Internet Computing: Using Reputation to Select Workers from a Pool

Evgenia Christoforou, Antonio Fernández Anta, Chryssis Georgiou, and Miguel A. Mosteiro

NETyS 2016

Internet-based task computing

- Increasing demand for processing computationally intensive tasks.
- Powerful parallel machines are expensive and are not globally available.
- Growing use and capabilities of personal computers.
- Wide access to the Internet.

Internet-based task computing

- Increasing demand for processing computationally intensive tasks.
- Powerful parallel machines are expensive and are not globally available.
- Growing use and capabilities of personal computers.
- Wide access to the Internet.

Internet emerges as a viable platform:

- Grid and cloud computing.
  - e.g. EGEE Grid, TERA Grid, Amazon's EC2
- Volunteering computing.
  - e.g. SETI@home, AIDS@home, Folding@home
- Crowd computing.
  - e.g. Amazon Mechanical Turk (human based-computing)

SETI@home by the numbers

As reported in November 2009:

- 278,832 active CPUs (out of a total of 2.4 million) in 234 countries.
- 769 TFLOPs.

SETI@home by the numbers

As reported in November 2009:

- 278,832 active CPUs (out of a total of 2.4 million) in 234 countries.
- 769 TFLOPs.

Proc. power comparable with supercomputers,

at a fraction of the cost!

Great potential limited by untrustworthy entities.

Internet-based task computing

A more dynamic and unpredictable setting:

- Number of tasks is not fixed or known a priori:
  - It may be anything from one to many, even unbounded.
  - Tasks arrive dynamically and continuously.
- Dynamic participation:
  - Participating processors (workers) could change over time, not only due to failures.
- Increased frequency of failures:
  - Failures are the norm rather than the exception.

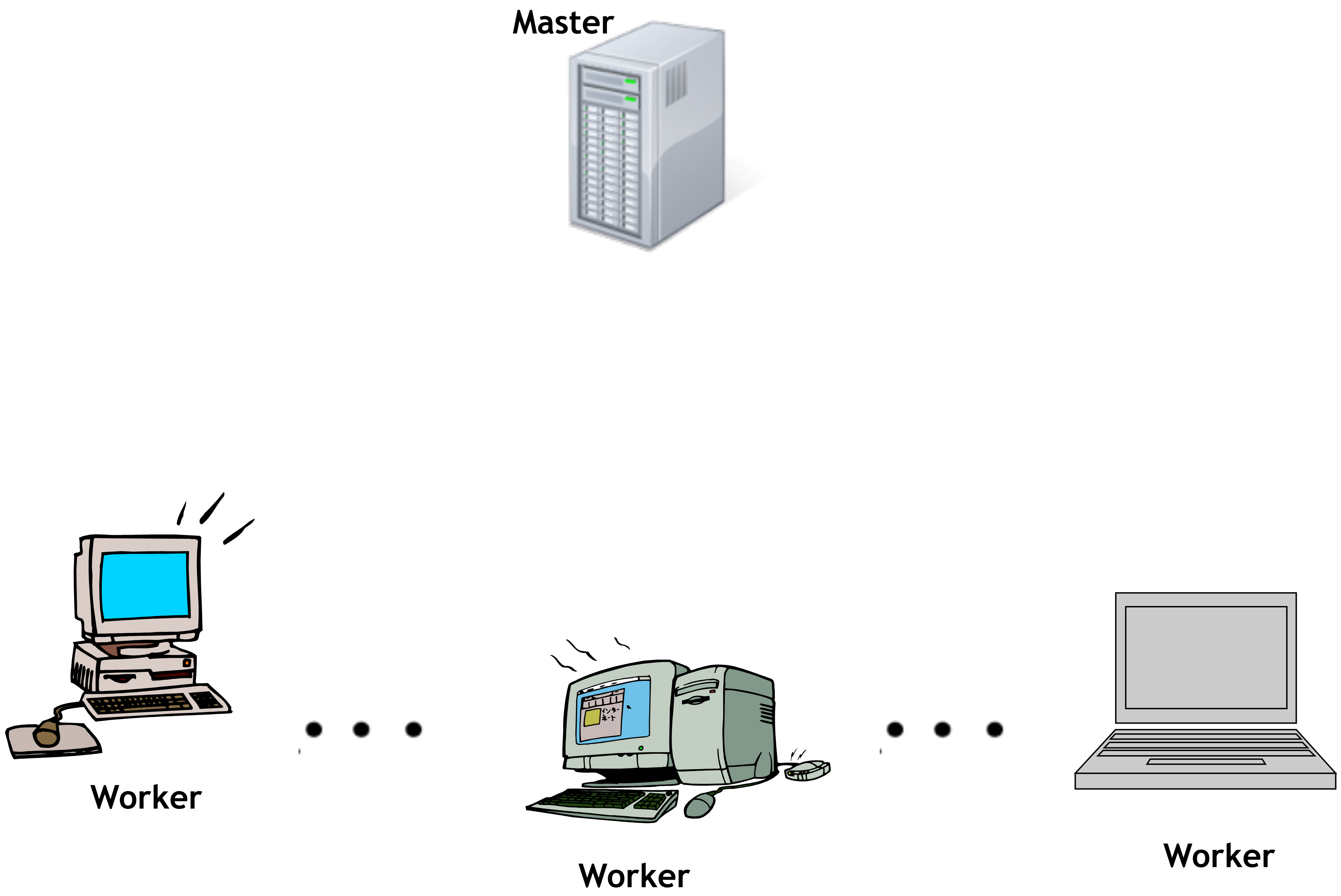
Internet-based task computing

A more dynamic and unpredictable setting:

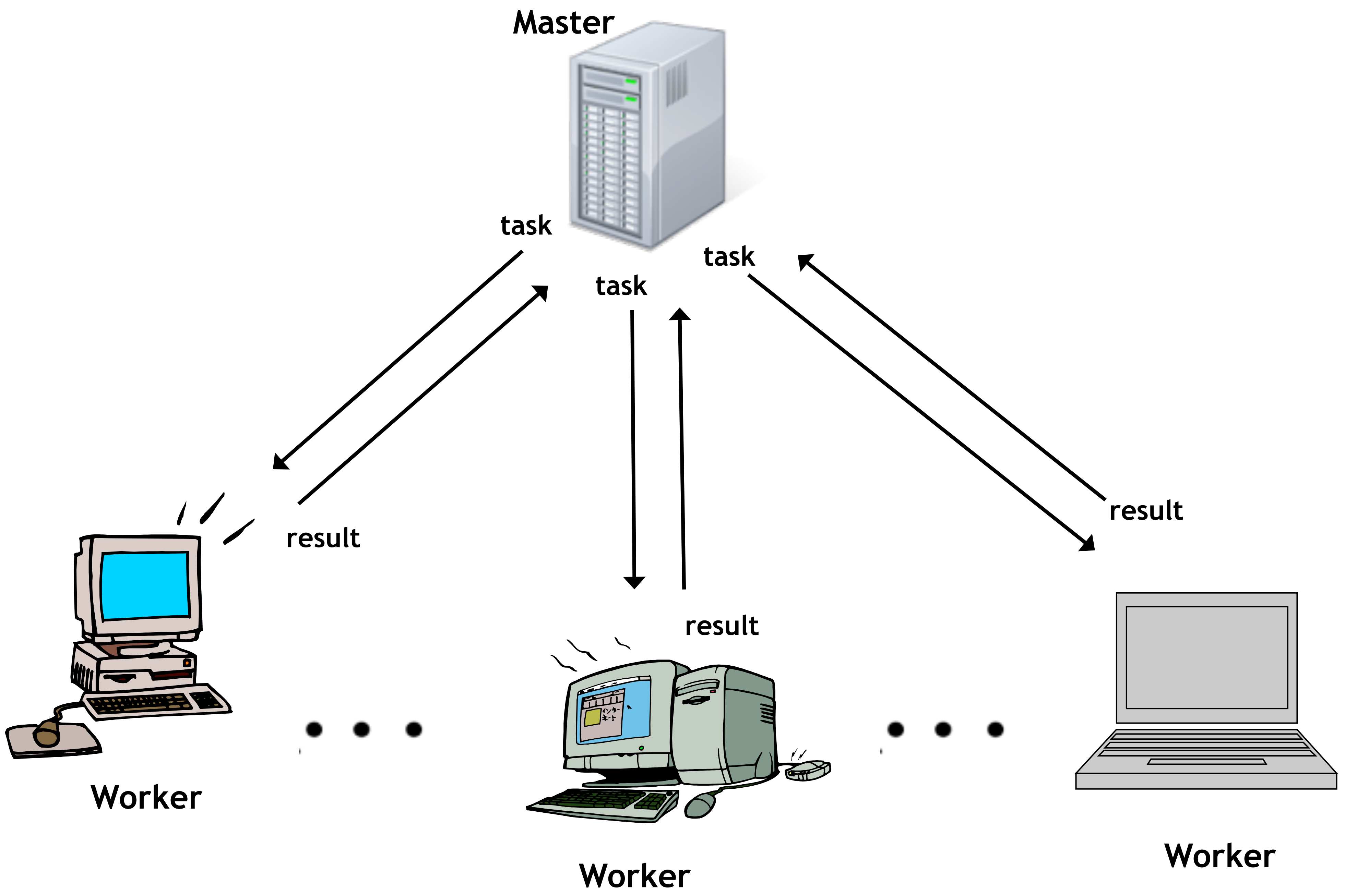
- Number of tasks is not fixed or known a priori:
  - It may be anything from one to many, even unbounded.
  - Tasks arrive dynamically and continuously.
- Dynamic participation:
  - Participating processors (workers) could change over time, not only due to failures.
- Increased frequency of failures:
  - Failures are the norm rather than the exception.

A whole new world to study tradeoffs between efficiency and fault tolerance!

