

# IMPLEMENTATION

set of skeletons/patterns

NEW LANGUAGE

LIBRARY

own syntax

tools for what?

tools

Prog

prog

↳ calls  
to comm.  
& synchro  
libraries

↳ "host language"  
(sep language)

part of  
syntax

part of  
the tools

P3L

L = P3L

L' = C + MPI calls

Muesli (MUNSTER)

L = C++ L' = C++ MPI  
OPENMP

P3L

ABSTRACT  
SYNTAX

ANACLETO

C/C++  
toolchain

```

seq F in(...) out(...)
$cf ..... }cf
end seq
form main in(...) out(...)
  F in(...) out(...)
end form

```

F3L source code

gcc g++ → compile bodies of seq constructs  
 reverse of extracting bodies

form →

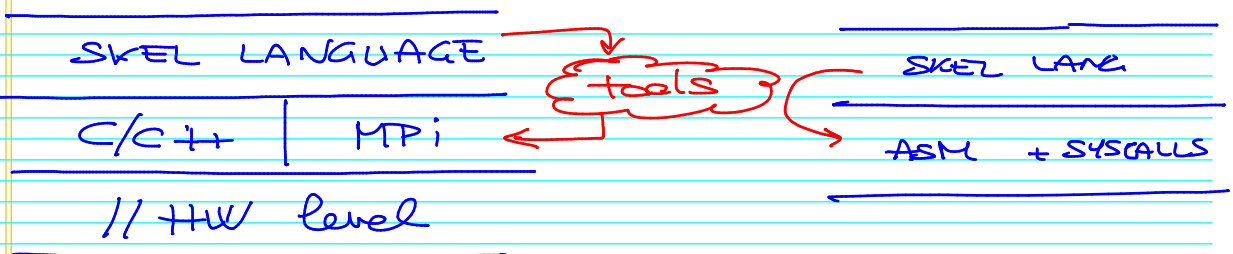
```

MPI program
main( ) {
  int p = MPI_Myrank( )
  switch (p) {
    case 0: { F ..... }
    case 1: { C ..... }
    default: { W ..... }
  }
  MPI_Send
  MPI_Receive
}

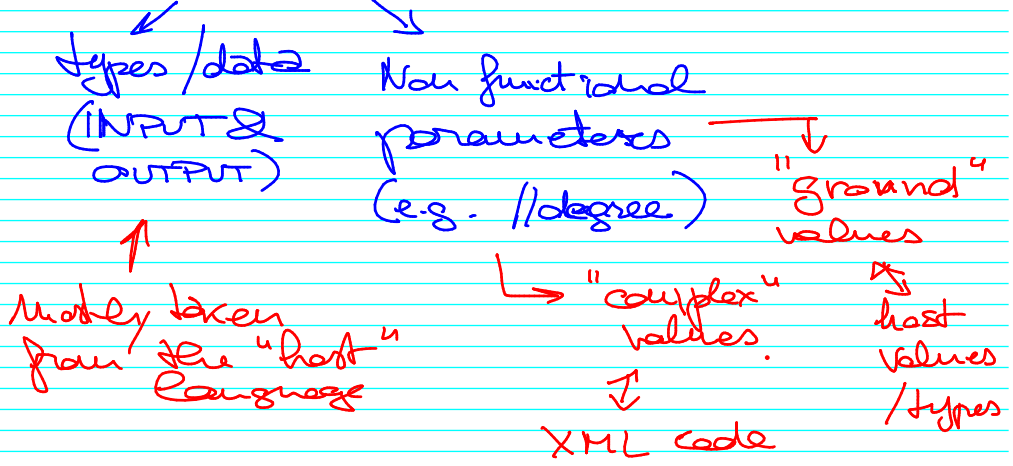
```

