Zero-Jitter Task Chains via Algebraic Rings

Enrico Bini, Paolo Pazzaglia, Martina Maggio
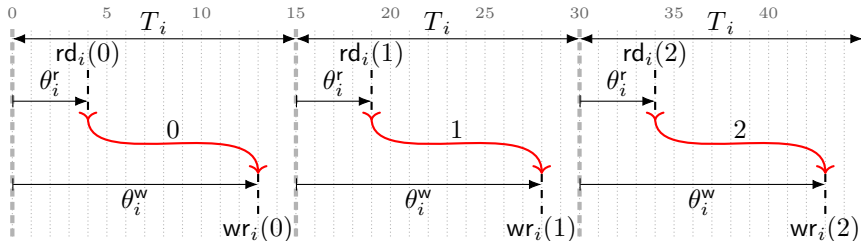
WaP @RTSS 2023, December 5th 2023
Taipei, Taiwan

Model of a LET task τ_i

- A LET *task* τ_i is composed by *periodic jobs* (curvy arrows)
  - in LET, we only care of the read and write instants

$$\theta_i^{\mathsf{r}} \qquad \qquad \textit{read phasing of } \tau_i \textit{, relative to the period } T_i$$

$$\theta_i^{\mathsf{w}} \qquad \qquad \textit{write phasing of } \tau_i \textit{, relative to the period } T_i$$

$$\mathsf{rd}_i(j) = j\,T_i + \theta_i^{\mathsf{r}} \qquad \textit{read instant of job } j \textit{ of } \tau_i$$

$$\mathsf{wr}_i(j) = j\,T_i + \theta_i^{\mathsf{w}} \qquad \textit{write instant of job } j \textit{ of } \tau_i$$
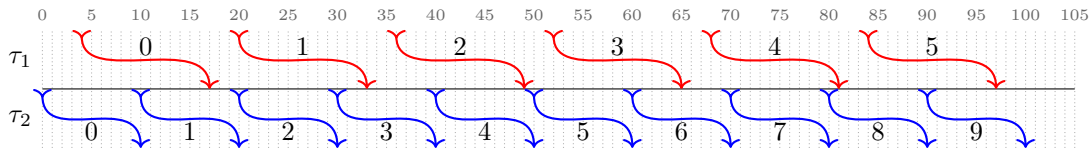
Chain of 2 LET tasks
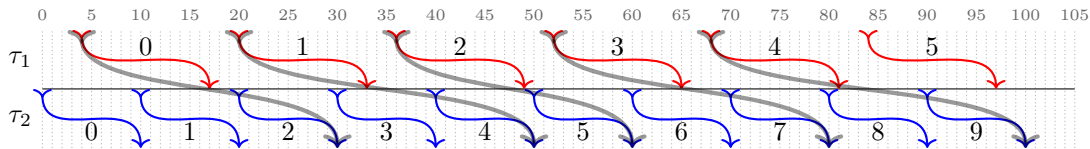

- Chain of LET tasks
  - each task reads data written
    by the previous one
  - data is on **shared memory**
    - τ_1 may over-write data
      before τ_2 reads (if $T_1 < T_2$)
   - τ_2 may read again the same
      data (if $T_2 < T_1$)

- Chain of LET tasks
  - each task reads data written by the previous one
  - data is on **shared memory**
    - $\star$ τ_1 may over-write data before τ_2 reads (if $T_1 < T_2$)
    - $\star$ τ_2 may read again the same data (if $T_2 < T_1$)
- Chains of jobs have **variable** read-to-write delay

(j_1, j_2)	$\mathsf{rd}_1(j_1)$	$\mathsf{wr}_2(j_2)$	$\mathsf{wr}_2(j_2) - \mathsf{rd}_1(j_1)$
$(0, 2)$	4	30	26
$(1, 4)$	20	50	30
$(2, 5)$	36	60	24
$(3, 7)$	52	80	28
$(4, 9)$	68	100	32
$\ldots$	$\ldots$	$\ldots$	$\ldots$

Variability of the input-output delay

- The source of variability is $\mathrm{rd}_2(j_2) - \mathrm{wr}_1(j_1)$. In the example: $3, 7, 1, 5, 9, 3, 7, \ldots$
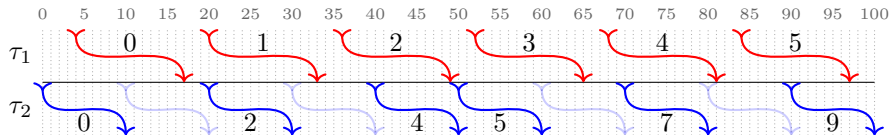
Variability of the input-output delay

