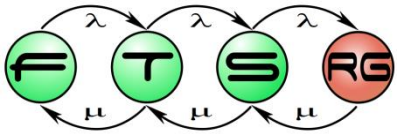
Fault Tolerant Allocation and Schedule

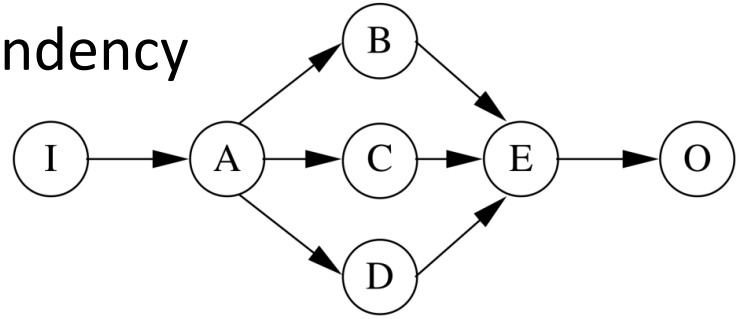
Budapesti Műszaki és Gazdaságtudományi Egyetem
Hibatűrő Rendszerek Kutatócsoport



Assumptions

- Given a set of hardware components (p^1 … p^N)
  - Multiple software can be allocated to P
  - Only one software can run at the same time
  - If a hardware fails, it cannot operate anymore.

- Given a set of software components (s^1 … s^M)
  - Processes can be described by *data-flow graphs*
    - Each vertex is a software component
    - Each directed edge is a data-dependency between software components.



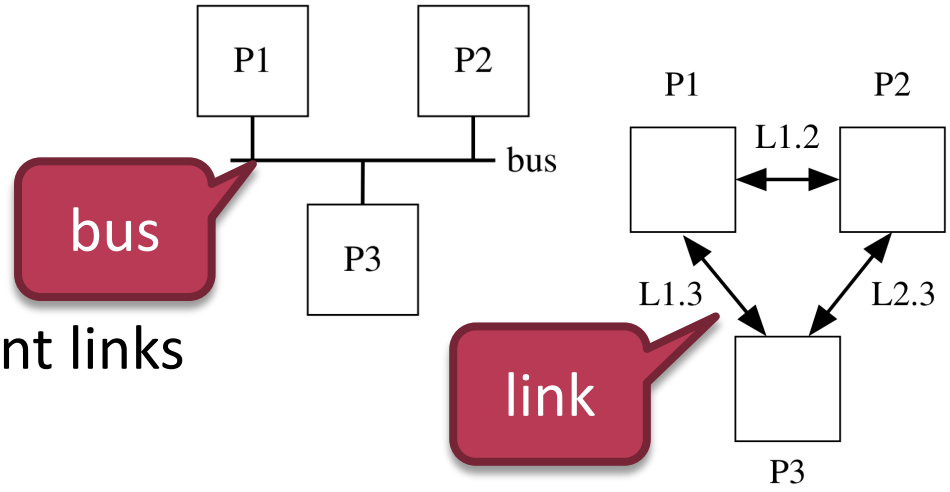
Goal

- Allocate and schedule the software components to hardware components to tolerate *K* hardware failures, where $K < N$. (*N* is the number of processors)
  - If *K* number of hardware components fail, the process should still provide output.
  - If *K* number of hardware components fail, the tasks should not be rescheduled to provide output.

Communication Time

- Intra-communication on the same P
  - Output of a task is sent to another task on the same P
  - For an intra-communication, the time is always **0**

- Inter-Communication between different Ps
  - Output of a task is sent to another task on different P
  - Bus
    - Always the same time
  - Links
    - Different time on different links



Notations

- ***Candidates^{itr}*:** The list of candidate S components, after the itrth iteration of the algorithm.
- ***Scheduled^{itr}*:** The list of scheduled S components, after the itrth iteration of the algorithm.
- ***Pred(s_i)*:** The set of predecessors of s_i component.
- ***Succ(s_i)*:** The set of successors of s_i component.
- ***S^{itr}(s_i,p_t)*:** The earliest time at which component s_i can start execution on processor p_t, taking into account all the predecessors replicas except if s_{i-1} and s_i are allocated to the same p_t
- ***Δ(s_i,p_t)*:** is the execution duration of s_i on p_t

Algorithm

Itr^0, initialize the list of candidates and scheduled sw components

$Scheduled^0=\emptyset$, $Candidates^1=\{s\in SW \mid Pred(s)\in Scheduled^0\}$

Itr^n, while $Candidates^n\neq\emptyset$ **do**

(1) Compute the scheduling pressure (σ) for each $s\in Candidates^n$ on all the hardware components and keep the first $K+1$ results for each s. $\sigma(s_i, p_t)=S^n(s_i,p_t)+\Delta(s_i,p_t)$

(2) Select the worst candidate s_i according to the remaining σ values, and schedule it to the first $K+1$ hardware components.

