list
$\bar{A}_{00}$
$\bar{A}_{10}$
cache
Execute task
Worker thread

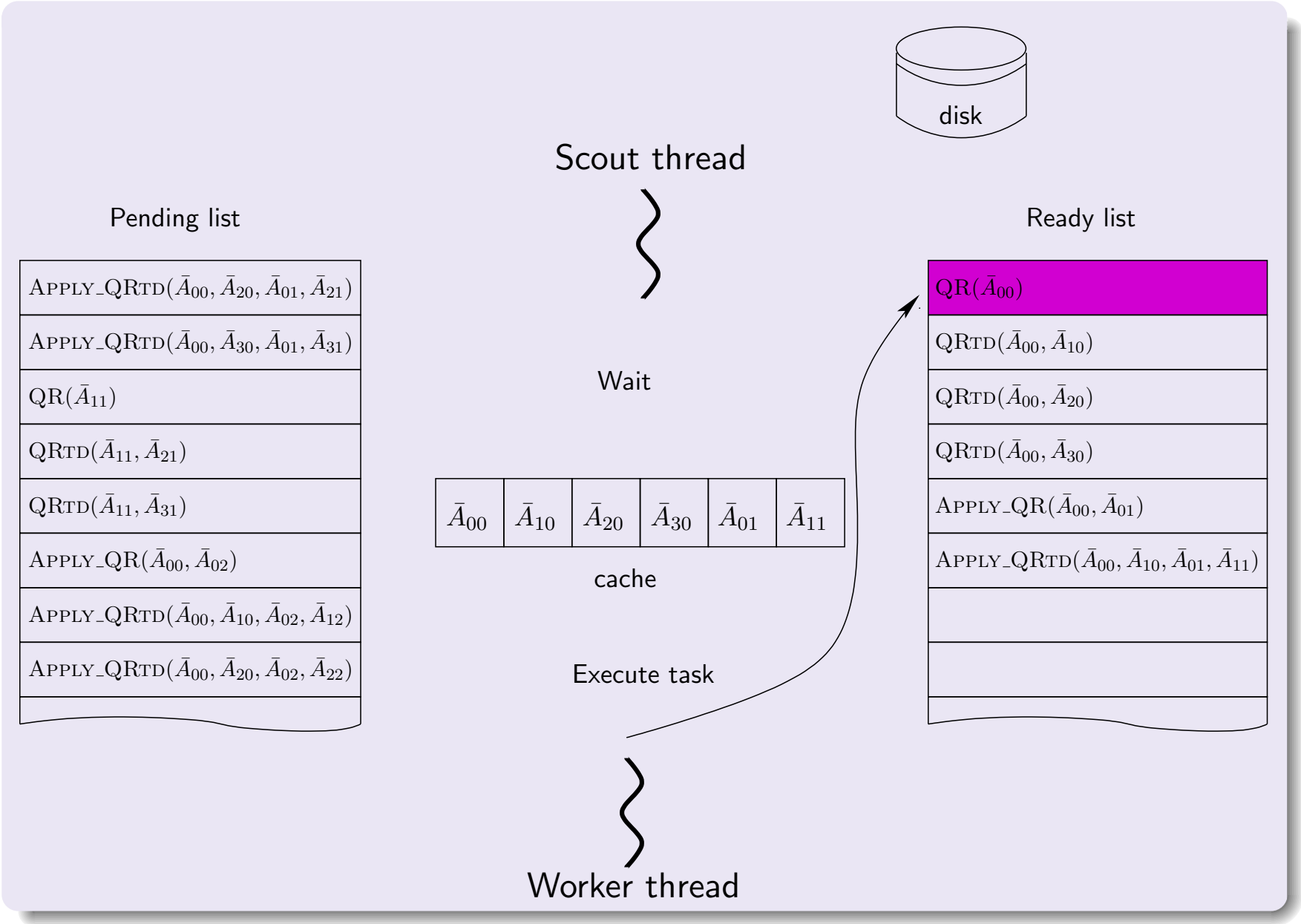

disk
Scout thread
Pending list
Ready list
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QRTD($\bar{A}_{11}, \bar{A}_{31}$)
QR($\bar{A}_{00}$)
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
(4) Move task
$\bar{A}_{00}$
$\bar{A}_{10}$
$\bar{A}_{20}$
cache
Execute task
Worker thread