- $\mathrm{rd}_2(j_2) - \mathrm{wr}_1(j_1) < T_2$, always. Let's zoom in …
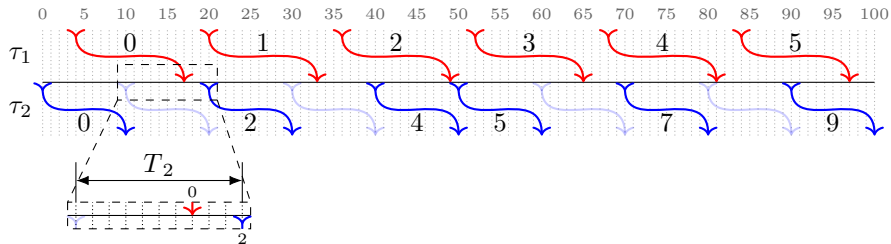
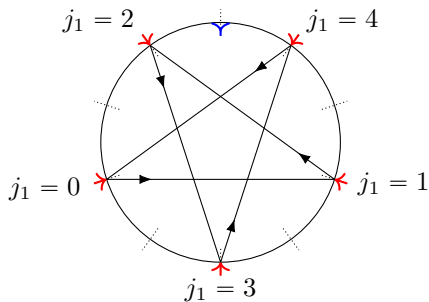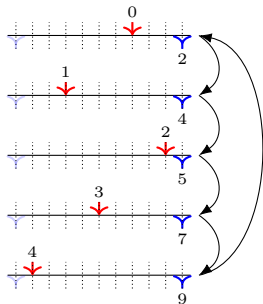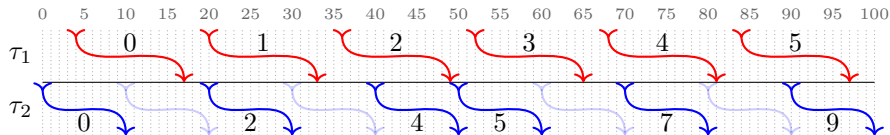- ...and align the read instants of consecutive τ_2 jobs.
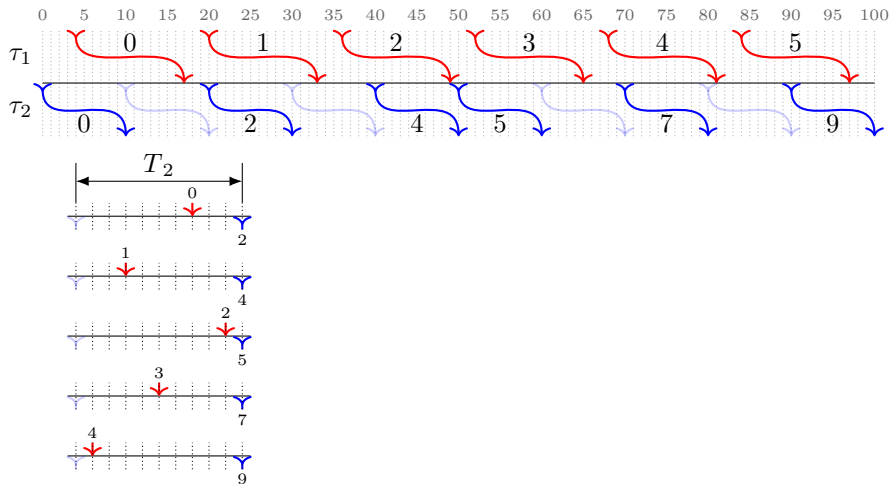
- A repetitive requence... mmhhh... better to visualize over the (algebraic) ring $\mathbb{Z}/T_2\mathbb{Z}$.
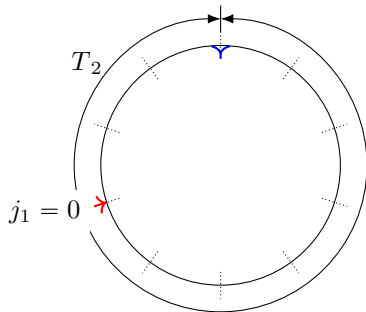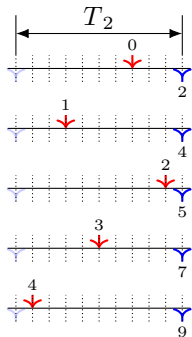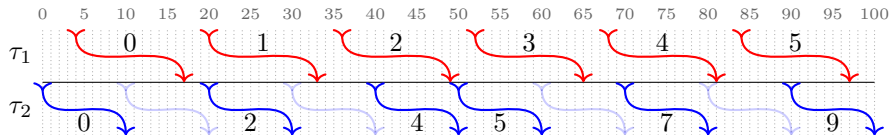
Variability of the input-output delay

- By inverting over the ring $\mathbb{Z}/T_2\mathbb{Z}$, max/min delay is found **without** unrolling the schedule.

Variability of the input-output delay

- Exploited to **eliminate the jitter** of a 2-tasks chain.