(3) Update the list of candidates and scheduled sw components

$Scheduled^n = Scheduled^{n-1} \cup \{s_i\}$,

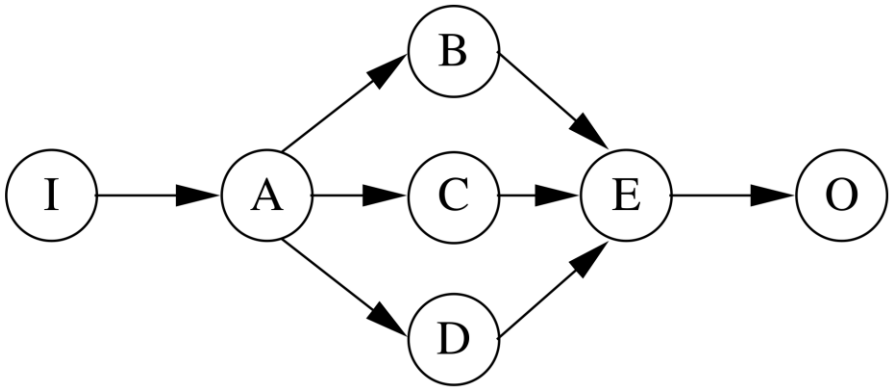
$Candidates^{n+1}=Candidates^n /\{s_i\} \cup$

$\{s_i' \in Succ(s_i) \mid Pred(s_i')\in Scheduled^n\}$

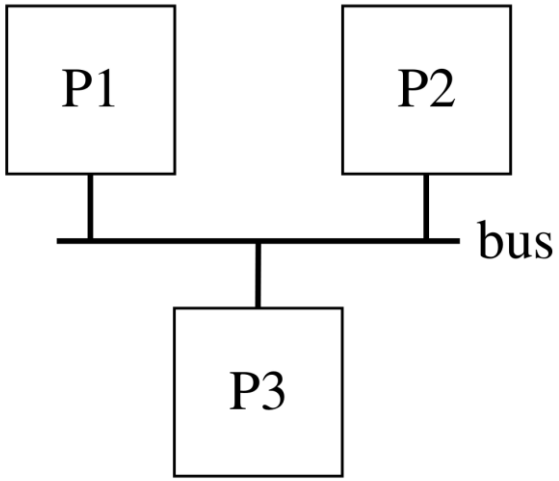
Add a successor only if its predicates are scheduled

Example I (Bus, K=1)

Software Architecture



Platform Architecture



Execution Time (ET)

		operation						
	time units	I	A	B	C	D	E	O
proc.	P1	1	2	3	2	3	1	1.5
	P2	1	2	1.5	3	1	1	1.5
	P3	∞	2	1.5	1	1	1	∞

Data Transfer (DT)

		data-dependency							
	time units	I ▷ A	A ▷ B	A ▷ C	A ▷ D	B ▷ E	C ▷ E	D ▷ E	E ▷ O
link	bus	1.25	0.5	0.5	1	0.5	0.6	0.8	1

Example from: Girault, Alain, et al. "Fault-tolerant static scheduling for real-time distributed embedded systems." *INT. CONF. ON DIST. COMP. SYS.*. Vol. 21. IEEE Computer Society; 1999, 2001.