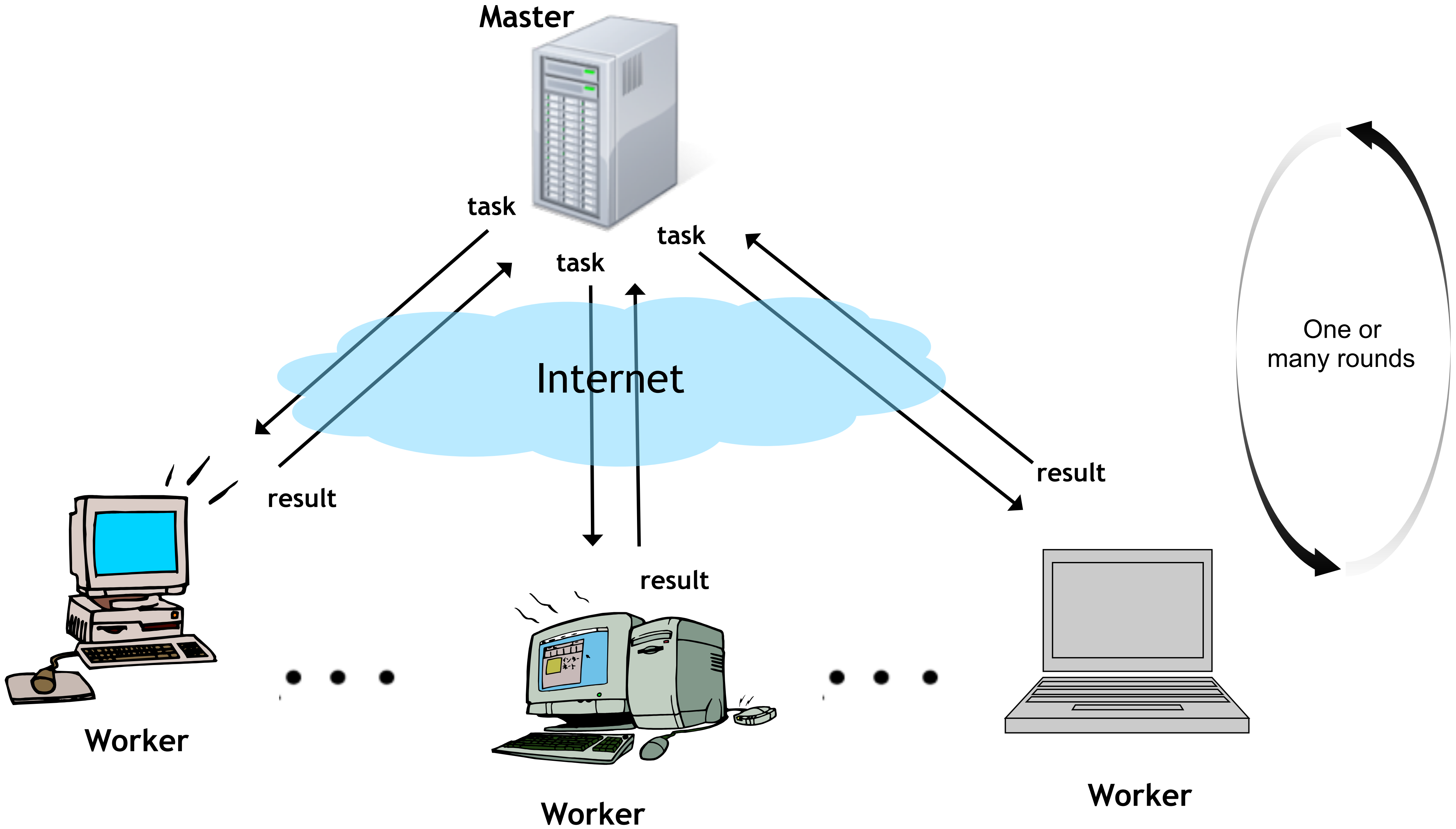
Master-worker computing



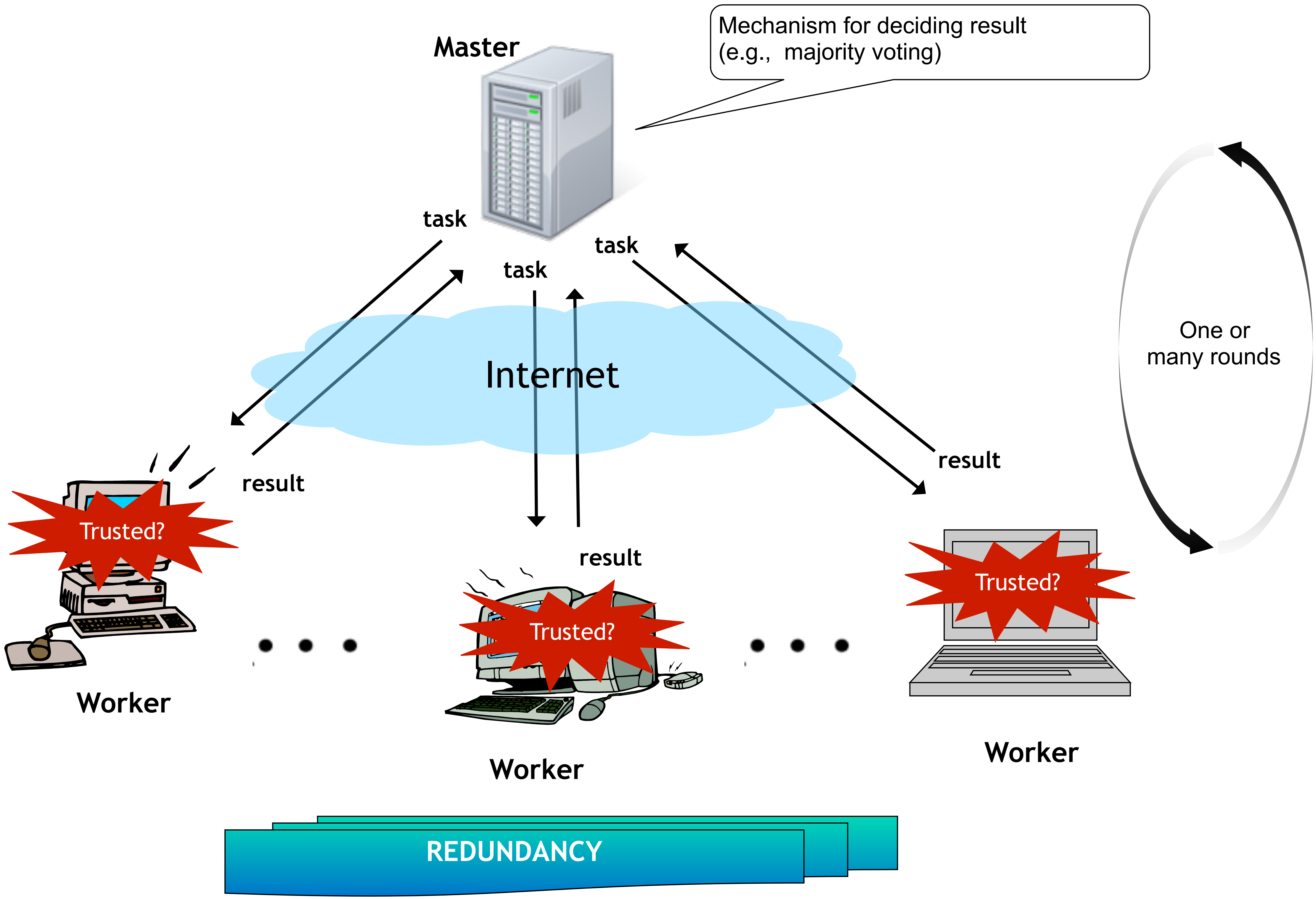
Master-worker computing



Master-worker computing



Master-worker computing



Types of workers

Classical distributed computing approach:

- Malicious workers: always return a fabricated incorrect result.
- Altruistic workers: always compute and return a correct result.

[Fernandez-Anta et al.; Konwar; Sarmenta]

Game theoretical approach:

- Rational workers: choose the strategy that maximizes benefit.

[Abraham et al;Golle et al;Shneidman et al; Yurkewych;NCA'08;PLOS One'15]

All three types considered:

- Mechanisms with reward/punishment schemes that
  - provide incentives for rationals to be honest
  - cope with malicious actions.

[IPDPS'10;NCA'11;DISC'11;TC'14]

Rewards/punishments

WP_C	worker punishment for being caught cheating
WC_T	worker cost of computing the task
WB_Y	worker benefit from master acceptance
MC_Y	master cost for accepting the worker answer
MC_A	master cost for auditing worker answer
MB_R	master benefit from accepting the correct answer
MP_W	master punishment for accepting a wrong answer

Task computing scheme

- Master assigns a task to n workers.
- Workers:
  - Malicious: fabricates a result.
  - Altruistic: computes the result.
  - Rational: cheats with probability pC.
- Master audits with probability pA.
- If master audits:
  - computes the task.
  - rewards honest workers and penalizes cheaters.
- If master does not audit:
  - accepts value returned by "majority" of workers.
  - rewards those in the "majority".

Types of computations

One-shot	Multi-round
•Model the decisions (cheat or not, audit or not) as a game.	•Evolutionary dynamics: pC and pA updated after each round (n is fixed).
•Find conditions on the parameters for Nash equilibria.	•Reinforcement learning: update function of worker profit aspiration and master tolerance to loss.
•The pC, pA, and n obtained yield correctness at low cost for the master, even in presence of malice.	•Objective: eventual correctness.
•Trade-off: probability of correct vs. cost.	•Trade-off: time to correct vs. cost.
•Constrains: malice, collusion, unreliable communication.	•Constrains/features: malice, unreliable communication, reputation system.
[NCA'08;IPDPS'10;DISC'11;NCA'11;TC'14, PLOS One'15]	[PODC'12, EUROPAR'12, OPODIS'13, JOSS'13, CCPE'13]

Types of computations

One-shot	Multi-round
•Model the decisions (cheat or not, audit or not) as a game.	•Evolutionary dynamics: pC and pA updated after each round (n is fixed).
•Find conditions on the parameters for Nash equilibria.	•Reinforcement learning: update function of worker profit aspiration and master tolerance to loss.
•The pC, pA, and n obtained yield correctness at low cost for the master, even in presence of malice.	•Objective: eventual correctness.
•Trade-off: probability of correct vs. cost.	•Trade-off: time to correct vs. cost.
•Constrains: malice, collusion, unreliable communication.	•Constrains/features: malice, unreliable communication, reputation system.
[NCA'08;IPDPS'10;DISC'11;NCA'11;TC'14, PLOS One'15]	[PODC'12, EUROPAR'12, OPODIS'13, JOSS'13, CCPE'13]

But, workers may not be available all the time!

Why not taking advantage of N>>n? (Internet scale)

Our contributions

- We present a mechanism that
  1. it is resilient to non-responsive and unreliable workers:
     - *responsiveness reputation:* replies/assignments ratio.
     - 3 *truthfulness reputations:* ~BOINC, [OPODIS'13], [Sonnek et al.].
  2. leverages availability of N>>n workers:

     the master chooses the most reputable n workers for each computation.

Our contributions

- We present a mechanism that
  1. it is resilient to non-responsive and unreliable workers:
     - *responsiveness reputation:* replies/assignments ratio.
     - 3 *truthfulness reputations:* ~BOINC, [OPODIS'13], [Sonnek et al.].
  2. leverages availability of N>>n workers:

     the master chooses the most reputable n workers for each computation.
- We study feasibility in absence of rationals:

  showing pools of workers such that eventual correctness
  - cannot be achieved unless the master always audits
  - achieved forever with minimal auditing.

Our contributions

- We present a mechanism that
  1. it is resilient to non-responsive and unreliable workers:
     - *responsiveness reputation:* replies/assignments ratio.
     - 3 *truthfulness reputations:* ~BOINC, [OPODIS'13], [Sonnek et al.].
  2. leverages availability of N>>n workers:

     the master chooses the most reputable n workers for each computation.
- We study feasibility in absence of rationals:

  showing pools of workers such that eventual correctness
  - cannot be achieved unless the master always audits
  - achieved forever with minimal auditing.
- Experimental evaluation:

  complements analysis for scenarios where rational workers exist.

  reputation-types comparison showing reliability/cost trade-offs.

Types of reputation

- Responsiveness:

$$\rho(w) = \frac{\texttt{replies}(w) + 1}{\texttt{assignments}(w) + 1}$$

- Truthfulness:

LINEAR:
[Sonnek et al.]

$$\rho(w) = \frac{\texttt{audited-correct}(w) + 1}{\texttt{audited}(w) + 1}$$

EXPONENTIAL:
[OPODIS'13]

$$\rho(w) = \frac{\kappa^{\texttt{audited-correct}(w)}}{\kappa^{\texttt{audited}(w)}}, \kappa > 1$$

BOINC:

$$\rho(w) = 1 - \frac{1}{\texttt{streak}(w)} \quad (0 \text{ if } \texttt{streak}(w) < 10)$$

Protocol

Algorithm 1 Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$
for $i \leftarrow 0$ *to* N **do**
    $select_i \leftarrow 0$; $reply\_select_i \leftarrow 0$; $audit\_reply\_select_i \leftarrow 0$; $correct\_audit_i \leftarrow 0$; $streak_i \leftarrow 0$
    $\rho_{rs_i} \leftarrow 1$; *initialize* ρ_{tr_i} *// initially all workers have the same reputation*
for $r \leftarrow 1$ *to* ∞ **do**
    $W^r \leftarrow \{i \in \mathcal{N}$: i *is chosen as one of the* n *workers with the highest* $\rho_i = \rho_{rs_i} \cdot \rho_{tr_i}$ $\}$
    $\forall i \in W^r : select_i \leftarrow select_i + 1$
    send *a task* T *to all workers in* W^r
    collect replies from workers in W^r **for** t *time*
    wait for t *time collecting replies as received from workers in* W^r
    $R \leftarrow \{i \in W^r$: *a reply from* i *was received by time* $t\}$
    $\forall i \in R : reply\_select_i \leftarrow reply\_select_i + 1$
    update *responsiveness reputation* ρ_{rs_i} *of each worker* $i \in W^r$
    audit *the received answers with probability* $p_{\mathcal{A}}$
    if *the answers were not audited* **then**
        accept *the value* m *returned by workers* $R_m \subseteq R$,
            where $\forall m', \rho_{tr_{R_m}} \geq \rho_{tr_{R_{m'}}}$ *// weighted majority of workers in* R
    else *// the master audits*
        foreach $i \in R$ **do**
            $audit\_reply\_select_i \leftarrow audit\_reply\_select_i + 1$
            if $i \in F$ **then** $streak_i \leftarrow 0$ *//* $F \subseteq R$ *is the set of responsive workers caught cheating*
            else $correct\_audit_i \leftarrow correct\_audit_i + 1$, $streak_i \leftarrow streak_i + 1$ *// honest responsive*
            update *truthfulness reputation* ρ_{tr_i} *// depending on the type used*
        if $\rho_{tr_R} = 0$ **then** $p_{\mathcal{A}} \leftarrow \min\{1, p_{\mathcal{A}} + \alpha_m\}$
        else
            $p'_{\mathcal{A}} \leftarrow p_{\mathcal{A}} + \alpha_m(\rho_{tr_F}/\rho_{tr_R} - \tau)$
            $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p'_{\mathcal{A}}\}\}$
    $\forall i \in W^r$: **return** Π_i *to worker* i *// the payoff of workers in* $W^r \setminus R$ *is zero*