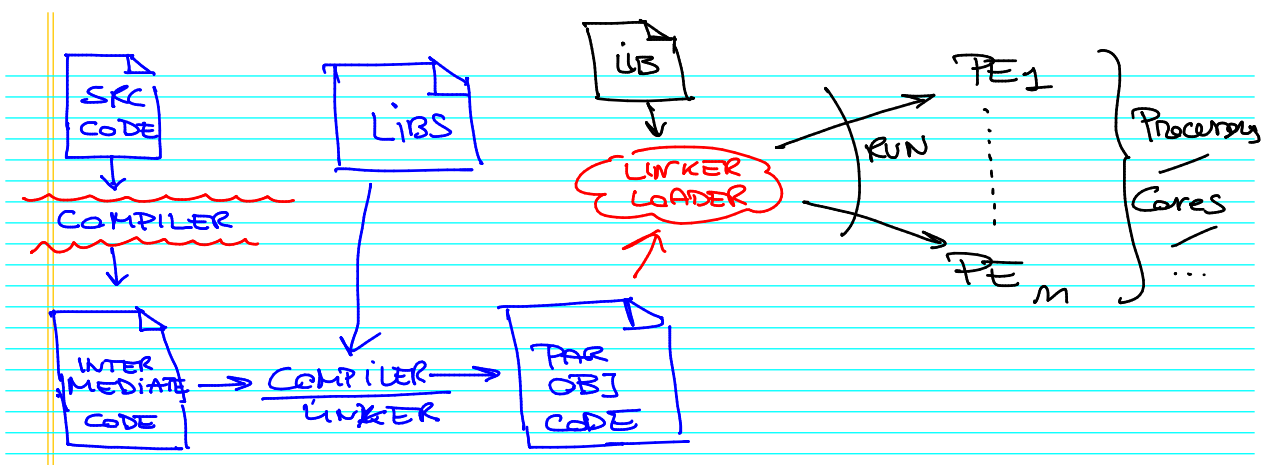
> mpicc prog.c  
 ↓  
 a.out

> mpiexec a.out -np 8

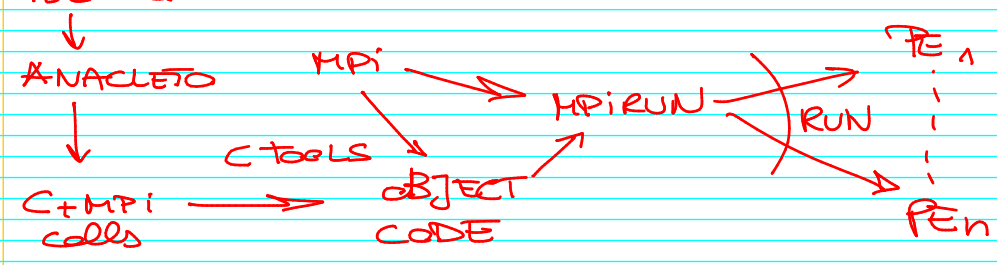


WAYS USED TO EXPRESS PARAMETERS

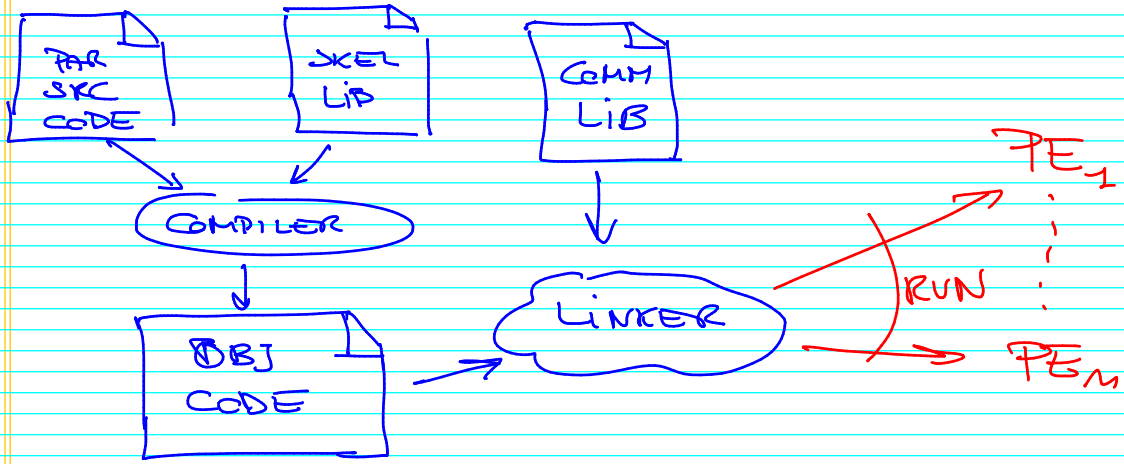




PSL Code



# LIBRARY



LIB		HOST LANG	COMM LIB
MUESLI	}	C/C++	} MPI
SKETO			
MUSKEL	}	JAVA	} JAVA RMI
SKANDIUM			} POSIX SOCKET
LIBERO			} RMI + SOCKETS
OCAML P3L			} POSIX SOCKETS
		OCAML	

TEMPLATE BASED  
VS  
MACRO DATA FLOW BASED