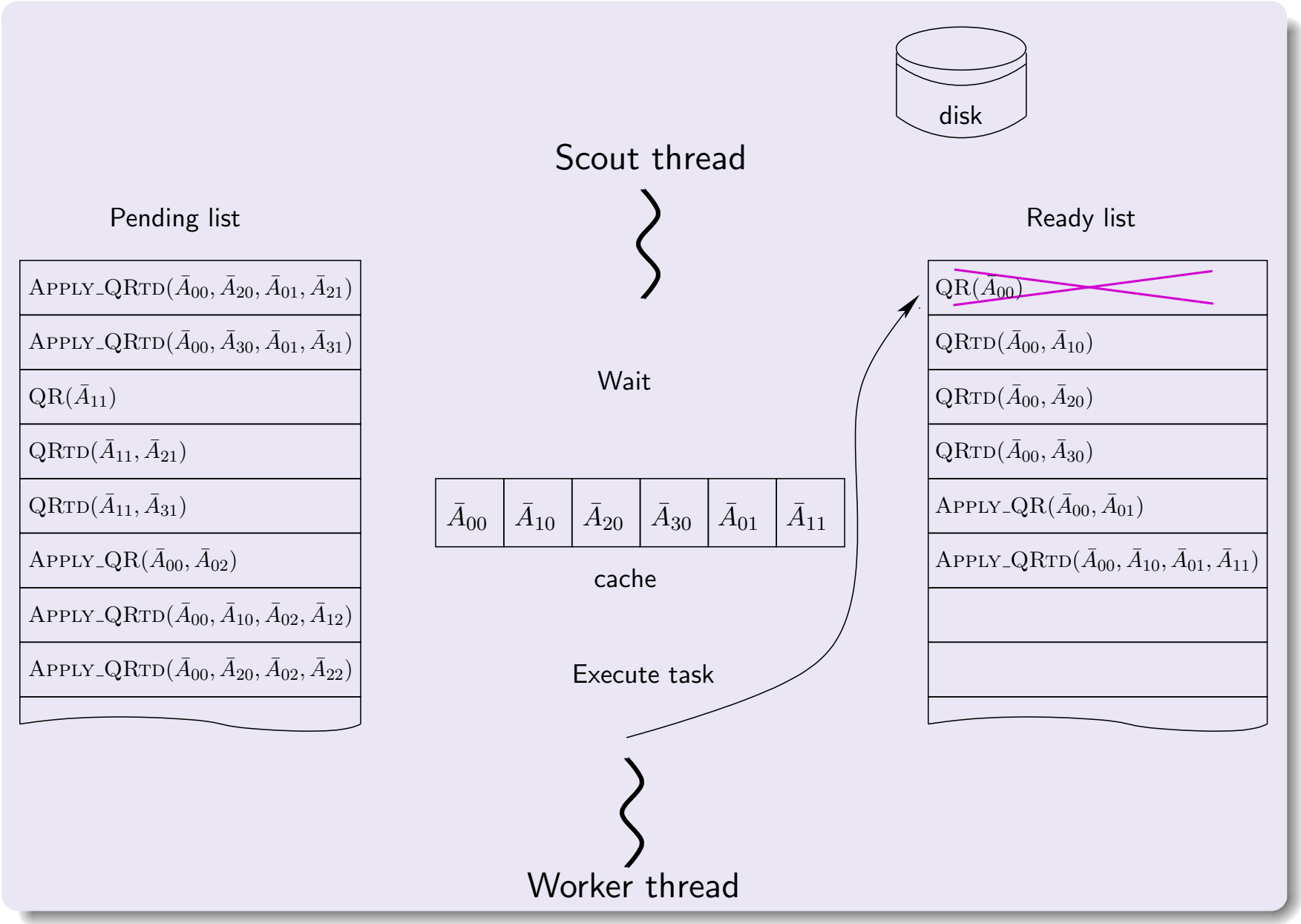

disk
Scout thread
Pending list
Ready list
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QRTD($\bar{A}_{11}, \bar{A}_{31}$)
(1) Inspect pending list
$\bar{A}_{00}$ $\bar{A}_{10}$ $\bar{A}_{20}$
cache
QR($\bar{A}_{00}$)
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
Execute task
Worker thread

