Distributed race detection

Igor Walukiewicz

	\uparrow
	Monitor

Self monitoring program
Distributed monitor

Race prediction · Operations re(x), w(x) acq(l), rel(l) t_1 t_2 · Lock discipline : critical sections of the $acq(\ell)$ same lock are disjoint 2 w(y)3 w(x)r(x)4 $rel(\ell)$ 5 $acq(\ell)$ 6 w(y) $acq(\ell)$ 8

Race prediction · Operations re(x), w(x) acq(l), rel(l) t_1 t_2 · Lock discipline : critical sections of the $acq(\ell)$ same lock are disjoint w(y)w(x)· Race (W, CM, F, (X)) or (W, (X), W, (X)) the two events are concurrent in 6 r(x)(can appear next to each other) $rel(\ell)$ 5 in some sound reordering of 5) $acq(\ell)$ 6 w(y) $acq(\ell)$ 8

Race prediction · Operations re(x), w(x) acq(l), rel(l) t_1 t₂ · Lock discipline : critical sections of the $acq(\ell)$ same lock are disjoint w(y)w(x)· Race (W, (M, F, (X)) or (W, (X), W, (X)) the two events are concurrent in 6 r(x)(can appear next to each other) $rel(\ell)$ 5 in some sound reordering of 5) $acq(\ell)$ w(y)Race prediction problem: looking at 6 detect if $acq(\ell)$ 8 there is a race

Race prediction · Operations ricx, with acq(l), rel(l)

· Lock discipline : critical sections of the same lock are disjoint

· Race (Wy (M, TY, (X)) or (Wy (X), Wy, (X)) the two events are concurrent in 6 (can appear next to each other) in some sound reordering of 5/

Race prediction problem: Looking at 5 detect if there is a race t_1 t_2 t_1 t_2 $acq(\ell)$ $acq(\ell)$ 2 3 r(x)w(z)r(x)w(z) $rel(\ell)$ w(x)w(x) $rel(\ell)$ 5 Execution σ Execution σ' No data race

Klappens- 6 efire order [lamport]

What is a round reordering of 5? · operations on different Threads permute t_1 t_2 t₃ unless they concern the same lock $acq(l_1)$ e e is the smallest partial-order s.t. w(x)2 $\operatorname{rel}(\ell_1)$ 3 · events of the same thread are ordered $\operatorname{Acq}(\ell_1)$ · if ,. relle. acgle. Then $acq(\ell_2)$ ShR rel(l2) MB $acq(l_2)$ w(x)8 $rel(\ell_1)$ 9 $rel(\ell_2)$ 10

Happens- before order [lamport]

What is a round reordering of 5? · operations on different Threads permute t_2 t_1 t_3 unless they concern the same lock $acq(\ell_1)$ w(x)e = e, is the smallest partial-order s.t. 2 $\operatorname{rel}(\ell_1)$ 3 · events of the same thread are ordered $\operatorname{Acq}(\ell_1)$ · if ,.. relle)... acgle)... then $acq(l_2)$ $\leq \frac{1}{hB}$ $\operatorname{rel}(\ell_2)$ 5 & 6' if the two have the same set of events acq(ℓ2) and give the same HB order w(x)8 $rel(\ell_1)$ $rel(\ell_2)$ Race detection: race prediction for RHB 10 By looking at & detect if there is 6 AB WITH a race situation W (X) FU(X)

Vector clocks [Lamport]

Clocks measuring HB-time

Ct : J-J/W Citt - program counter of t Ct(t') - the last program counter of t' seen by t

 $\mathcal{L}_{t}^{(r)} = \mathcal{L}_{t}(t) := \mathcal{L}_{t}(t) + 1$ Ft (x), Wt (X)

Vector clocks [Lanport] Clocks measuring HB-time C+: J->// C_t(t) - program counter of t Ct(t') - the last program counter of t' seen by t $\mathcal{L}_{t}^{(r)} = \mathcal{L}_{t}(t) := \mathcal{L}_{t}(t) + 1$ $\Gamma_t(x), \omega_t(x)$ $C_t(t) := C_t(t) + I$ ť acqt (l) 711 acqtle) $\begin{array}{ccc} \forall t' \neq t & \mathcal{C}_{t}(t') = \max\left(\mathcal{C}_{t}(t') & \mathcal{C}_{t}(t')\right) \\ \forall t' & \mathcal{C}_{e}(t') = \underbrace{\mathcal{U}_{t}(t')}_{t'} & \underbrace{\mathcal{U}_{t'}(t')}_{t'} & \underbrace{\mathcal{U}_{t'$ 4

Vector clocks [Langort] Clocks measuring HB-time $C_{t}: J \rightarrow M$ Cilt) - program counter of t Cilt') - the last program counter of t' seen by t $\mathcal{L}_{t}^{(r)} = \mathcal{L}_{t}(t) := \mathcal{L}_{t}(t) + 1$ $\Gamma_t(x), \omega_t(x)$ $C_t(t) := C_t(t) + 1$ ť acqt (l) acqtle) $\begin{array}{l} \forall t' \neq t \quad \mathcal{C}_{t}(t') = \max\left(\mathcal{C}_{t}(t') \quad \mathcal{C}_{t}(t')\right) \\ \forall t' \quad \mathcal{C}_{t}(t') = \underbrace{\mathcal{U}_{t}(t')}_{t'} = \underbrace{\mathcal{U}_{t}(t')}_{t'} \end{array}$ rel (ll) $C_t(t) := C_t(t) + 1$ $rel_{+}(l)$ $\forall t' : C(t') = C_{\ell}(t')$ P

Race detection

Ct : J-J/N C_t(t) - program counter of t Ct(t') - the last program counter of t' seen by t

WE JXIN Thread that made the last write to x and its PC for every thread last read from x R*: J-JM

Race check

if W'= (t',i) and Ct(t')=i then race else R×(t):=Ct(t) F, (X) Wt (x) or $\exists t' \in \mathcal{C}_t(t') \subset \mathcal{R}^*(t')$ then race else $W^* = (t, C(t))$

Lamport. 1978. Time, Clocks, and the Ordering of Events in a Distributed System. Commun. ACM 21

Itzkovitz, Schuster, Zeev-Ben-Mordehai. 1999. Toward Integration of Data Race Detection in DSM Systems. J. Parallel Distrib. Comput.

