Getting Your Priorities Right

Fine-grained threading
Computation
e.g. sort, compress

Interaction
e.g. GUI

E-mail client

Need priorities - Millions of threads, no way to distinguish

2 problems w/ most ways of handling priorities
1. Fixed order → Anti-modular
2. Priority inversions - high-prio thread waiting for low-prio

Solution: PriML
Partially ordered prios - prio + order decls
spawn/sync + priorities
Qsort code

Type sys. prevents prio inversions

$\tau ::= \cdots \mid \tau\ \text{thread}[\rho] \mid \tau\ \text{cmd}[\rho] \mid \forall \pi : C . \tau$

$\rho ::= p \mid \pi$

$e ::= \cdots \mid \Lambda \pi : C . e$

$C ::= \rho \leq \rho \mid C \wedge C$

$m ::= x \leftarrow e; m \mid \text{spawn}[\rho; \tau]\{m\} \mid \text{sync}\ e \mid \text{ret}\ e$

$\Gamma \vdash m \div \tau @ p$: m returns a τ, runs @prio. p

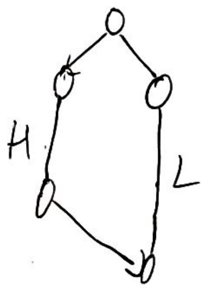
$$\frac{\Gamma \vdash m \div \tau @ p'}{\Gamma \vdash \text{spawn}[p';\tau]\{m\} \div \tau\ \text{thread}[p] @ p} \quad \text{(SPAWN)}$$

$$\frac{\Gamma \vdash e : \tau\ \text{thread}[p] \quad \Gamma \vdash p \preceq p'}{\Gamma \vdash \text{sync}\ e \div \tau @ p} \quad \text{(SYNC)}$$

$$\frac{\Gamma, \pi, C \vdash e : \tau}{\Gamma \vdash \Lambda \pi : C.e : \forall \pi : C.\tau}$$

Cost Semantics

Model program as a DAG - label threads w/priorities



Greedy schedule - Assign vertices to procs so no procs are idle unless necessary.

$T \le \frac{W}{P} + S$

Prompt schedule - Greedy + run highest-prio vertices possible

Bound response time: How long from when a thread is spawned until it completes.

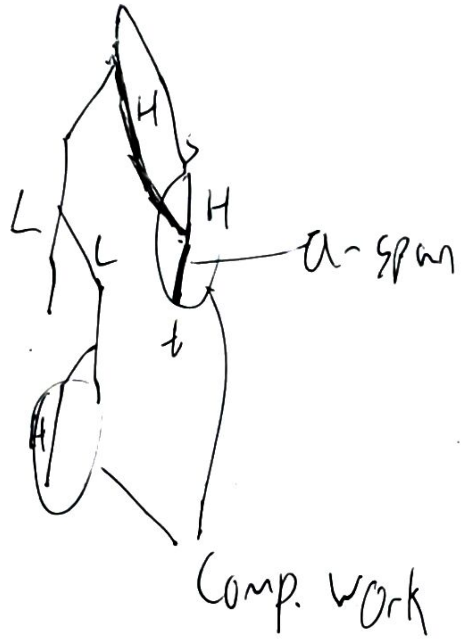
Let $a = s \cdots t$

$$RT(a) \le \frac{W_{\nprec p}(a)}{P} + S_a$$

Competitor work

a-span (longest path ending at t)

Bound holds unless there is a prio. inv.



Cost semantics: $m \downarrow v; g$ ↙ graph

Theorem: If $\cdot \vdash m \div \tau @ p$, and $m \downarrow v; g$, then
g does not have a prio. inversion.

Dynamic semantics: $\mu \Rightarrow \mu'$
↑
thread pool
$a \overset{p}{\hookrightarrow} m$.

Thm: If $\cdot \vdash m \div \tau @ p$ and $a \overset{p}{\hookrightarrow} m \Rightarrow^* \mu'$ and μ' final and a
thread a is active for T transitions. Then $m \downarrow v; g$ and $\exists$ a
prompt schedule of g in which $RT(a) = T$.