} IMPLEMENTATIONS

TEMPLATE

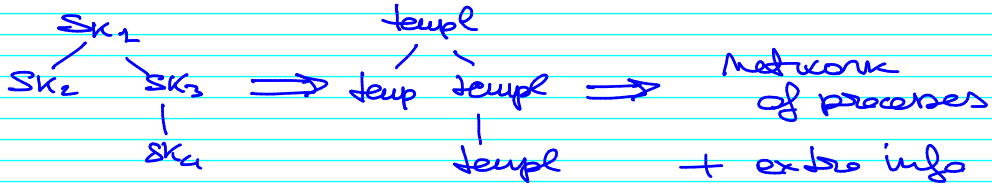
set of PRE DEFINED TEMPLATES

↓ TEMPLATE  
(AT LEAST)  
X SKELETON

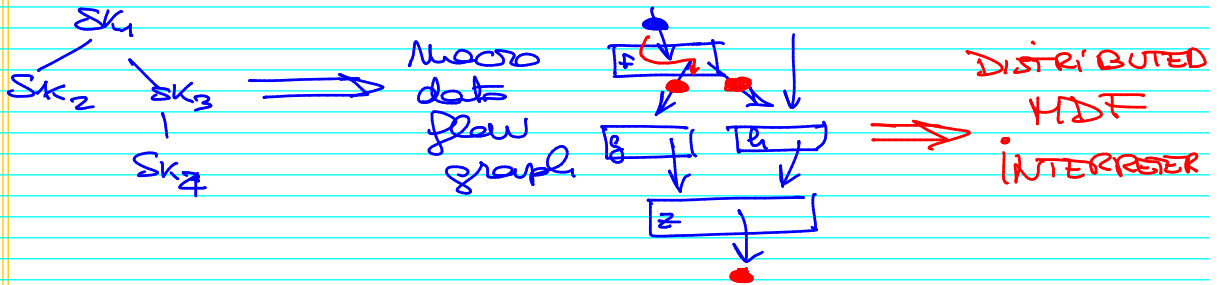
PROCESS

↓ PROCESS NETWORK

MATCHING PHASE

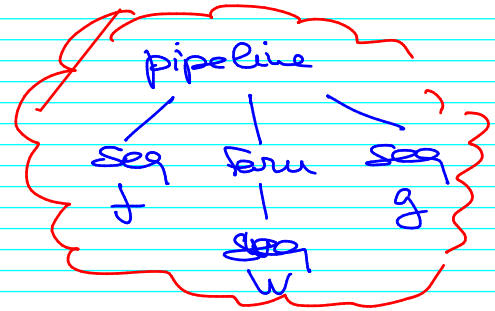


# MACRO DATA FLOW APPROACH



```

Skeleton f = new F(...);
Skeleton g = new G(...);
Skeleton w = new W(...);
Skeleton form = new Form(w);
Skeleton main =
    Pipe(Pipe(f, form), g);
    
```