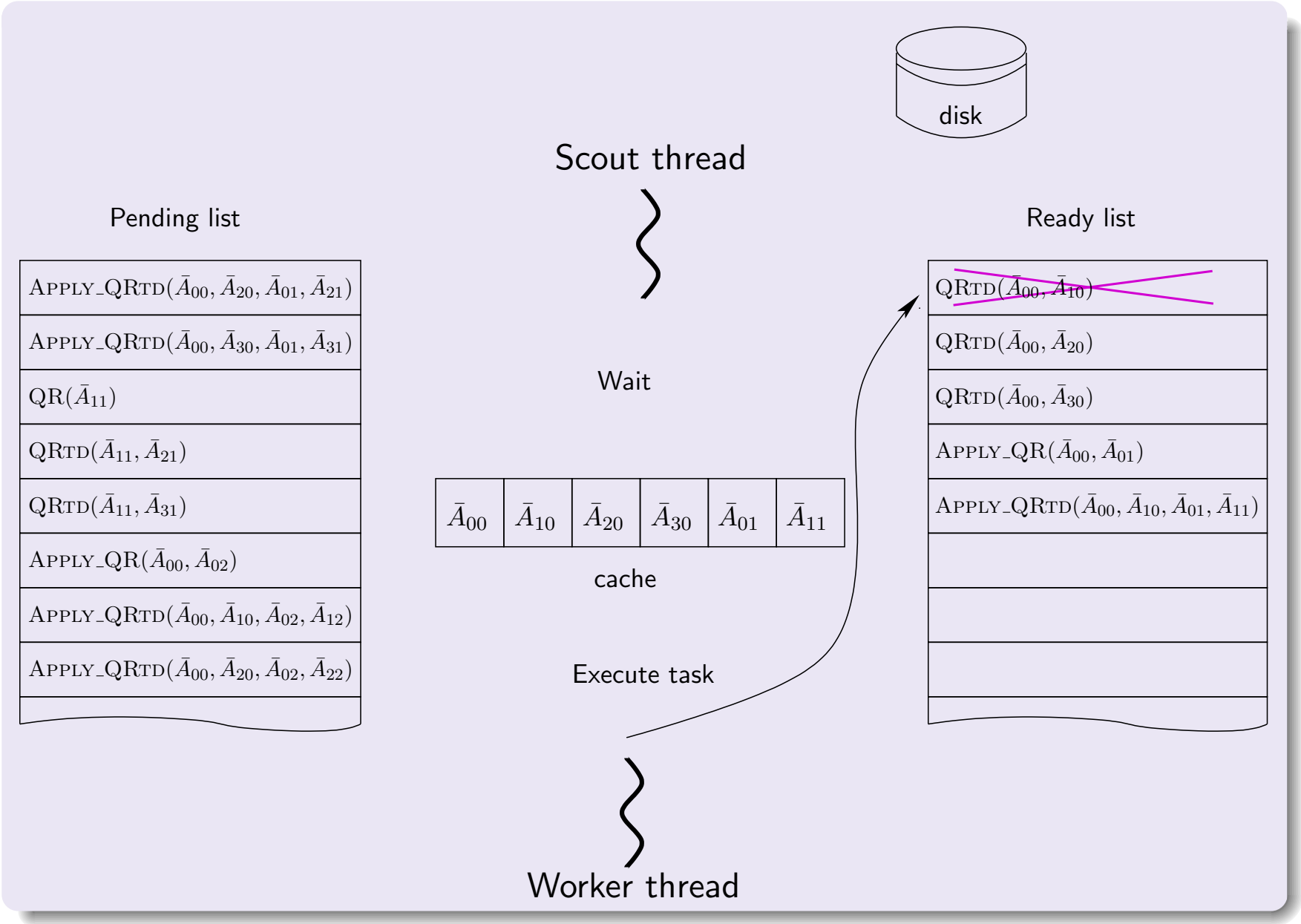

disk
Scout thread
Pending list
Ready list
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QRTD($\bar{A}_{11}, \bar{A}_{31}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{02}$)
QR($\bar{A}_{00}$)
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
(4) Move task
$\bar{A}_{00}$ $\bar{A}_{10}$ $\bar{A}_{20}$ $\bar{A}_{30}$
cache
Execute task
Worker thread