Flanagan, Freund. 2009. FastTrack: Efficient and Precise Dynamic Race Detection. PLDI '09

Thokair, Zhang, Mathur, Viswanathan, Dynamic Race Detection with O(1) Samples, POPL'23

Race detection (distributed)

Ct: J-J/W C_t(t) - program counter of t C_t(t') - the last program counter of t' seen by t

WEE JX/M thred that made the last write to x, and its PC RX: J-JM for every thread last read from x

Race check $\begin{array}{cccc} t & \exists x & W_{t}(x) = (t', c') & \mathcal{C}_{t}(t') < i \\ & & & \\ & & & \\ & & & \\ & & & \\ e & & & \\ e & & & \\ & & & \\ e & & & \\ & & & \\ e & & & \\ & & & \\ \end{array}$ Wtilx) Wtu(X)

Race detection (distributed)

Ct: J-J/W C_t(t) - program counter of t C_t(t') - the last program counter of t' seen by t

WE JXIN Thred that made the last write to x, and its PC for every thread last read from x RX: J-JM

Race check Ix Welk = (t', c) Celt') ci Wyilx) $\alpha_{e}(x) = (t_{ij}^{*}) \quad C_{e}(t_{ij}^{*}) < j$ $J_{x} W_{t} \ell x = (\ell', c) C_{\ell} (\ell') < i$ Wyilx) $\frac{\alpha cq_t(t)}{\exists t'' \ R(x,t'') > C_t(t')}$ Tyal X)

Race detection (distributed)

Ct : J-J/N Ct(t) - program counter of t Ct(t') - the last program counter of t' seen by t

With TXIN thread that made the last write to x, and its PC Rt: J-> Muls for every thread last read from x

Update is difficult acqt^[6]

 $for x \in Var \\ W_{t}^{k} := (t', i) \\ if C_{\ell}(t') = i \quad then \quad W_{\ell}^{k} := (t', i)$ for t'ej if R't(t') > Telt' then R't') = R't(t')

Distributed vace detection, take 2

 $W_{\pm} : J \rightarrow (M \times V_{ars})^{\neq}$ Ry J - 1 (IN × Var, × Valo)* t WCX1 acqt[e] F (X/ C Race if have is x & Vous t, t2 & J Juck Pat $(c, x) \in W_t(t_1)$ $\cdot c_{p}(t_{n}) < i$ $\cdot (j, x, d) \in R_p(t_2)$ $\cdot \ell_t(t_1) < j$

Distributed race detection, take 2

Wy (t') sortal by M $W_{\pm} = J - J \left(M \times V_{avs} \right)^{\pm}$ Ry J -> (W × Var, × Vals)* acqt^[e] Race if have is x & Vous t, t, EJ such Pat $l_{\ell,K} \neq W_{\ell}(t_{j})$ $\cdot c_p(t_n) = i$ $\cdot (j, x, \alpha) \in \mathcal{R}_p(t_2)$ $\cdot \ell_{f}(t_{1}) < j$

ty moved in the ld if Celtal > Celtal MI to GMIt × writen in thell if (i,x) e W(t) is Celta to WXt

for tre MIe $C_{t}(t_{2})$ for $(i, x, \alpha) \in R_{\ell}(t_2)$ with $i > C_{\ell}(t_2)$ $| | | R_{p}(t_{u})$ if x E WX t then vacc

Distributed race detection, take 2

 $W_{\pm} = J - J \left(M \times V_{ars} \right)^{\pm}$ Ry J - 1 (IN × Var, × Valo)* acqt^[e] Race if Prove is XEVans t, t, EJ Juck Pat $(t, K) \in W_{t}(t_{1})$ $\cdot c_p(t_n) = i$ $\cdot (j, x, \alpha) \in \mathcal{R}_p(t_2)$ $\cdot \ell_t(t_1) < j$

ty moved in the ld if Celtal > Celtal MI to GMIt × writen in thell if (i,x) e V (t) i> Celto + WX+

for tre MIe $C_t(t_1)$ for $(i, x, \alpha) \in R_p(t_2)$ with $i > C_p(t_2)$ $| | | R_{p}(t_{u})$ if x EWX then vacc The work is proportional to the size of squanetsic difference.

TOPO

- Faster distributed monitoring

- Distributed monitoring for other properties · serializability \overline{V}

· ang Zz property

TOPO

- Faster distributed monitoring - Distributed monitoring for other properties · Serializability \overline{V} \bigtriangledown t_1 t_2 t_3 · any Ez property $acq(l_1)$ 2 w(x) $\operatorname{rel}(\ell_1)$ - Race prediction $*acq(l_1)$ reads from equivalence 4 $acq(\ell_2)$ 5 rel(l2) MB 6 $acq(\ell_2)$ w(x)8 $rel(\ell_1)$ 9 $rel(\ell_2)$ 10