Protocol

for each round of computation

Algorithm 1 Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$
for $i \leftarrow 0$ *to* N **do**
  $select_i \leftarrow 0$; $reply\_select_i \leftarrow 0$; $audit\_reply\_select_i \leftarrow 0$; $correct\_audit_i \leftarrow 0$; $streak_i \leftarrow 0$
  $\rho_{rs_i} \leftarrow 1$; *initialize* ρ_{tr_i} *// initially all workers have the same reputation*
for $r \leftarrow 1$ *to* ∞ **do**
  $W^r \leftarrow \{i \in \mathcal{N}$: *i is chosen as one of the n workers with the highest* $\rho_i = \rho_{rs_i} \cdot \rho_{tr_i}$ $\}$
  $\forall i \in W^r : select_i \leftarrow select_i + 1$
  send *a task* T *to all workers in* W^r
  collect replies from workers in W^r **for** t *time*
  wait for t *time collecting replies as received from workers in* W^r
  $R \leftarrow \{i \in W^r$: *a reply from i was received by time t*$\}$
  $\forall i \in R : reply\_select_i \leftarrow reply\_select_i + 1$
  update *responsiveness reputation* ρ_{rs_i} *of each worker* $i \in W^r$
  audit *the received answers with probability* $p_{\mathcal{A}}$
  if *the answers were not audited* **then**
    accept *the value* m *returned by workers* $R_m \subseteq R$,
      where $\forall m', \rho_{tr_{R_m}} \geq \rho_{tr_{R_{m'}}}$ *// weighted majority of workers in R*
  else *// the master audits*
    foreach $i \in R$ **do**
      $audit\_reply\_select_i \leftarrow audit\_reply\_select_i + 1$
      if $i \in F$ **then** $streak_i \leftarrow 0$ *//* $F \subseteq R$ *is the set of responsive workers caught cheating*
      else $correct\_audit_i \leftarrow correct\_audit_i + 1$, $streak_i \leftarrow streak_i + 1$ *// honest responsive*
      update *truthfulness reputation* ρ_{tr_i} *// depending on the type used*
    if $\rho_{tr_R} = 0$ **then** $p_{\mathcal{A}} \leftarrow \min\{1, p_{\mathcal{A}} + \alpha_m\}$
    else
      $p'_{\mathcal{A}} \leftarrow p_{\mathcal{A}} + \alpha_m(\rho_{tr_F}/\rho_{tr_R} - \tau)$
      $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p'_{\mathcal{A}}\}\}$
  $\forall i \in W^r$: **return** Π_i *to worker i* *// the payoff of workers in* $W^r \setminus R$ *is zero*

Protocol

select workers according to reputation

Algorithm 1 Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$
for $i \leftarrow 0$ *to* N **do**
  $select_i \leftarrow 0$; $reply\_select_i \leftarrow 0$; $audit\_reply\_select_i \leftarrow 0$; $correct\_audit_i \leftarrow 0$; $streak_i \leftarrow 0$
  $\rho_{rs_i} \leftarrow 1$; *initialize* ρ_{tr_i} *// initially all workers have the same reputation*
for $r \leftarrow 1$ *to* ∞ **do**
  $W^r \leftarrow \{i \in \mathcal{N}$: *i is chosen as one of the* n *workers with the highest* $\rho_i = \rho_{rs_i} \cdot \rho_{tr_i}$ $\}$
  $\forall i \in W^r : select_i \leftarrow select_i + 1$
  send *a task* T *to all workers in* W^r
  collect replies from workers in W^r **for** t *time*
  wait for t *time collecting replies as received from workers in* W^r
  $R \leftarrow \{i \in W^r$: *a reply from* i *was received by time* $t\}$
  $\forall i \in R : reply\_select_i \leftarrow reply\_select_i + 1$
  update *responsiveness reputation* ρ_{rs_i} *of each worker* $i \in W^r$
  audit *the received answers with probability* $p_{\mathcal{A}}$
  if *the answers were not audited* **then**
    accept *the value* m *returned by workers* $R_m \subseteq R$,
      where $\forall m', \rho_{tr_{R_m}} \geq \rho_{tr_{R_{m'}}}$ *// weighted majority of workers in* R
  else *// the master audits*
    foreach $i \in R$ **do**
      $audit\_reply\_select_i \leftarrow audit\_reply\_select_i + 1$
      if $i \in F$ **then** $streak_i \leftarrow 0$ *//* $F \subseteq R$ *is the set of responsive workers caught cheating*
      else $correct\_audit_i \leftarrow correct\_audit_i + 1$, $streak_i \leftarrow streak_i + 1$ *// honest responsive*
      update *truthfulness reputation* ρ_{tr_i} *// depending on the type used*
    if $\rho_{tr_R} = 0$ **then** $p_{\mathcal{A}} \leftarrow \min\{1, p_{\mathcal{A}} + \alpha_m\}$
    else
      $p'_{\mathcal{A}} \leftarrow p_{\mathcal{A}} + \alpha_m(\rho_{tr_F}/\rho_{tr_R} - \tau)$
      $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p'_{\mathcal{A}}\}\}$
  $\forall i \in W^r$: **return** Π_i *to worker* i *// the payoff of workers in* $W^r \setminus R$ *is zero*

Protocol

send task and collect replies

Algorithm 1 Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$
for $i \leftarrow 0$ *to* N **do**
  $select_i \leftarrow 0$; $reply\_select_i \leftarrow 0$; $audit\_reply\_select_i \leftarrow 0$; $correct\_audit_i \leftarrow 0$; $streak_i \leftarrow 0$
  $\rho_{rs_i} \leftarrow 1$; *initialize* ρ_{tr_i} // *initially all workers have the same reputation*
for $r \leftarrow 1$ *to* ∞ **do**
  $W^r \leftarrow \{i \in \mathcal{N}$: *i is chosen as one of the n workers with the highest* $\rho_i = \rho_{rs_i} \cdot \rho_{tr_i}$ $\}$
  $\forall i \in W^r : select_i \leftarrow select_i + 1$
  send *a task T to all workers in* W^r
  collect replies from workers in W^r **for** *t time*
  wait for *t time collecting replies as received from workers in* W^r
  $R \leftarrow \{i \in W^r$: *a reply from i was received by time t*$\}$
  $\forall i \in R : reply\_select_i \leftarrow reply\_select_i + 1$
  update *responsiveness reputation* ρ_{rs_i} *of each worker* $i \in W^r$
  audit *the received answers with probability* $p_{\mathcal{A}}$
  if *the answers were not audited* **then**
    accept *the value m returned by workers* $R_m \subseteq R$,
      where $\forall m', \rho_{tr_{R_m}} \geq \rho_{tr_{R_{m'}}}$ // *weighted majority of workers in R*
  else // *the master audits*
    foreach $i \in R$ **do**
      $audit\_reply\_select_i \leftarrow audit\_reply\_select_i + 1$
      if $i \in F$ **then** $streak_i \leftarrow 0$ // $F \subseteq R$ *is the set of responsive workers caught cheating*
      else $correct\_audit_i \leftarrow correct\_audit_i + 1$, $streak_i \leftarrow streak_i + 1$ // *honest responsive*
      update *truthfulness reputation* ρ_{tr_i} // *depending on the type used*
    if $\rho_{tr_R} = 0$ **then** $p_{\mathcal{A}} \leftarrow \min\{1, p_{\mathcal{A}} + \alpha_m\}$
    else
      $p'_{\mathcal{A}} \leftarrow p_{\mathcal{A}} + \alpha_m(\rho_{tr_F}/\rho_{tr_R} - \tau)$
      $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p'_{\mathcal{A}}\}\}$
  $\forall i \in W^r$: **return** Π_i *to worker i* // *the payoff of workers in* $W^r \setminus R$ *is zero*

Protocol

update responsiveness reputation

Algorithm 1 Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$
for $i \leftarrow 0$ *to* N **do**
    $select_i \leftarrow 0$; $reply\_select_i \leftarrow 0$; $audit\_reply\_select_i \leftarrow 0$; $correct\_audit_i \leftarrow 0$; $streak_i \leftarrow 0$
    $\rho_{rs_i} \leftarrow 1$; *initialize* ρ_{tr_i} *// initially all workers have the same reputation*
for $r \leftarrow 1$ *to* ∞ **do**
    $W^r \leftarrow \{i \in \mathcal{N}$: i *is chosen as one of the* n *workers with the highest* $\rho_i = \rho_{rs_i} \cdot \rho_{tr_i}$ $\}$
    $\forall i \in W^r : select_i \leftarrow select_i + 1$
    send *a task* T *to all workers in* W^r
    collect replies from workers in W^r **for** t *time*
    wait for t *time collecting replies as received from workers in* W^r
    $R \leftarrow \{i \in W^r$: *a reply from* i *was received by time* $t\}$
    $\forall i \in R : reply\_select_i \leftarrow reply\_select_i + 1$
    update *responsiveness reputation* ρ_{rs_i} *of each worker* $i \in W^r$
    audit *the received answers with probability* $p_{\mathcal{A}}$
    if *the answers were not audited* **then**
        accept *the value* m *returned by workers* $R_m \subseteq R$,
            where $\forall m', \rho_{tr_{R_m}} \geq \rho_{tr_{R_{m'}}}$ *// weighted majority of workers in* R
    else *// the master audits*
        foreach $i \in R$ **do**
            $audit\_reply\_select_i \leftarrow audit\_reply\_select_i + 1$
            if $i \in F$ **then** $streak_i \leftarrow 0$ *//* $F \subseteq R$ *is the set of responsive workers caught cheating*
            else $correct\_audit_i \leftarrow correct\_audit_i + 1$, $streak_i \leftarrow streak_i + 1$ *// honest responsive*
            update *truthfulness reputation* ρ_{tr_i} *// depending on the type used*
        if $\rho_{tr_R} = 0$ **then** $p_{\mathcal{A}} \leftarrow \min\{1, p_{\mathcal{A}} + \alpha_m\}$
        else
            $p'_{\mathcal{A}} \leftarrow p_{\mathcal{A}} + \alpha_m(\rho_{tr_F}/\rho_{tr_R} - \tau)$
            $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p'_{\mathcal{A}}\}\}$
    $\forall i \in W^r$: **return** Π_i *to worker* i *// the payoff of workers in* $W^r \setminus R$ *is zero*