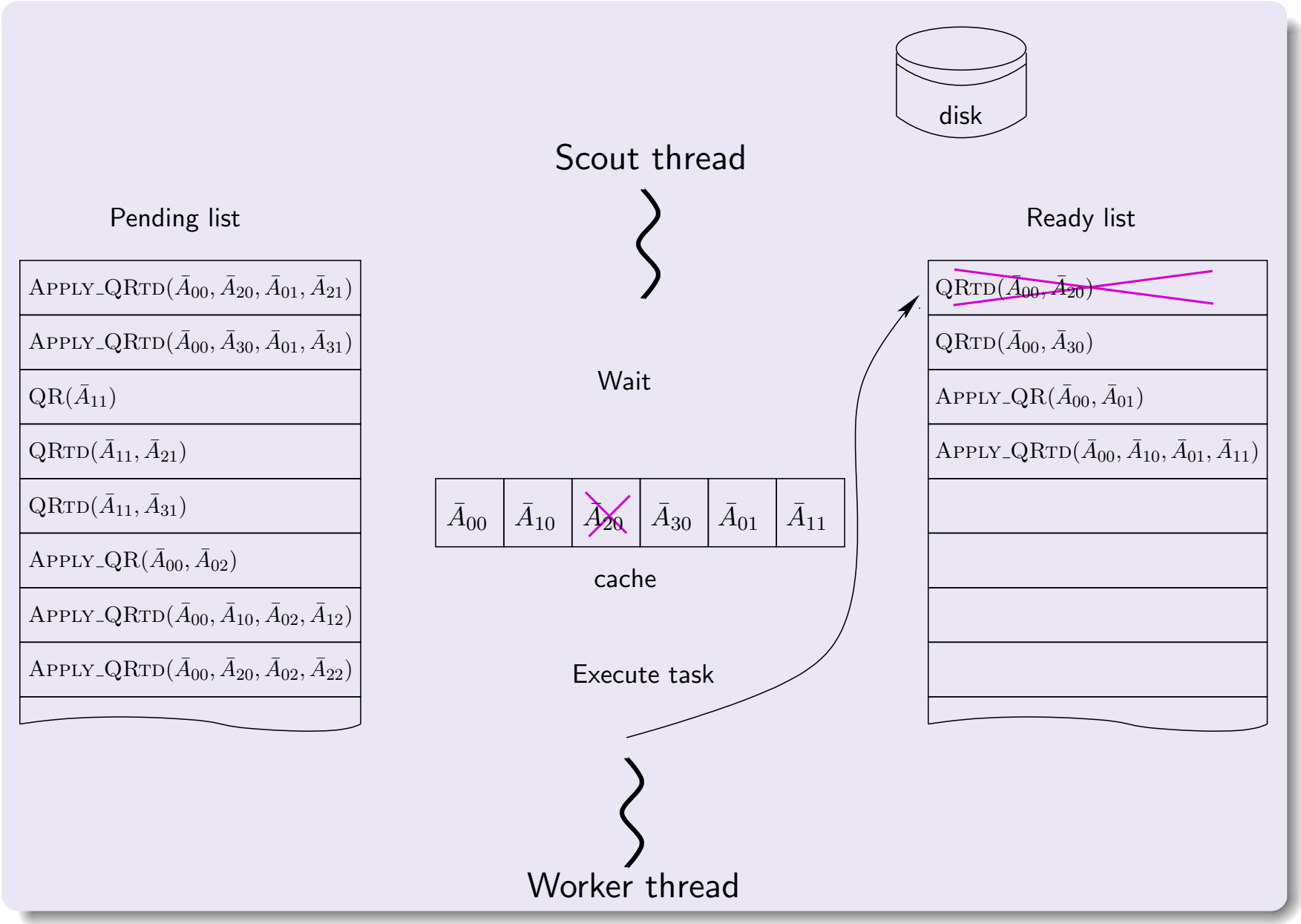

disk
Scout thread
Pending list
Ready list
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QRTD($\bar{A}_{11}, \bar{A}_{31}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{02}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{02}, \bar{A}_{12}$)
QR($\bar{A}_{00}$)
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
(4) Move task
$\bar{A}_{00}$ $\bar{A}_{10}$ $\bar{A}_{20}$ $\bar{A}_{30}$ $\bar{A}_{01}$
cache
Execute task
Worker thread