Example I (Bus, K=1) – Solution – Itr: 0

$Scheduled^0 = \emptyset$, $Candidates^1 = \{I\}$

Example I (Bus, K=1) – Solution – Itr: 1

$Candidates^1 \neq \emptyset \rightarrow do$

$\sigma(I,P1)=0+1=1$

$\sigma(I,P2)=0+1=1$

$\sigma(I,P3)=0+undefined=undefined$

Example I (Bus, K=1) – Solution – Itr: 1

$Candidates^1 \neq \emptyset \rightarrow do$

$\sigma(I,P1)=0+1=1$

$\sigma(I,P2)=0+1=1$

$\sigma(I,P3)=0+undefined=undefined$

Select K+1 lowest

Example I (Bus, K=1) – Solution – Itr: 1

$Candidates^1 \neq \emptyset \rightarrow do$

$\sigma(I,P1)=0+1=1$

$\sigma(I,P2)=0+1=1$

$\sigma(I,P3)=0+undefined=undefined$

Select K+1 lowest

Select highest from the remaining

Example I (Bus, K=1) – Solution – Itr: 1

$Candidates^1 \neq \emptyset \rightarrow do$

$\sigma(I,P1)=0+1=1$

$\sigma(I,P2)=0+1=1$

$\sigma(I,P3)=0+undefined=undefined$

Select K+1 lowest

Select highest from the remaining

allocate I to P1 and P2,

schedule I on P1 from 0,

schedule I on P2 from 0

Example I (Bus, K=1) – Solution – Itr: 1

$Candidates^1 \neq \emptyset \rightarrow do$

$\sigma(I,P1)=0+1=1$

$\sigma(I,P2)=0+1=1$

$\sigma(I,P3)=0+undefined=undefined$

Select K+1 lowest

Select highest from the remaining

allocate I to P1 and P2,
schedule I on P1 from 0,
schedule I on P2 from 0

$Scheduled^1=\{I\}, Candidates^2=\{A\}$

Example I (Bus, K=1) – Solution – Itr: 2

$Candidates^2 \neq \emptyset \rightarrow do$

$\sigma(A,P1)=1+2=3$

$\sigma(A,P2)=1+2=3$

$\sigma(A,P3)=2.25+2=4.25$

allocate A to P1 and P2,
schedule A on P1 from 1,
schedule A on P2 from 1

- "A" can start immediately after "I" on P1 and P2 → **S = "I done"**
- "A" has to wait the output of "I" from P1 and P2 on P3 → **S = max("I done"+DT(IA))**

DT from all replicas, but in this case these two are equal

$Scheduled^1 = \{I,A\}$, $Candidates^2 = \{BCD\}$

Example I (Bus, K=1) – Solution – Itr: 3

$Candidates^3 \neq \emptyset \rightarrow do$

$\sigma(B,P1)=3+3=6$

$\sigma(B,P2)=3+1.5=4.5$

$\sigma(B,P3)=3.5+1.5=5$

$\sigma(C,P1)=3+2=5$

$\sigma(C,P2)=3+3=6$

$\sigma(C,P3)=3.5+1=4.5$

$\sigma(D,P1)=3+3=6$

$\sigma(D,P2)=3+1=4$

$\sigma(D,P3)=4+1=5$

- **K+1 lowest for B,C,D**

$\sigma(B,P3)=5$

$\sigma(B,P1)=4.5$

$\sigma(C,P3)=4.5$

$\sigma(C,P1)=5$

$\sigma(D,P2)=4$

$\sigma(D,P3)=5$

From 2 equal values, you can choose randomly

Select the highest from the remaining: $\sigma(B,P3)=5 \rightarrow allocate\ B$

Example I (Bus, K=1) – Solution – Itr: 3

$Candidates^3 \neq \emptyset \rightarrow do$

$\sigma(B,P1)=3+3=6$

$\sigma(B,P2)=3+1.5=4.5$

$\sigma(B,P3)=3.5+1.5=5$

$\sigma(C,P1)=3+2=5$

$\sigma(C,P2)=3+3=6$

$\sigma(C,P3)=3.5+1=4.5$

$\sigma(D,P1)=3+3=6$

$\sigma(D,P2)=3+1=4$

$\sigma(D,P3)=4+1=5$

- **K+1 lowest for B,C,D**

$\sigma(B,P3)=5$

$\sigma(B,P2)=4.5$

$\sigma(C,P3)=4.5$

$\sigma(C,P1)=5$

$\sigma(D,P2)=4$

$\sigma(D,P3)=5$

allocate B to P2 and P3,
schedule B on P2 from 3,
schedule B on P3 from 3.5

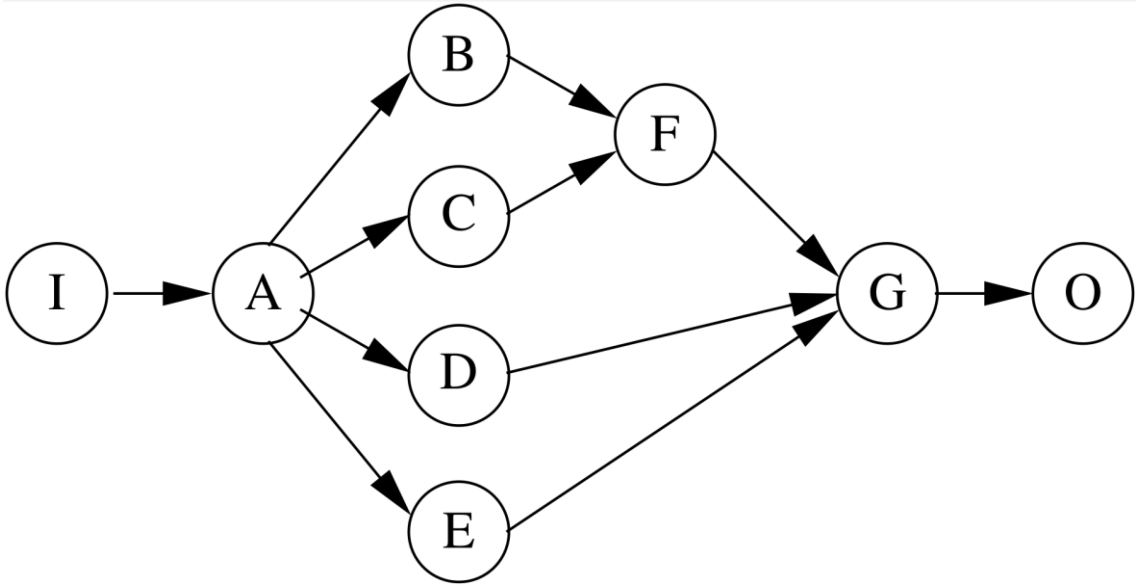
$Scheduled^2 = \{I,A,B\}$,
$Candidates^3 = \{CD\}$