Protocol

Algorithm 1 Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$
for $i \leftarrow 0$ *to* N **do**
    $select_i \leftarrow 0$; $reply\_select_i \leftarrow 0$; $audit\_reply\_select_i \leftarrow 0$; $correct\_audit_i \leftarrow 0$; $streak_i \leftarrow 0$
    $\rho_{rs_i} \leftarrow 1$; *initialize* ρ_{tr_i} *// initially all workers have the same reputation*
for $r \leftarrow 1$ *to* ∞ **do**
    $W^r \leftarrow \{i \in \mathcal{N}$: *i is chosen as one of the* n *workers with the highest* $\rho_i = \rho_{rs_i} \cdot \rho_{tr_i}$ $\}$
    $\forall i \in W^r : select_i \leftarrow select_i + 1$
    send *a task* T *to all workers in* W^r
    collect replies from workers in W^r **for** t *time*
    wait for t *time collecting replies as received from workers in* W^r
    $R \leftarrow \{i \in W^r$: *a reply from* i *was received by time* $t\}$
    $\forall i \in R : reply\_select_i \leftarrow reply\_select_i + 1$
    update *responsiveness reputation* ρ_{rs_i} *of each worker* $i \in W^r$
    audit *the received answers with probability* $p_{\mathcal{A}}$
    if *the answers were not audited* **then**
        accept *the value* m *returned by workers* $R_m \subseteq R$,
            where $\forall m', \rho_{tr_{R_m}} \geq \rho_{tr_{R_{m'}}}$ *// weighted majority of workers in* R
    else *// the master audits*
        foreach $i \in R$ **do**
            $audit\_reply\_select_i \leftarrow audit\_reply\_select_i + 1$
            if $i \in F$ **then** $streak_i \leftarrow 0$ *//* $F \subseteq R$ *is the set of responsive workers caught cheating*
            else $correct\_audit_i \leftarrow correct\_audit_i + 1$, $streak_i \leftarrow streak_i + 1$ *// honest responsive*
            update *truthfulness reputation* ρ_{tr_i} *// depending on the type used*
        if $\rho_{tr_R} = 0$ **then** $p_{\mathcal{A}} \leftarrow \min\{1, p_{\mathcal{A}} + \alpha_m\}$
        else
            $p'_{\mathcal{A}} \leftarrow p_{\mathcal{A}} + \alpha_m(\rho_{tr_F}/\rho_{tr_R} - \tau)$
            $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p'_{\mathcal{A}}\}\}$
    $\forall i \in W^r$: **return** Π_i *to worker* i *// the payoff of workers in* $W^r \setminus R$ *is zero*

audit with
probability pA

Protocol

Algorithm 1 Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$
for $i \leftarrow 0$ *to* N **do**
    $select_i \leftarrow 0; reply\_select_i \leftarrow 0; audit\_reply\_select_i \leftarrow 0; correct\_audit_i \leftarrow 0; streak_i \leftarrow 0$
    $\rho_{rs_i} \leftarrow 1$; *initialize* ρ_{tr_i} *// initially all workers have the same reputation*
for $r \leftarrow 1$ *to* ∞ **do**
    $W^r \leftarrow \{i \in \mathcal{N}$: i *is chosen as one of the* n *workers with the highest* $\rho_i = \rho_{rs_i} \cdot \rho_{tr_i}\}$
    $\forall i \in W^r : select_i \leftarrow select_i + 1$
    send *a task* T *to all workers in* W^r
    collect replies from workers in W^r **for** t *time*
    wait for t *time collecting replies as received from workers in* W^r
    $R \leftarrow \{i \in W^r$: *a reply from* i *was received by time* $t\}$
    $\forall i \in R : reply\_select_i \leftarrow reply\_select_i + 1$
    update *responsiveness reputation* ρ_{rs_i} *of each worker* $i \in W^r$
    audit *the received answers with probability* $p_{\mathcal{A}}$
    if *the answers were not audited* **then**
        accept *the value* m *returned by workers* $R_m \subseteq R$,
            where $\forall m', \rho_{tr_{R_m}} \geq \rho_{tr_{R_{m'}}}$ *// weighted majority of workers in* R
    else *// the master audits*
        foreach $i \in R$ **do**
            $audit\_reply\_select_i \leftarrow audit\_reply\_select_i + 1$
            if $i \in F$ **then** $streak_i \leftarrow 0$ *//* $F \subseteq R$ *is the set of responsive workers caught cheating*
            else $correct\_audit_i \leftarrow correct\_audit_i + 1, streak_i \leftarrow streak_i + 1$ *// honest responsive*
            update *truthfulness reputation* ρ_{tr_i} *// depending on the type used*
        if $\rho_{tr_R} = 0$ **then** $p_{\mathcal{A}} \leftarrow \min\{1, p_{\mathcal{A}} + \alpha_m\}$
        else
            $p'_{\mathcal{A}} \leftarrow p_{\mathcal{A}} + \alpha_m(\rho_{tr_F}/\rho_{tr_R} - \tau)$
            $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p'_{\mathcal{A}}\}\}$
    $\forall i \in W^r$: **return** Π_i *to worker* i *// the payoff of workers in* $W^r \setminus R$ *is zero*

if NOT audited,
accept majority
weighted by
truthfulness rep

Protocol

Algorithm 1 Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$
for $i \leftarrow 0$ *to* N **do**
  $select_i \leftarrow 0$; $reply\_select_i \leftarrow 0$; $audit\_reply\_select_i \leftarrow 0$; $correct\_audit_i \leftarrow 0$; $streak_i \leftarrow 0$
  $\rho_{rs_i} \leftarrow 1$; *initialize* ρ_{tr_i} *// initially all workers have the same reputation*
for $r \leftarrow 1$ *to* ∞ **do**
  $W^r \leftarrow \{i \in \mathcal{N}$: i *is chosen as one of the* n *workers with the highest* $\rho_i = \rho_{rs_i} \cdot \rho_{tr_i}\}$
  $\forall i \in W^r : select_i \leftarrow select_i + 1$
  send *a task* T *to all workers in* W^r
  collect replies from workers in W^r **for** t *time*
  wait for t *time collecting replies as received from workers in* W^r
  $R \leftarrow \{i \in W^r$: *a reply from* i *was received by time* $t\}$
  $\forall i \in R : reply\_select_i \leftarrow reply\_select_i + 1$
  update *responsiveness reputation* ρ_{rs_i} *of each worker* $i \in W^r$
  audit *the received answers with probability* $p_{\mathcal{A}}$
  if *the answers were not audited* **then**
    accept *the value* m *returned by workers* $R_m \subseteq R$,
      where $\forall m', \rho_{tr_{R_m}} \geq \rho_{tr_{R_{m'}}}$ *// weighted majority of workers in* R
  else *// the master audits*
    foreach $i \in R$ **do**
      $audit\_reply\_select_i \leftarrow audit\_reply\_select_i + 1$
      if $i \in F$ **then** $streak_i \leftarrow 0$ *//* $F \subseteq R$ *is the set of responsive workers caught cheating*
      else $correct\_audit_i \leftarrow correct\_audit_i + 1$, $streak_i \leftarrow streak_i + 1$ *// honest responsive*
      update *truthfulness reputation* ρ_{tr_i} *// depending on the type used*
    if $\rho_{tr_R} = 0$ **then** $p_{\mathcal{A}} \leftarrow \min\{1, p_{\mathcal{A}} + \alpha_m\}$
    else
      $p'_{\mathcal{A}} \leftarrow p_{\mathcal{A}} + \alpha_m(\rho_{tr_F}/\rho_{tr_R} - \tau)$
      $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p'_{\mathcal{A}}\}\}$
  $\forall i \in W^r$: **return** Π_i *to worker* i *// the payoff of workers in* $W^r \setminus R$ *is zero*

if audited,
update truthfulness
reputation

Protocol

Algorithm 1 Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$
for $i \leftarrow 0$ *to* N **do**
    $select_i \leftarrow 0; reply\_select_i \leftarrow 0; audit\_reply\_select_i \leftarrow 0; correct\_audit_i \leftarrow 0; streak_i \leftarrow 0$
    $\rho_{rs_i} \leftarrow 1$; *initialize* ρ_{tr_i} // *initially all workers have the same reputation*
for $r \leftarrow 1$ *to* ∞ **do**
    $W^r \leftarrow \{i \in \mathcal{N}$: i *is chosen as one of the* n *workers with the highest* $\rho_i = \rho_{rs_i} \cdot \rho_{tr_i}$ $\}$
    $\forall i \in W^r : select_i \leftarrow select_i + 1$
    send *a task* T *to all workers in* W^r
    collect replies from workers in W^r **for** t *time*
    wait for t *time collecting replies as received from workers in* W^r
    $R \leftarrow \{i \in W^r$: *a reply from* i *was received by time* $t\}$
    $\forall i \in R : reply\_select_i \leftarrow reply\_select_i + 1$
    update *responsiveness reputation* ρ_{rs_i} *of each worker* $i \in W^r$
    audit *the received answers with probability* $p_{\mathcal{A}}$
    if *the answers were not audited* **then**
        accept *the value* m *returned by workers* $R_m \subseteq R$,
            where $\forall m', \rho_{tr_{R_m}} \geq \rho_{tr_{R_{m'}}}$ // *weighted majority of workers in* R
    else // *the master audits*
        foreach $i \in R$ **do**
            $audit\_reply\_select_i \leftarrow audit\_reply\_select_i + 1$
            if $i \in F$ **then** $streak_i \leftarrow 0$ // $F \subseteq R$ *is the set of responsive workers caught cheating*
            else $correct\_audit_i \leftarrow correct\_audit_i + 1$, $streak_i \leftarrow streak_i + 1$ // *honest responsive*
            update *truthfulness reputation* ρ_{tr_i} // *depending on the type used*
        if $\rho_{tr_R} = 0$ **then** $p_{\mathcal{A}} \leftarrow \min\{1, p_{\mathcal{A}} + \alpha_m\}$
        else
            $p'_{\mathcal{A}} \leftarrow p_{\mathcal{A}} + \alpha_m(\rho_{tr_F}/\rho_{tr_R} - \tau)$
            $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p'_{\mathcal{A}}\}\}$
    $\forall i \in W^r$: **return** Π_i *to worker* i // *the payoff of workers in* $W^r \setminus R$ *is zero*

... and pA

tolerance
to loss

Protocol

Algorithm 1 Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$

for $i \leftarrow 0$ *to* N **do**

  $select_i \leftarrow 0$; $reply\_select_i \leftarrow 0$; $audit\_reply\_select_i \leftarrow 0$; $correct\_audit_i \leftarrow 0$; $streak_i \leftarrow 0$

  $\rho_{rs_i} \leftarrow 1$; *initialize* ρ_{tr_i} *// initially all workers have the same reputation*

for $r \leftarrow 1$ *to* ∞ **do**

  $W^r \leftarrow \{i \in \mathcal{N} : i$ *is chosen as one of the* n *workers with the highest* $\rho_i = \rho_{rs_i} \cdot \rho_{tr_i}\}$

  $\forall i \in W^r : select_i \leftarrow select_i + 1$

  send *a task* T *to all workers in* W^r

  collect replies from workers in W^r **for** t *time*

  wait for t *time collecting replies as received from workers in* W^r

  $R \leftarrow \{i \in W^r :$ *a reply from* i *was received by time* $t\}$

  $\forall i \in R : reply\_select_i \leftarrow reply\_select_i + 1$

  update *responsiveness reputation* ρ_{rs_i} *of each worker* $i \in W^r$

  audit *the received answers with probability* $p_{\mathcal{A}}$

  if *the answers were not audited* **then**

    accept *the value* m *returned by workers* $R_m \subseteq R$,

      where $\forall m', \rho_{tr_{R_m}} \geq \rho_{tr_{R_{m'}}}$ *// weighted majority of workers in* R

  else *// the master audits*

    foreach $i \in R$ **do**

      $audit\_reply\_select_i \leftarrow audit\_reply\_select_i + 1$

      if $i \in F$ **then** $streak_i \leftarrow 0$ *//* $F \subseteq R$ *is the set of responsive workers caught cheating*

      else $correct\_audit_i \leftarrow correct\_audit_i + 1$, $streak_i \leftarrow streak_i + 1$ *// honest responsive*

      update *truthfulness reputation* ρ_{tr_i} *// depending on the type used*

    if $\rho_{tr_R} = 0$ **then** $p_{\mathcal{A}} \leftarrow \min\{1, p_{\mathcal{A}} + \alpha_m\}$

    else

      $p'_{\mathcal{A}} \leftarrow p_{\mathcal{A}} + \alpha_m(\rho_{tr_F}/\rho_{tr_R} - \tau)$