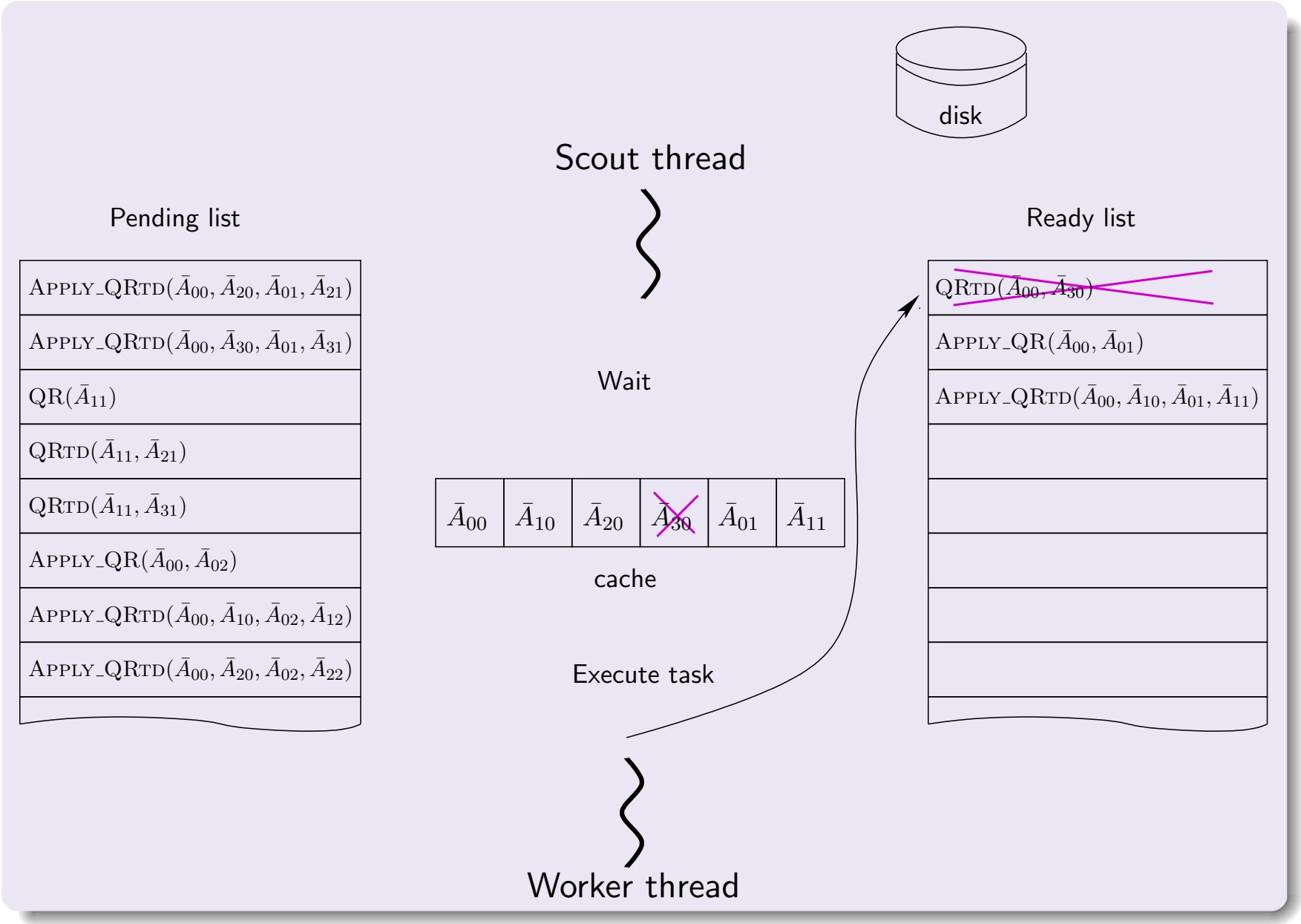

disk
Scout thread
Pending list
Ready list
$\textsc{Apply\_QRTD}(\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21})$
$\textsc{Apply\_QRTD}(\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31})$
$\mathrm{QR}(\bar{A}_{11})$
$\textsc{QRTD}(\bar{A}_{11}, \bar{A}_{21})$
$\textsc{QRTD}(\bar{A}_{11}, \bar{A}_{31})$
$\textsc{Apply\_QR}(\bar{A}_{00}, \bar{A}_{02})$
$\textsc{Apply\_QRTD}(\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{02}, \bar{A}_{12})$
$\textsc{Apply\_QRTD}(\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{02}, \bar{A}_{22})$
$\mathrm{QR}(\bar{A}_{00})$
$\textsc{QRTD}(\bar{A}_{00}, \bar{A}_{10})$
$\textsc{QRTD}(\bar{A}_{00}, \bar{A}_{20})$
$\textsc{QRTD}(\bar{A}_{00}, \bar{A}_{30})$
$\textsc{Apply\_QR}(\bar{A}_{00}, \bar{A}_{01})$
$\textsc{Apply\_QRTD}(\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11})$
(4) Move task
$\bar{A}_{00}$ $\bar{A}_{10}$ $\bar{A}_{20}$ $\bar{A}_{30}$ $\bar{A}_{01}$ $\bar{A}_{11}$
cache
Execute task
Worker thread