Example I (Bus, K=1) – Solution – Itr: final

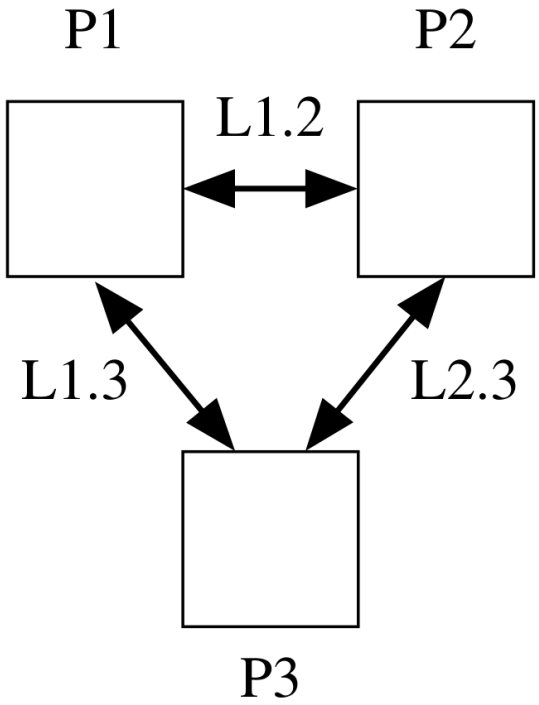
	P1		Bus (1↔2)			P2		Bus (2↔3)			P3		Bus (3↔1)		
	Start	End	Data	Start	End	Start	End	Data	Start	End	Start	End	Data	Start	End
I	0	1				0	1								
A	1	3				1	3	A>B	3	3.5			A>B	3	3.5
B						3	4.5				3.5	5			
C	3	5	C>E	5	5.6			C>E	6	6.6	5	6			
D						4.5	5.5				6	7			
E			E>O	7.6	8.6	6.6	7.6				7	8	E>O	8	9
O	9	10.5				7.6	9.1								

Example II (Links, K=1)

Software Architecture



Platform Architecture



Example from: Girault, Alain, et al. "An algorithm for automatically obtaining distributed and fault-tolerant static schedules." Proceeding of International Conference on Dependable Systems and Networks. IEEE, 2003.

Example II (Links, K=1)

time	operation								
proc.	I	A	B	C	D	E	F	G	O
P1	1	2	3	2	3	1	2	1.4	1.4
P2	1.3	1.5	1	3	1.7	1.2	2.5	1	∞
P3	∞	1	1.5	1	3	2	1	1.5	1.8

time	data-dependency					
link	I ▷ A	A ▷ B	A ▷ C	A ▷ D	A ▷ E	B ▷ F
L1.2	1.75	1	1	1.5	1	1
L2.3	1.25	0.5	0.5	1	0.5	0.5
L1.3	1.25	0.5	0.5	1	0.5	0.5

time	data-dependency				
link	C ▷ F	D ▷ G	E ▷ G	F ▷ G	G ▷ O
L1.2	1.3	1.9	1.3	1	1.1
L2.3	0.8	1.4	0.8	0.5	0.6
L1.3	0.8	1.4	0.8	0.5	0.6

Example II (Links, K=1) - Solution

	P1		L1.2			P2		L2.3			P3		L1.3		
	Start	End	Data	Start	End	Start	End	Data	Start	End	Start	End	Data	Start	End
I	0	1				0	1.3								
A	1	3				1.3	2.8	A>C	2.8	3.3			A>C	3	3.5
D	3	6				2.8	4.5	A>B	2.8	3.3			A>B	3	3.5
C			C>F	7.5	8.8	4.5	7.5				3.5	4.5	C>F	4.5	5.3
B						7.5	8.5				4.5	6			
F	8.8	10.8									6	7			
E			E>G	9.7	11	8.5	9.7				7	9	E>G	9	9.8
G	11	12.4									9	10.5			
O	12.4	13.8									10.5	12.3			

Note

- This algorithm is a simplified version of

  Alain G., Hamoudi K., Mihaela S., Yves S.:
  An Algorithm for Automatically Obtaining Distributed and Fault-Tolerant Static Schedules.
  DSN 2003: 159-168