      $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p'_{\mathcal{A}}\}\}$

  $\forall i \in W^r$: **return** Π_i *to worker* i *// the payoff of workers in* $W^r \setminus R$ *is zero*

tolerance to loss

pay/penalize accordingly

Protocol

Algorithm 2 Algorithm for Rational Worker i

$p_{Ci} \leftarrow y$, *where* $y \in [0, 1]$
repeat forever
    wait for *a task* T *from the master*
    if *available* **then**
        decide whether to cheat or not independently with distribution $P(cheat) = p_{Ci}$
        if *the decision was to cheat* **then**
            send *arbitrary solution to the master*
            get payoff Π_i
            $p_{Ci} \leftarrow \max\{0, \min\{1, p_{Ci} + \alpha_w(\Pi_i - a_i)\}\}$
        else
            send $compute(T)$ *to the master*
            get payoff Π_i
            $p_{Ci} \leftarrow \max\{0, \min\{1, p_{Ci} - \alpha_w(\Pi_i - WC_{\mathcal{T}} - a_i)\}\}$

Protocol

cheat with probability pC

Algorithm 2 Algorithm for Rational Worker i

$p_{Ci} \leftarrow y$, *where* $y \in [0, 1]$
repeat forever
    wait for *a task* T *from the master*
    if *available* **then**
        decide whether to cheat or not independently with distribution $P(cheat) = p_{Ci}$
        if *the decision was to cheat* **then**
            send *arbitrary solution to the master*
            get payoff Π_i
            $p_{Ci} \leftarrow \max\{0, \min\{1, p_{Ci} + \alpha_w(\Pi_i - a_i)\}\}$
        else
            send $compute(T)$ *to the master*
            get payoff Π_i
            $p_{Ci} \leftarrow \max\{0, \min\{1, p_{Ci} - \alpha_w(\Pi_i - WC_{\mathcal{T}} - a_i)\}\}$

Protocol

Algorithm 2 Algorithm for Rational Worker i

$p_{Ci} \leftarrow y$, *where* $y \in [0, 1]$
repeat forever
  wait for *a task* T *from the master*
  if *available* **then**
    decide whether to cheat or not independently with distribution $P(cheat) = p_{Ci}$
    if *the decision was to cheat* **then**
      send *arbitrary solution to the master*
      get payoff Π_i
      $p_{Ci} \leftarrow \max\{0, \min\{1, p_{Ci} + \alpha_w(\Pi_i - a_i)\}\}$
    else
      send $compute(T)$ *to the master*
      get payoff Π_i
      $p_{Ci} \leftarrow \max\{0, \min\{1, p_{Ci} - \alpha_w(\Pi_i - WC_{\mathcal{T}} - a_i)\}\}$

cheat with probability pC

update pC

profit aspiration

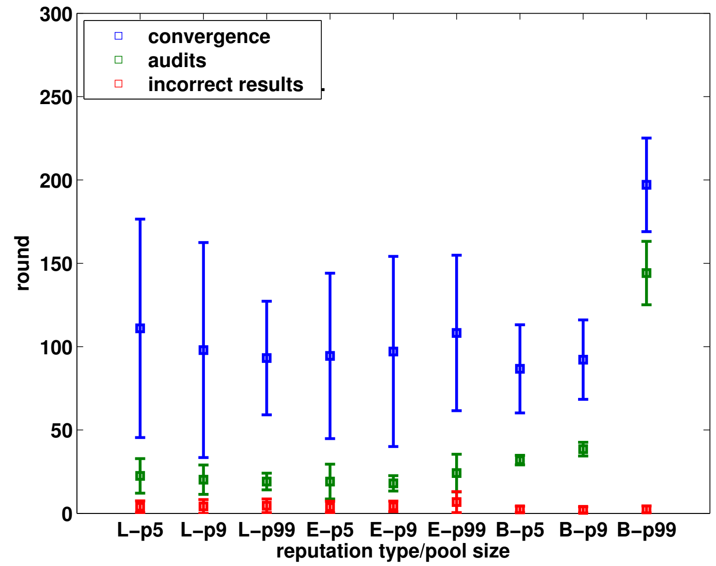
Feasibility without rationals

Theorem 3. *Consider a system in which workers are either altruistic or malicious and there is at least one altruistic worker i with $d_i = 1$ in the pool. Eventual correctness is satisfied if the mechanism of Algorithm 1 is used with the responsiveness reputation and any of the truthfulness reputations* LINEAR *or* EXPONENTIAL.

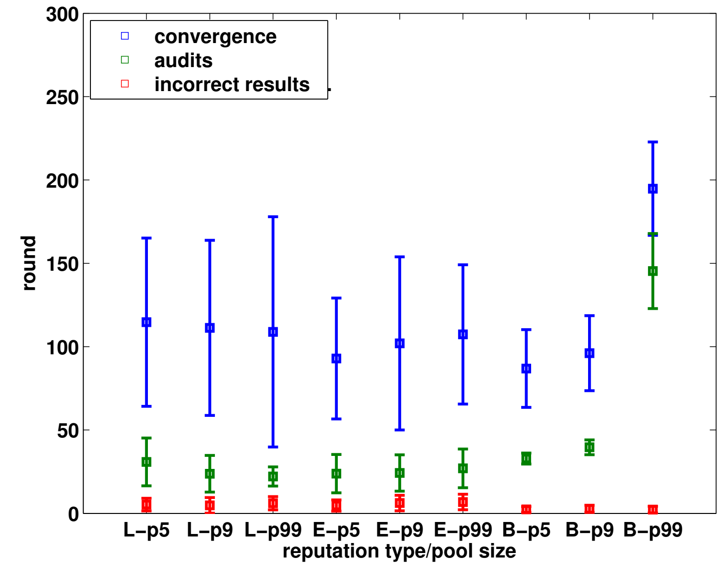
Theorem 4. *Consider a system in which workers are either altruistic or malicious and there is at least one altruistic worker i with $d_i = 1$ in the pool. In this system, the mechanism of Algorithm 1 is used with the responsiveness reputation and the truthfulness reputation* BOINC. *Then, eventual correctness is satisfied if and only if the number of altruistic workers with $d_j < 1$ is smaller than n.*

Simulations: workers always available (d=1)