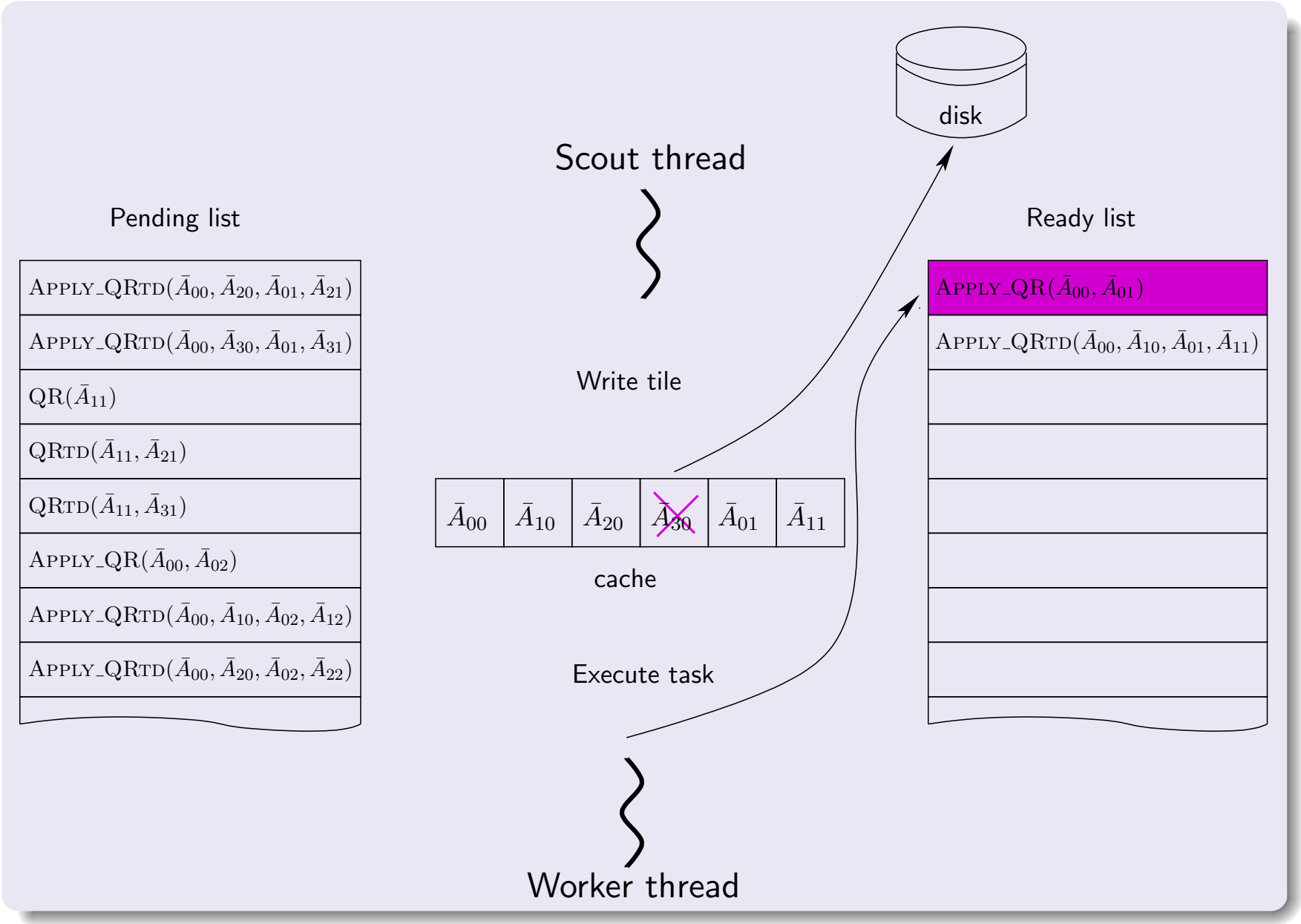

disk
Scout thread
Pending list
Ready list
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QRTD($\bar{A}_{11}, \bar{A}_{31}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{02}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{02}, \bar{A}_{12}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{02}, \bar{A}_{22}$)
QR($\bar{A}_{00}$)
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
Wait
$\bar{A}_{00}$ $\bar{A}_{10}$ $\bar{A}_{20}$ $\bar{A}_{30}$ $\bar{A}_{01}$ $\bar{A}_{11}$
cache
Execute task
Worker thread