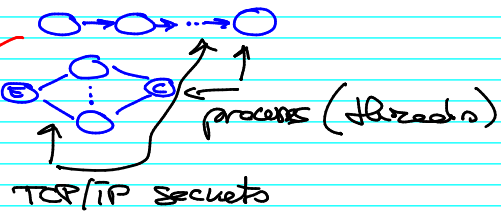
```

Manager mg = new Manager(main, infile, outfile);
mg.run();
    
```

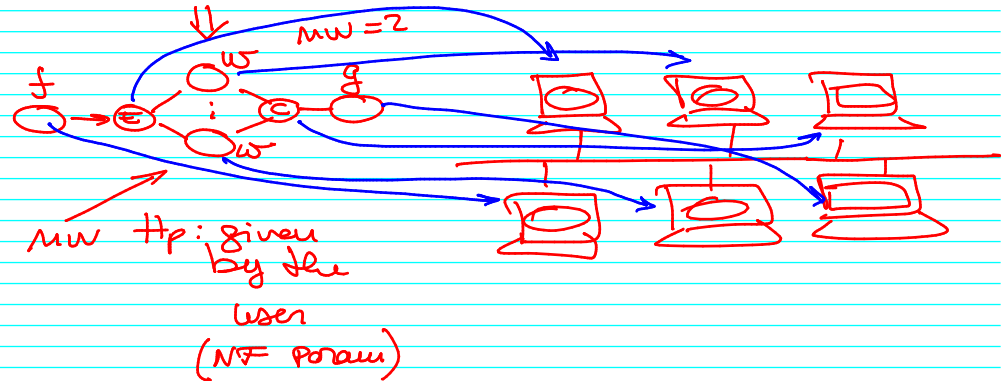
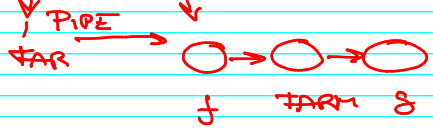
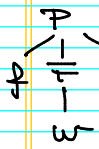
# TEMPLATE BASED IMPL

LIB:

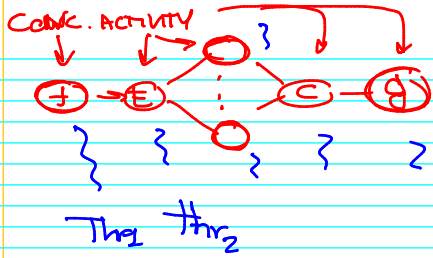
pipeline  
form

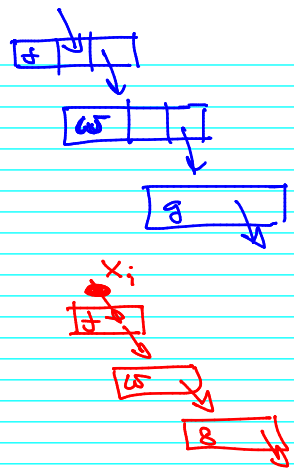
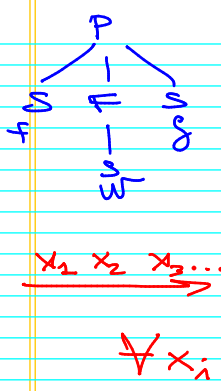


- templates
- parametric
  - functions computed
  - types
  - parallelism degrees
  - ...

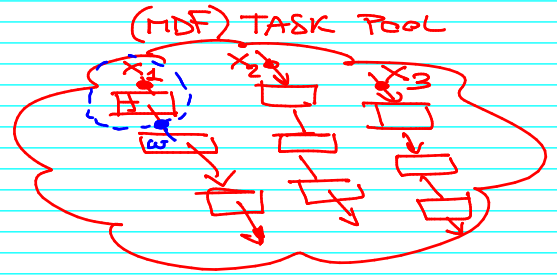








MDF  
templates



interpreters ::  $\rho_{df}$   
 (n, distributed, ed)

- 1) FETCH FIREABLE INSTRUCTION FROM TASK POOL
- 2) EXECUTE IT
- 3) STORE RESULT IN THE PROPER TOKENS