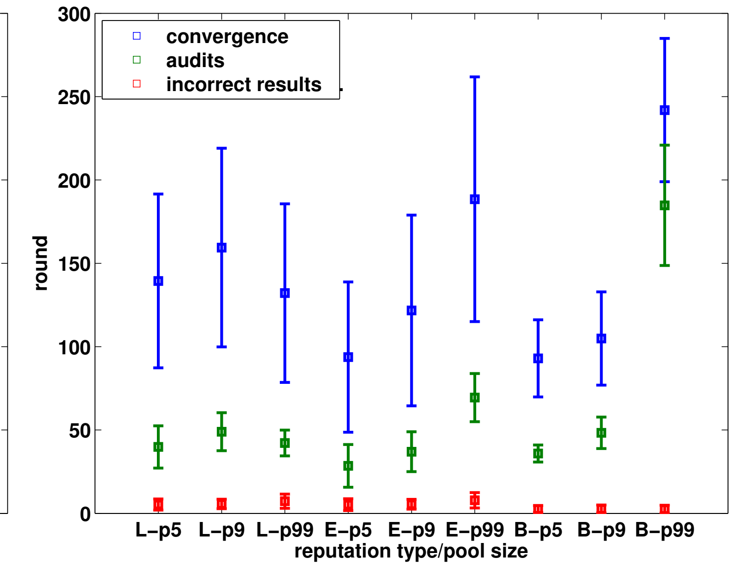
n=9, convergence: pA = pAmin = 0.01



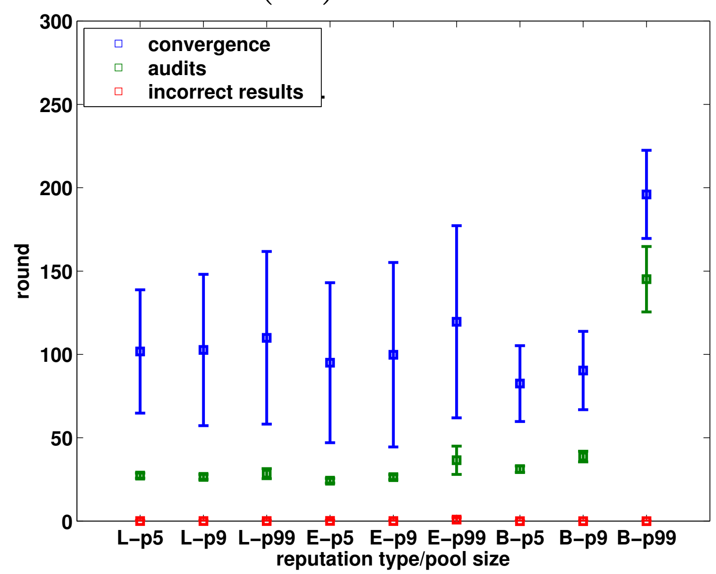
$(a1)$



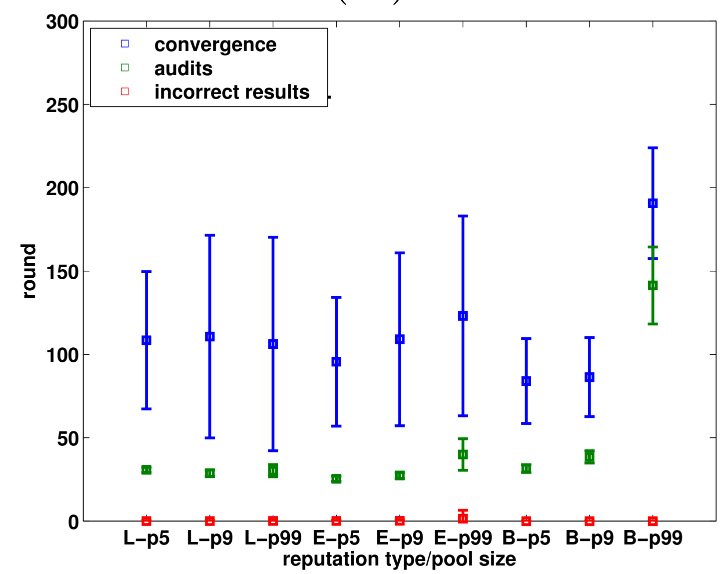
$(b1)$



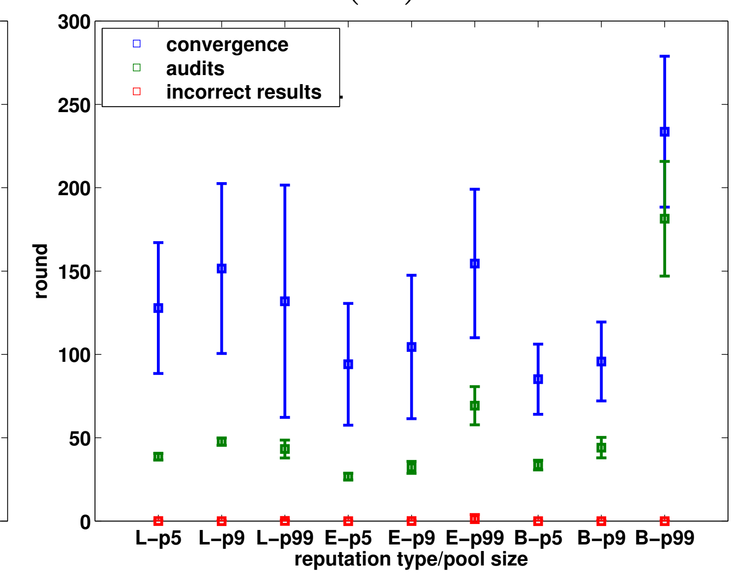
$(c1)$



$(a2)$



$(b2)$



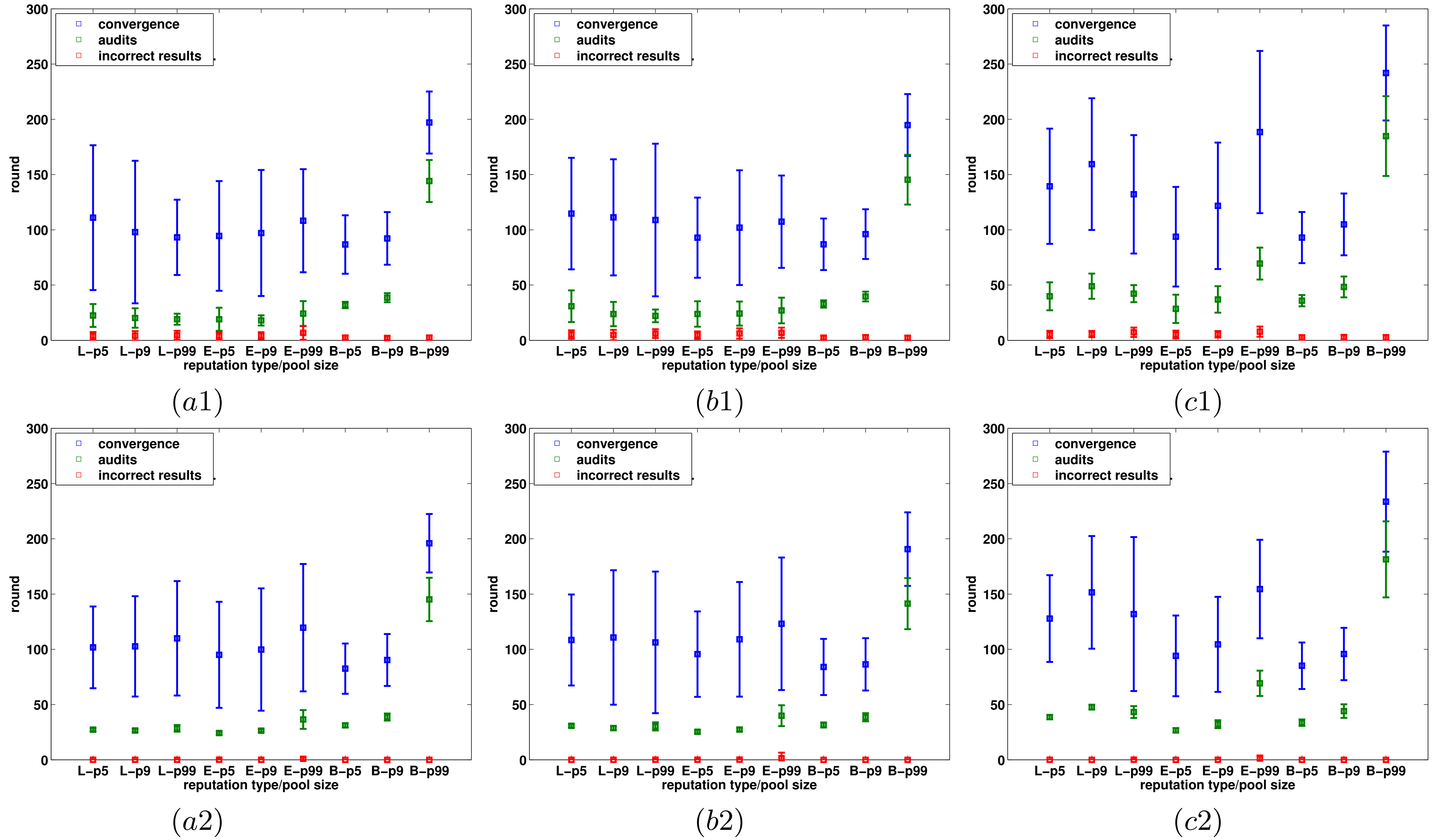
$(c2)$

Simulations: workers always available (d=1)

n=9, convergence: pA = pAmin = 0.01

initial pA=0.5

initial pA=1



rational / # malicious = 5/4 4/5 1/8

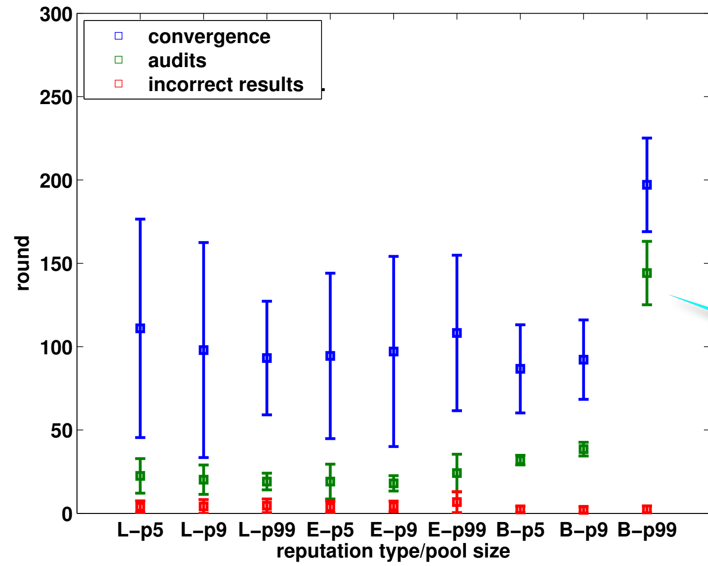
Simulations: workers always available (d=1)