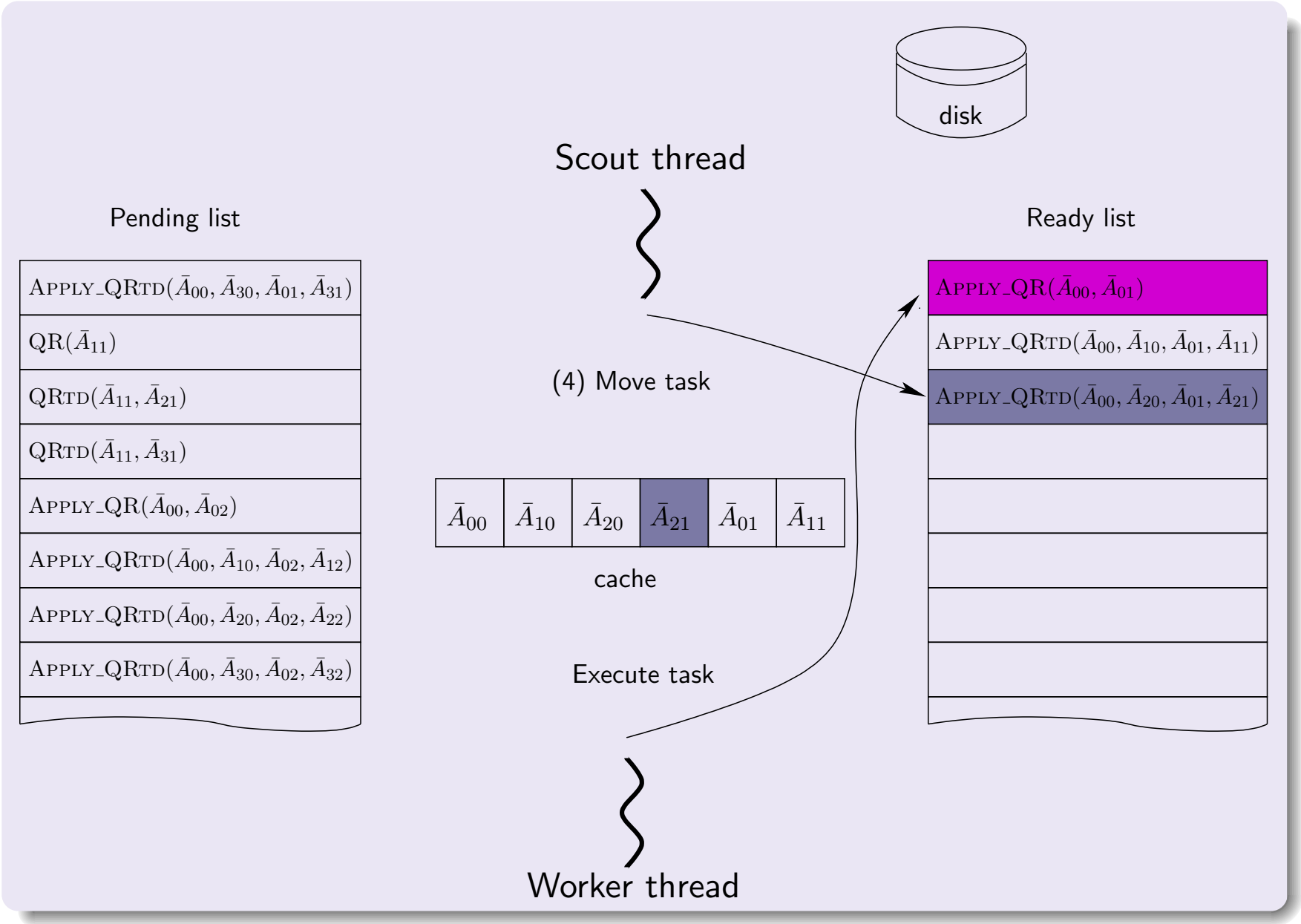

disk
Scout thread
Pending list
Ready list
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QRTD($\bar{A}_{11}, \bar{A}_{31}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{02}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{02}, \bar{A}_{12}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{02}, \bar{A}_{22}$)
QR($\bar{A}_{00}$)
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
Wait
$\bar{A}_{00}$ $\bar{A}_{10}$ $\bar{A}_{20}$ $\bar{A}_{30}$ $\bar{A}_{01}$ $\bar{A}_{11}$
cache
Execute task
Worker thread


disk
Scout thread
Pending list
Ready list
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QRTD($\bar{A}_{11}, \bar{A}_{31}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{02}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{02}, \bar{A}_{12}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{02}, \bar{A}_{22}$)
QRTD($\bar{A}_{00}, \bar{A}_{10}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
Wait
$\bar{A}_{00}$ $\bar{A}_{10}$ $\bar{A}_{20}$ $\bar{A}_{30}$ $\bar{A}_{01}$ $\bar{A}_{11}$
cache
Execute task
Worker thread


disk
Scout thread
Pending list
Ready list
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QRTD($\bar{A}_{11}, \bar{A}_{31}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{02}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{02}, \bar{A}_{12}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{02}, \bar{A}_{22}$)
QRTD($\bar{A}_{00}, \bar{A}_{20}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
Wait
$\bar{A}_{00}$ $\bar{A}_{10}$ $\bar{A}_{20}$ $\bar{A}_{30}$ $\bar{A}_{01}$ $\bar{A}_{11}$
cache
Execute task
Worker thread


disk
Scout thread
Pending list
Ready list
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QRTD($\bar{A}_{11}, \bar{A}_{31}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{02}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{02}, \bar{A}_{12}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{02}, \bar{A}_{22}$)
QRTD($\bar{A}_{00}, \bar{A}_{30}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
Wait
$\bar{A}_{00}$ $\bar{A}_{10}$ $\bar{A}_{20}$ $\bar{A}_{30}$ $\bar{A}_{01}$ $\bar{A}_{11}$
cache
Execute task
Worker thread


disk
Scout thread
Pending list
Ready list
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QRTD($\bar{A}_{11}, \bar{A}_{31}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{02}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{02}, \bar{A}_{12}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{02}, \bar{A}_{22}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
Write tile
$\bar{A}_{00}$ $\bar{A}_{10}$ $\bar{A}_{20}$ $\bar{A}_{30}$ $\bar{A}_{01}$ $\bar{A}_{11}$
cache
Execute task
Worker thread


disk
Scout thread
Pending list
Ready list
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{01}, \bar{A}_{31}$)
QR($\bar{A}_{11}$)
QRTD($\bar{A}_{11}, \bar{A}_{21}$)
QRTD($\bar{A}_{11}, \bar{A}_{31}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{02}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{02}, \bar{A}_{12}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{02}, \bar{A}_{22}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{30}, \bar{A}_{02}, \bar{A}_{32}$)
APPLY_QR($\bar{A}_{00}, \bar{A}_{01}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{10}, \bar{A}_{01}, \bar{A}_{11}$)
APPLY_QRTD($\bar{A}_{00}, \bar{A}_{20}, \bar{A}_{01}, \bar{A}_{21}$)
(4) Move task
$\bar{A}_{00}$ $\bar{A}_{10}$ $\bar{A}_{20}$ $\bar{A}_{21}$ $\bar{A}_{01}$ $\bar{A}_{11}$
cache
Execute task
Worker thread

# Outline

# Parallelization of In-Core Task

## Target: multi-core processor

$$\mathrm{QR}(\bar{A}_{k,k})$$
$$\textsc{Apply\_QR}(\bar{A}_{k,k}, \bar{A}_{k,k+j})$$

$\rightarrow$ Standard parallelization using multithreaded BLAS

$$\mathrm{QR}_{\mathrm{TD}}(\bar{A}_{k,k}, \bar{A}_{k+i,k})$$
$$\textsc{Apply\_QR}_{\mathrm{TD}}(\bar{A}_{k,k}, \bar{A}_{k,k+j}, \ldots)$$