n=9, convergence: pA = pAmin = 0.01

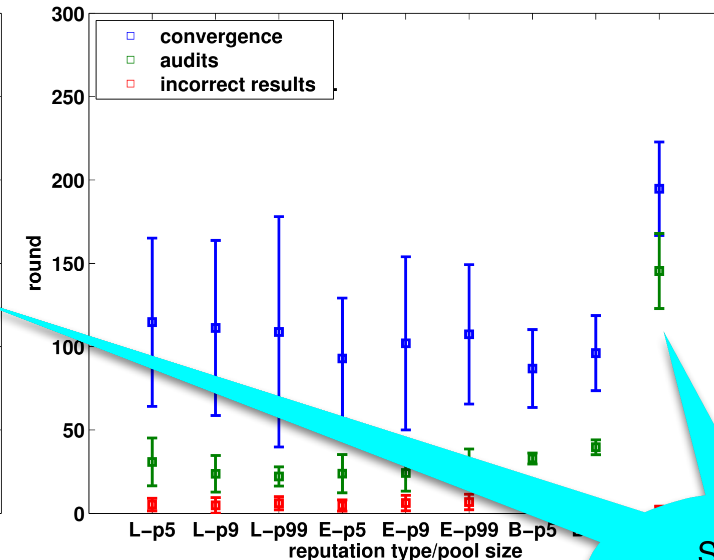
initial pA=0.5

initial pA=1

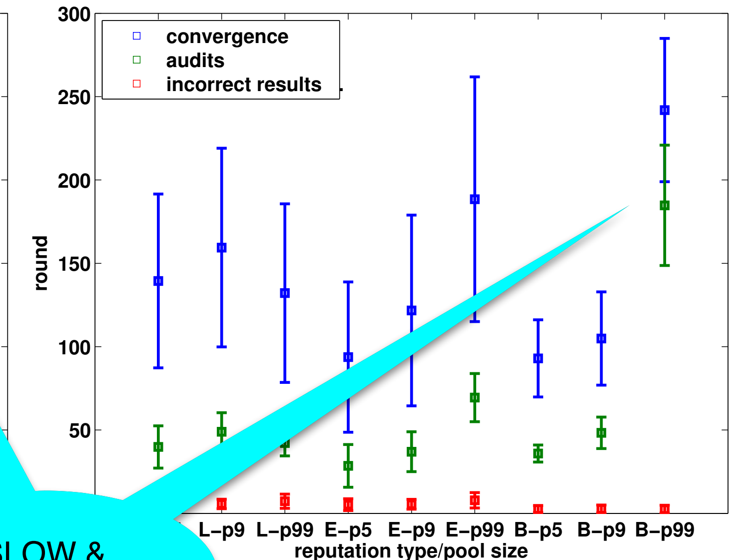
SLOW & EXPENSIVE



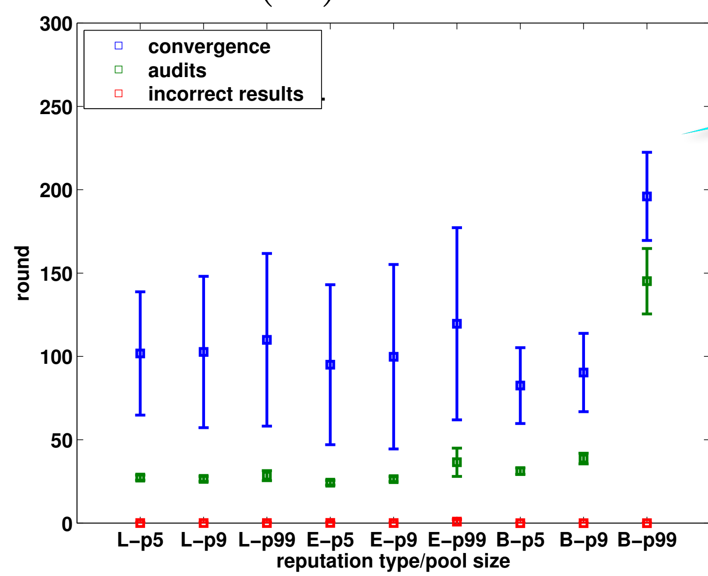
$(a1)$



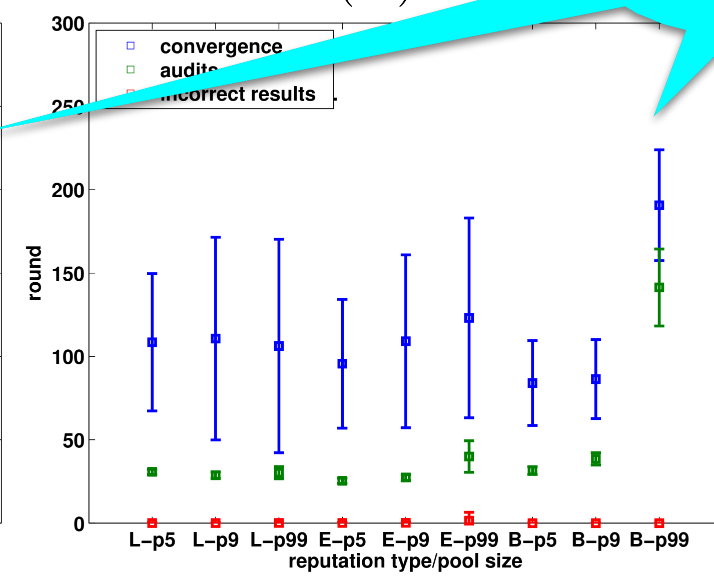
$(b1)$



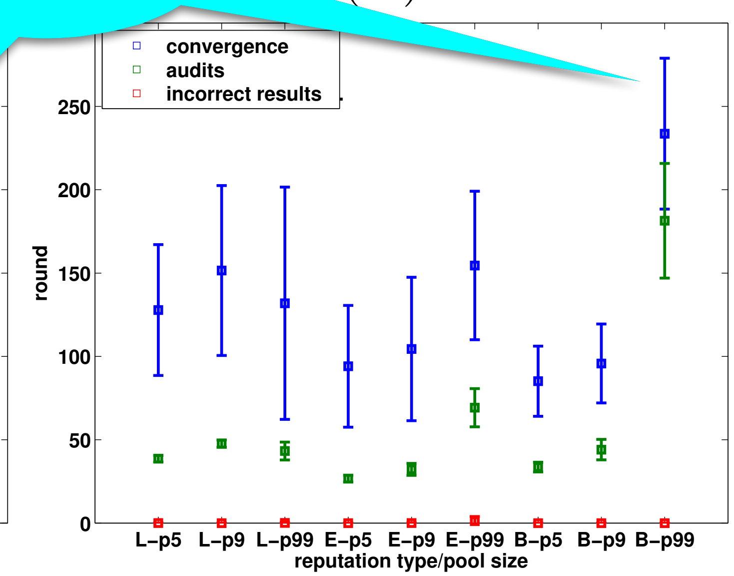
$(c1)$



$(a2)$



$(b2)$

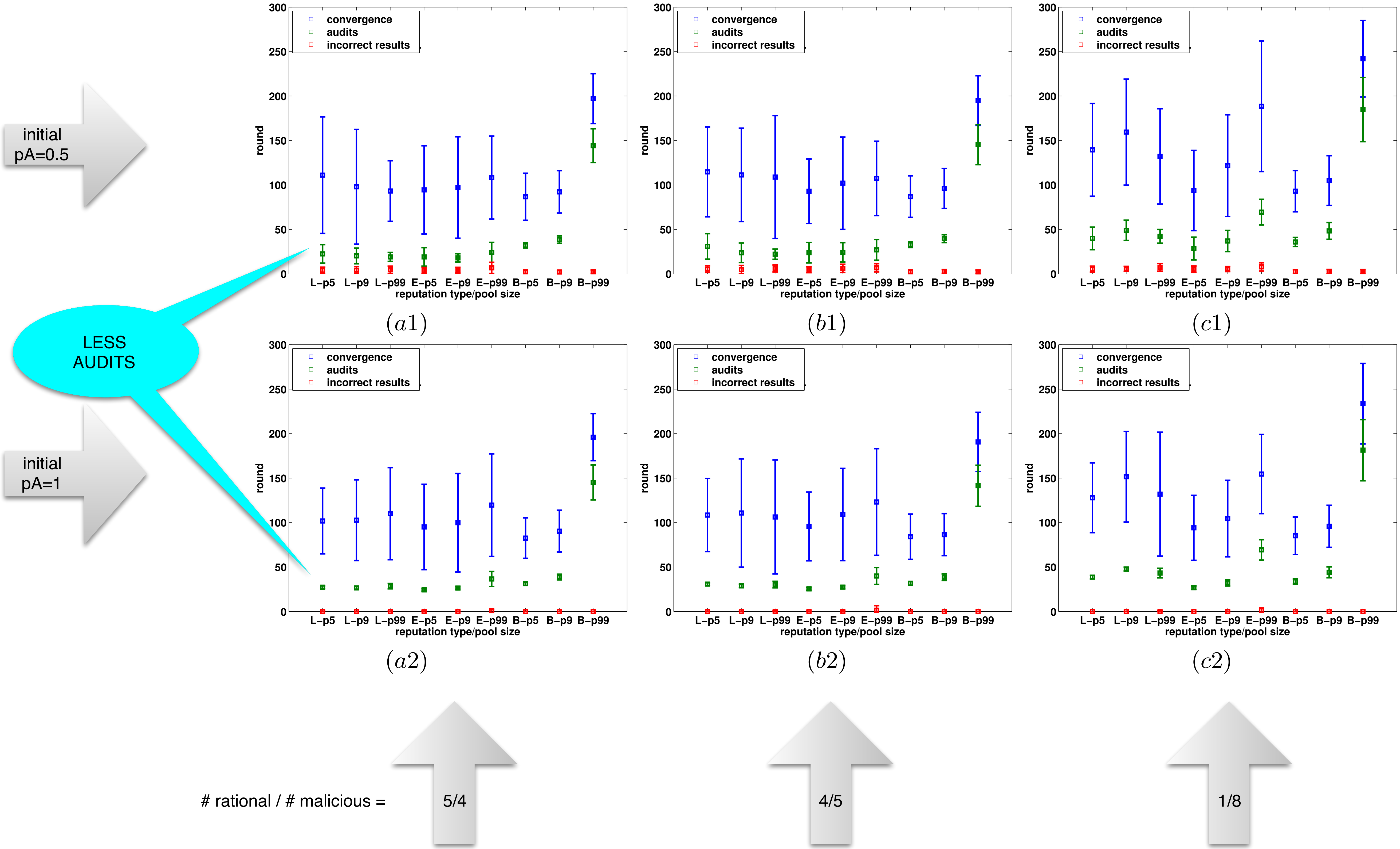


$(c2)$

rational / # malicious = 5/4 4/5 1/8

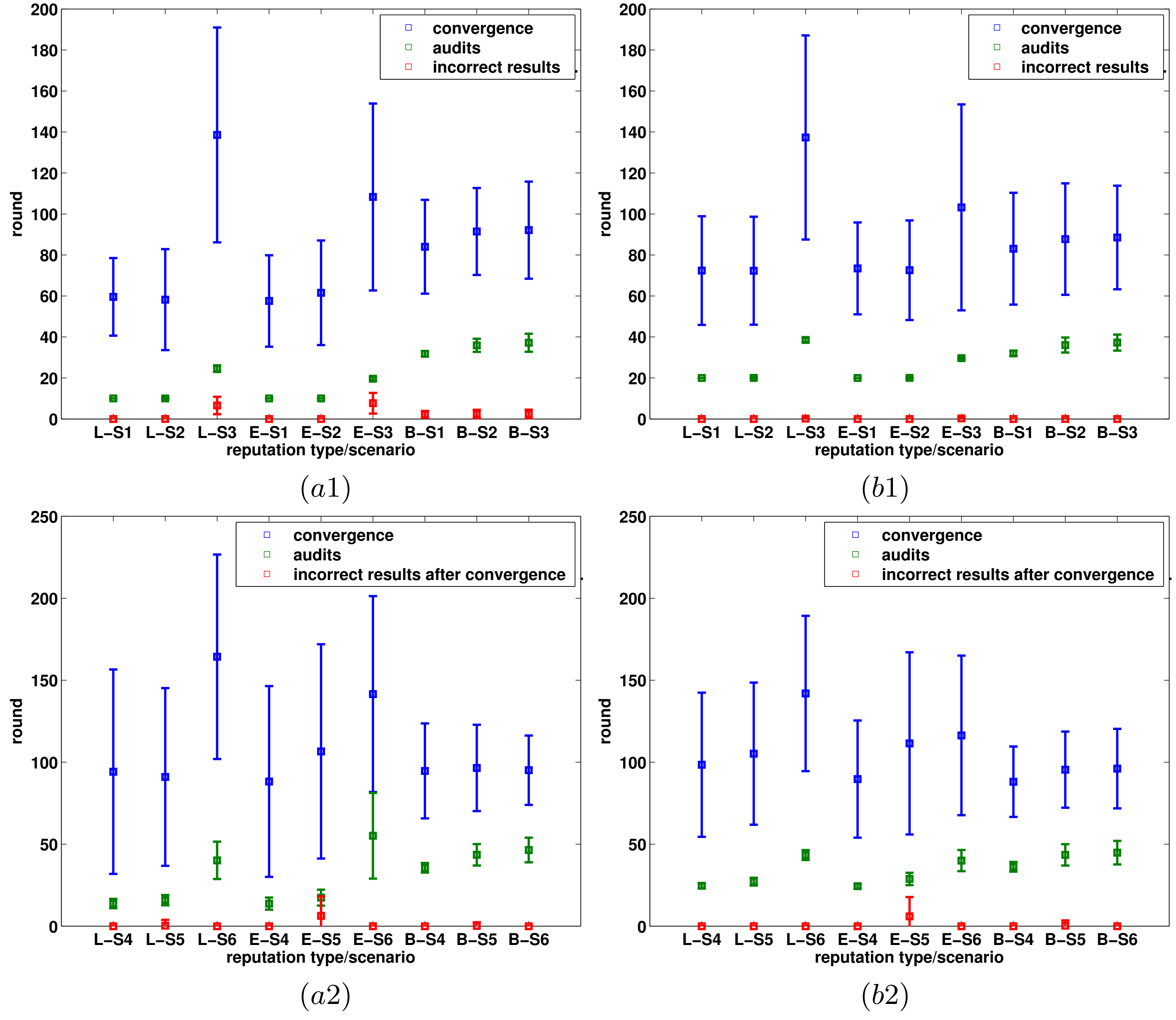
Simulations: workers always available (d=1)

n=9, convergence: pA = pAmin = 0.01



Simulations: workers partially available (d≤1)

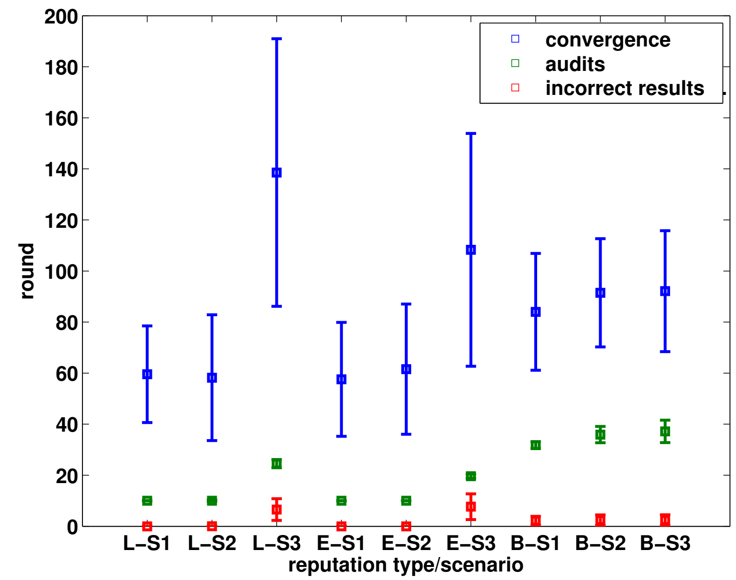
N=9, n=5, convergence: pA = pAmin = 0.01



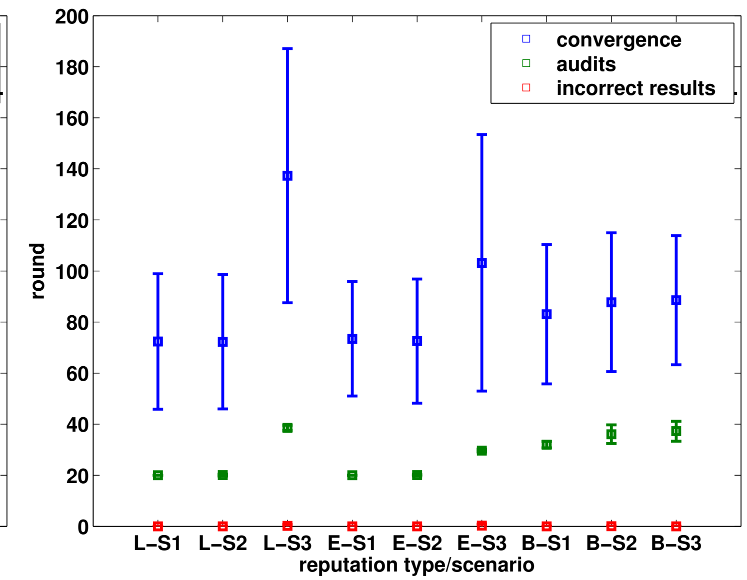
Simulations: workers partially available (d≤1)

N=9, n=5, convergence: pA = pAmin = 0.01

	M	A
S1		9)d=1
S2		1)d=1 8)d=0.5
S3	8)d=0.5	1)d=1

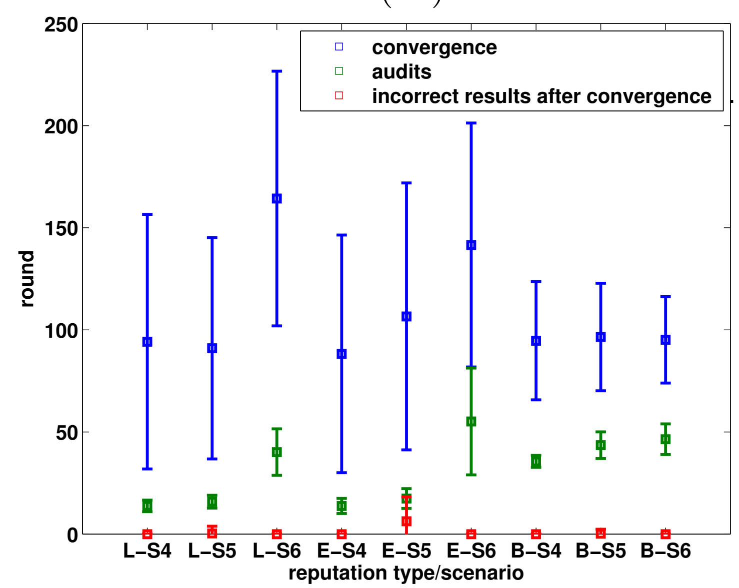


(a1)

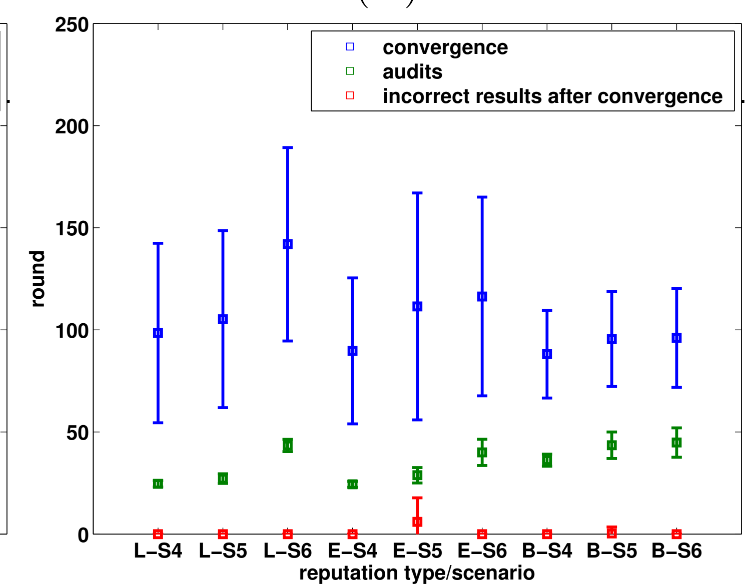


(b1)

	M	R
S4		9)d=1
S5		1)d=1 8)d=0.5
S6	8)d=0.5	1)d=1



(a2)



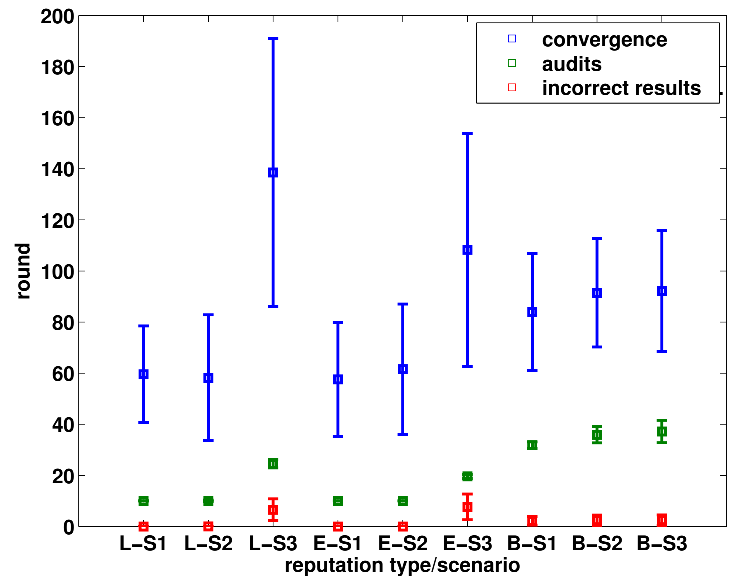
(b2)

initial pA = 0.5 | 1

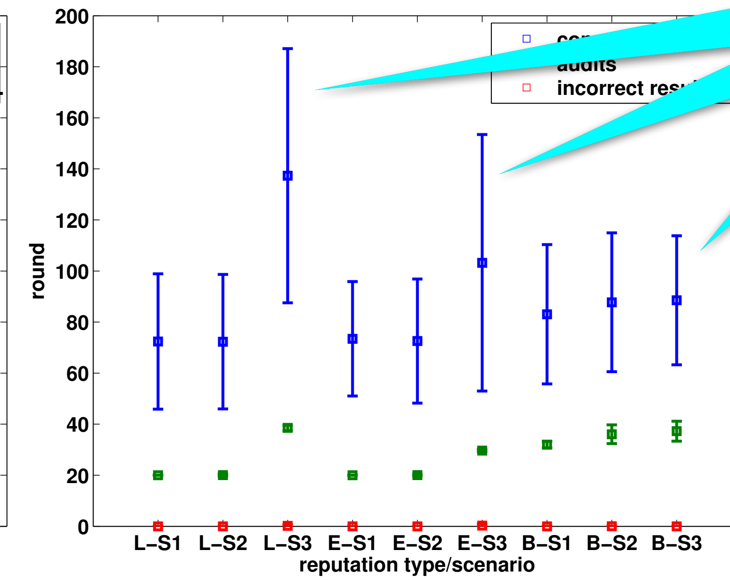
Simulations: workers partially available (d≤1)

N=9, n=5, convergence: pA = pAmin = 0.01

	M	A
S1		9)d=1
S2		1)d=1 8)d=0.5
S3	8)d=0.5	1)d=1



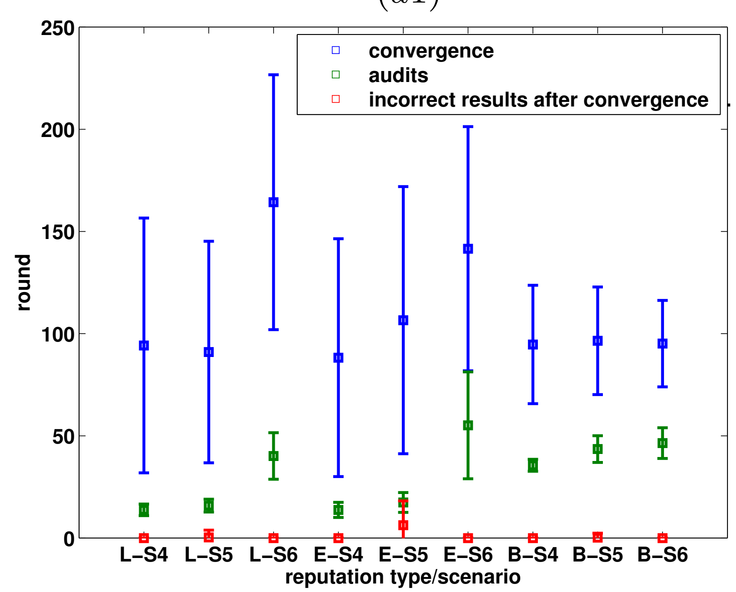
(a1)



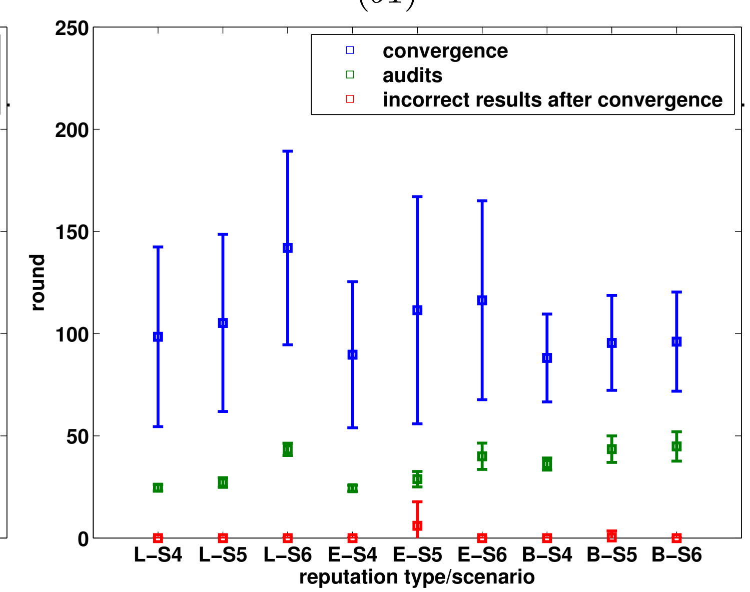
(b1)

WITH
LOTS OF MALICIOUS
BOINC IS BETTER

	M	R
S4		9)d=1
S5		1)d=1 8)d=0.5
S6	8)d=0.5	1)d=1



(a2)



(b2)

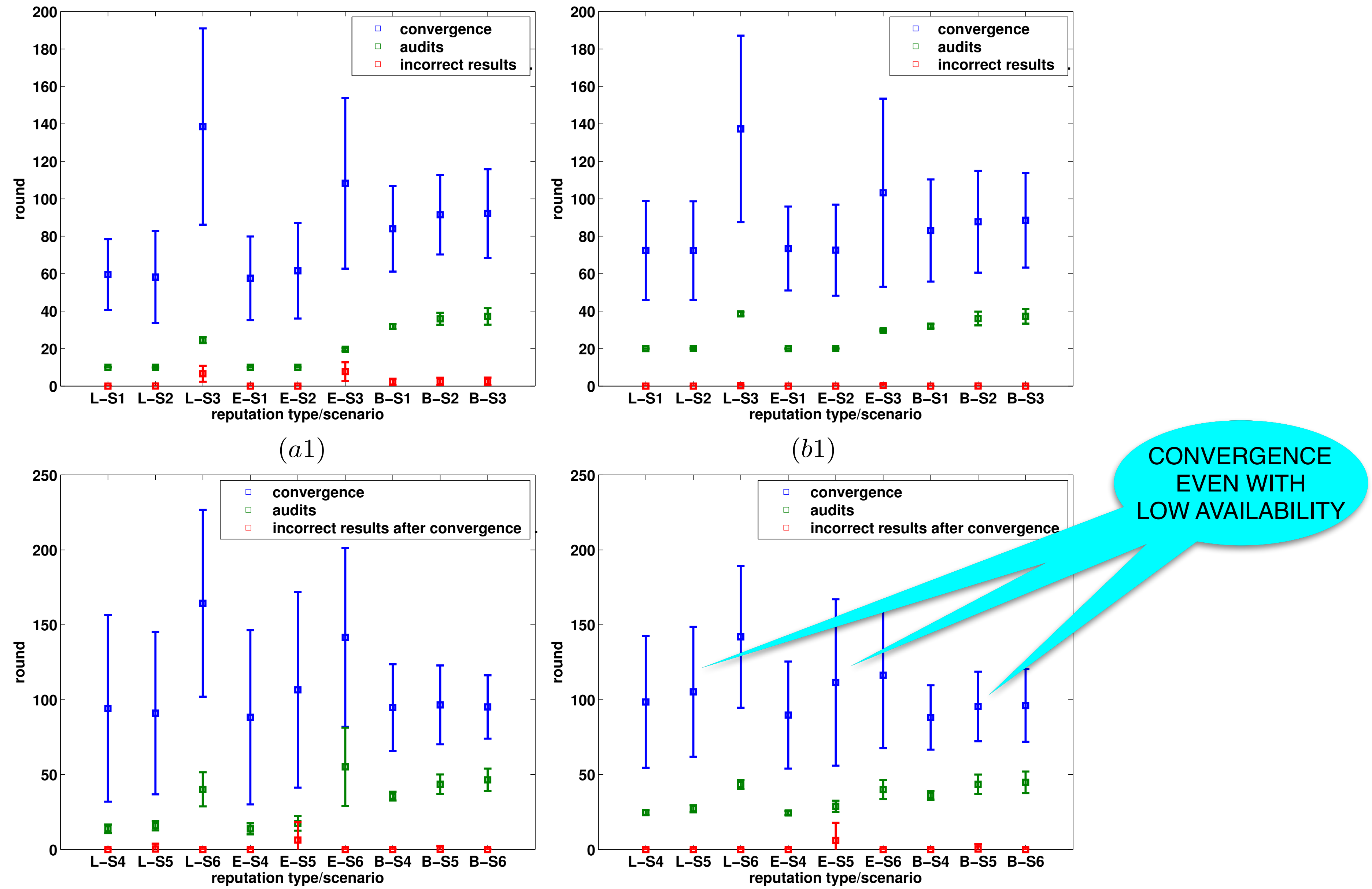
initial pA = 0.5 1

Simulations: workers partially available (d≤1)

N=9, n=5, convergence: pA = pAmin = 0.01

	M	A
S1		9)d=1
S2		1)d=1 8)d=0.5
S3	8)d=0.5	1)d=1

	M	R
S4		9)d=1
S5		1)d=1 8)d=0.5
S6	8)d=0.5	1)d=1



(a1) (b1)

(a2) (b2)

initial pA = 0.5 1

Ongoing and future work

- Application of repeated games framework for provable guarantees in multiround computations.
- Experimental comparison of both approaches.
- Integration of our mechanisms into emBoinc.

Thank you!