$\rightarrow$ Block column-wise parallelization

# Outline

# Porting `libflame`. Experimental results

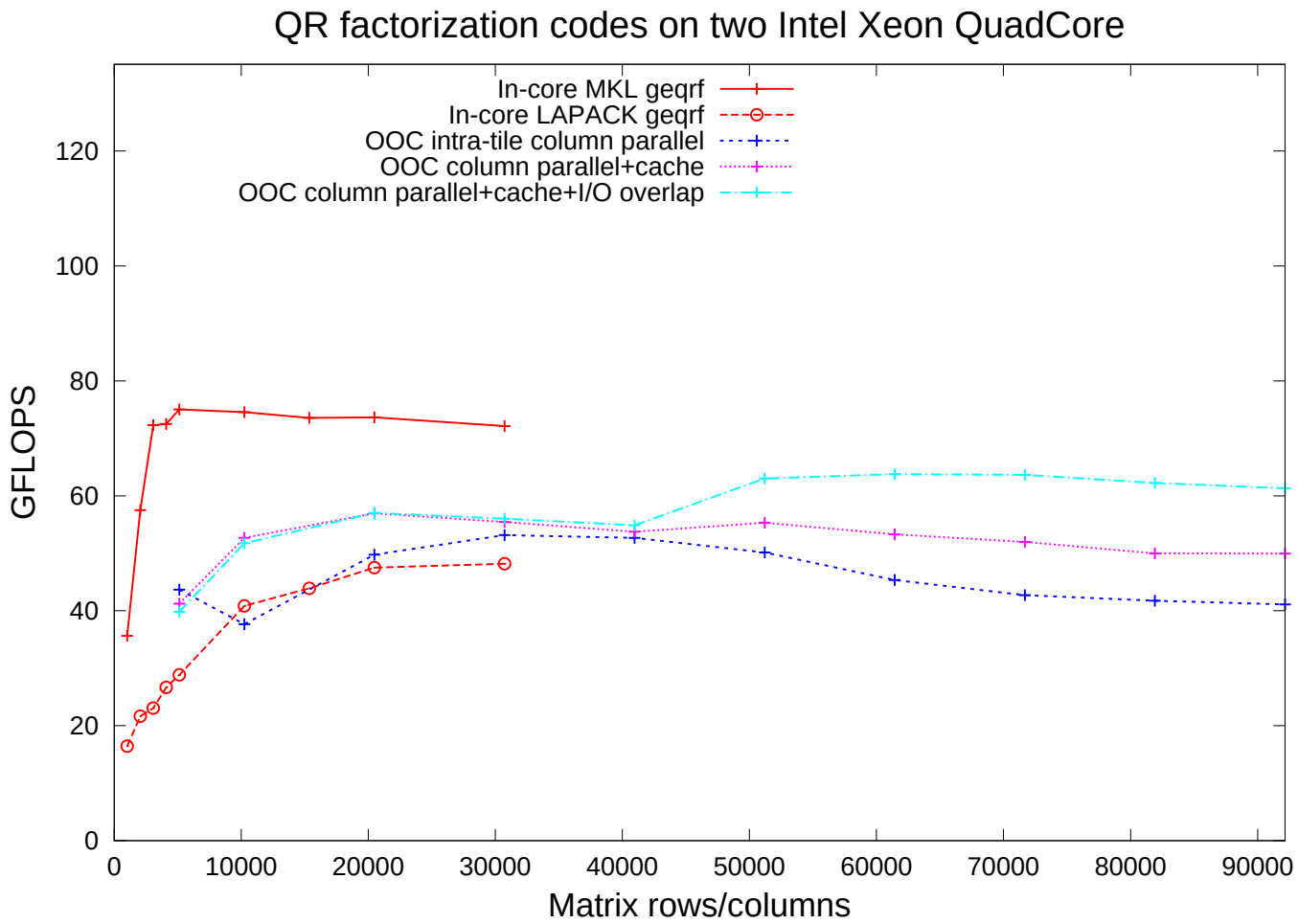
## Hardware & Software

- Two Intel Xeon QuadCore E5405, 2.0 GHz, 8 GBytes RAM
- I/O interface of 1.5 Gbites/sec
- SATA-I disk with 160 GBytes
- MKL 10.1, single precision

## Algorithms & Implementations

- In-core MKL
- OOC Traditional
- OOC Cache
- OOC Reordered + data-flow
- OOC Overlap I/O

# Porting libflame. Experimental results



# Porting `libflame`. Experimental results

| Matrix size $n \times n$ | Time (hours, minutes, seconds) | | | Memory needs (Gbytes) |
|---|---|---|---|---|
| 51,200 | | 46m | 44.5s | 9.76 |
| 61,440 | 1h | 21m | 20.6s | 14.06 |
| 71,680 | 2h | 8m | 2.4s | 19.14 |
| 81,920 | 3h | 16m | 2.0s | 25.00 |
| 92,160 | 4h | 48m | 48.2s | 31.64 |
| 102,400 | 6h | 19m | 32.1s | 39.06 |

# Conclusion

## Against conventional wisdom

For linear algebra problems...

- Disk is fast enough to feed the processor
- Programming OOC is transparent to the library developer
  - Job for the runtime
  - Transparent port of the functionality of `libflame`
- Solving large problems possible using a few multi-core processors

## For more information...

Visit `http://www.cs.utexas.edu/users/flame/`