Leveraging Smart Phones to Reduce Mobility Footprints

GZ06 Mobile and Cloud Computing

Motivation

- mobile users
- users demand same desktop experience regardless of location
- how to store and replicate user's data and desktop
  - reliably?
  - securely?
  - maintaining a crisp, low-latency desktop experience?
  - coping with unreliable network connections?
  - without saddling the user with a high mobility footprint?

Mobility Footprint

Is defined as:

- size
- weight
- energy demand

of hardware necessary for an individual to carry in order to be effective on the go.

Two Extremes

"carry all the hardware you might need"

- *very low latency*
- *independence from network conditions*
- *high mobility footprint*

"carry nothing and live off the land"

- *low mobility footprint*
- *(potential) high latency*
- *more dependent on network conditions*
- *may not work without high-bandwidth connection*

Current Solutions

- local filesystem and user account (laptop)
- remote filesystem
- remote login / thin client (VNC/GoToMyPC)
- software/application virtualisation
- virtual machine on USB flash drive
- virtual machine downloaded over the Internet
  - e.g. *"Internet Suspend/Resume"* (ISR)

Current Solutions

- local filesystem and user account (laptop)
- remote filesystem
- remote login / thin client (VNC/GoToMyPC)
- software/application virtualisation
- virtual machine on USB flash drive
- virtual machine downloaded over the Internet
  - e.g. *"Internet Suspend/Resume"* (ISR)

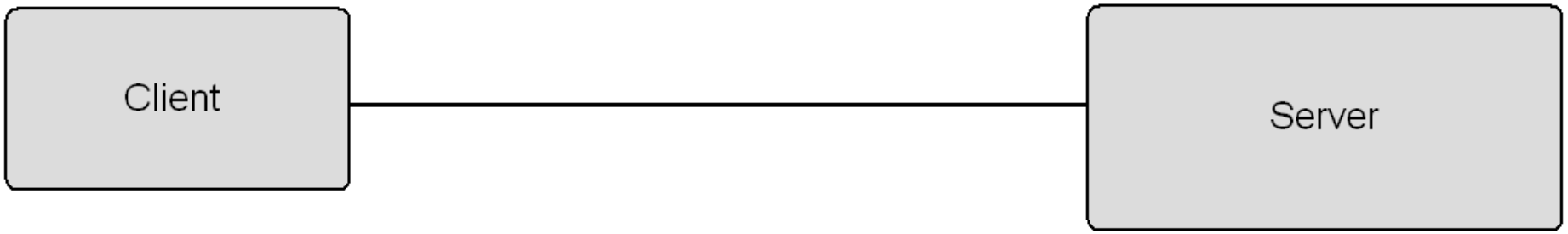
Current Solutions

- local filesystem and user account (laptop)
- remote filesystem
- remote login / thin client (VNC/GoToMyPC)
- software/application virtualisation
- virtual machine on USB flash drive
- virtual machine downloaded over the Internet
  - e.g. *"Internet Suspend/Resume"* (ISR)

Current Solutions

- local filesystem and user account (laptop)
- remote filesystem
- remote login / thin client (VNC/GoToMyPC)
- software/application virtualisation
- virtual machine on USB flash drive
- virtual machine downloaded over the Internet
  - e.g. *"Internet Suspend/Resume"* (ISR)

Traditional Client-Server Architecture



Requires a reliable, high-bandwidth network connection, especially as the client becomes "thinner".

"Thinner" client:

- lower mobility footprint
- better connection required

"Fatter" client:

- higher mobility footprint
- worse connection usable

Horatio

Allows trade-off between "fat" and "thin" clients to be overcome:

- low mobility footprint
- worse connections usable

Horatio

Allows trade-off between "fat" and "thin" clients to be overcome:

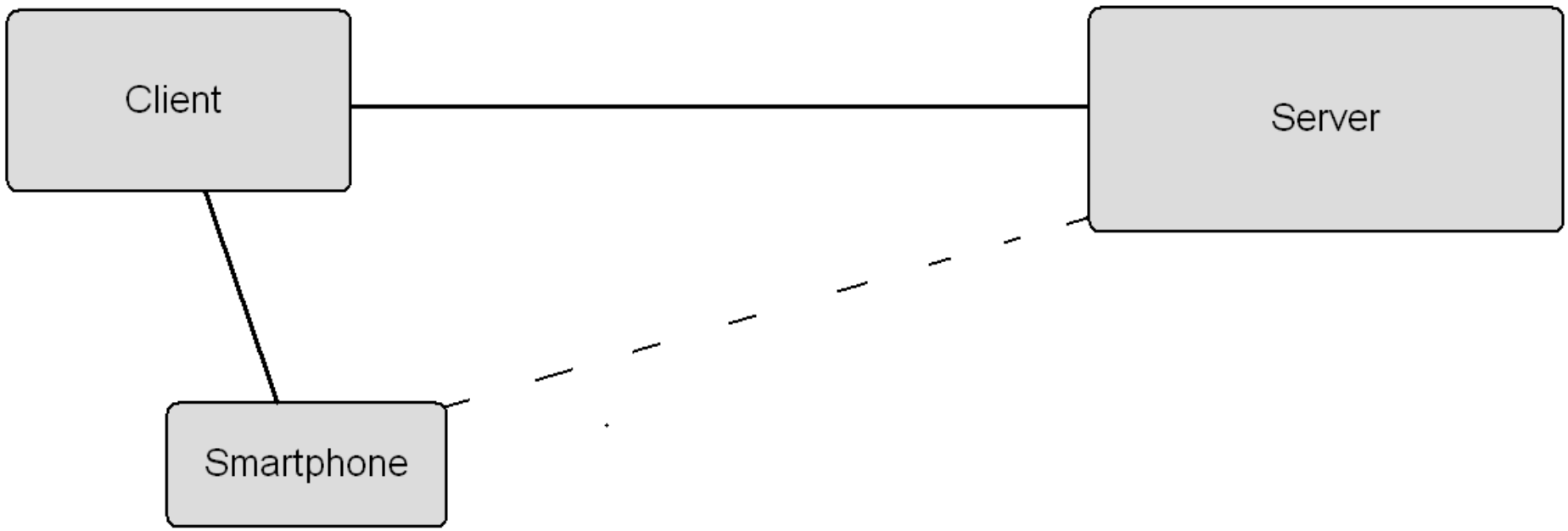
- low mobility footprint
- worse connections usable
  - uses transient high-quality connection to sync with server
  - no "always-on" connection required
  - uses the user's smartphone as a cache when connection unavailable

Goals of Horatio

To *"alleviate...ISR limitations"* and reduce:

- *"resume latency"* - large download to client at startup
- *"slowdown"* - parcels transferred to server during a session
- *"suspend latency"* - remaining parcels of modified VM state uploaded to server at shutdown

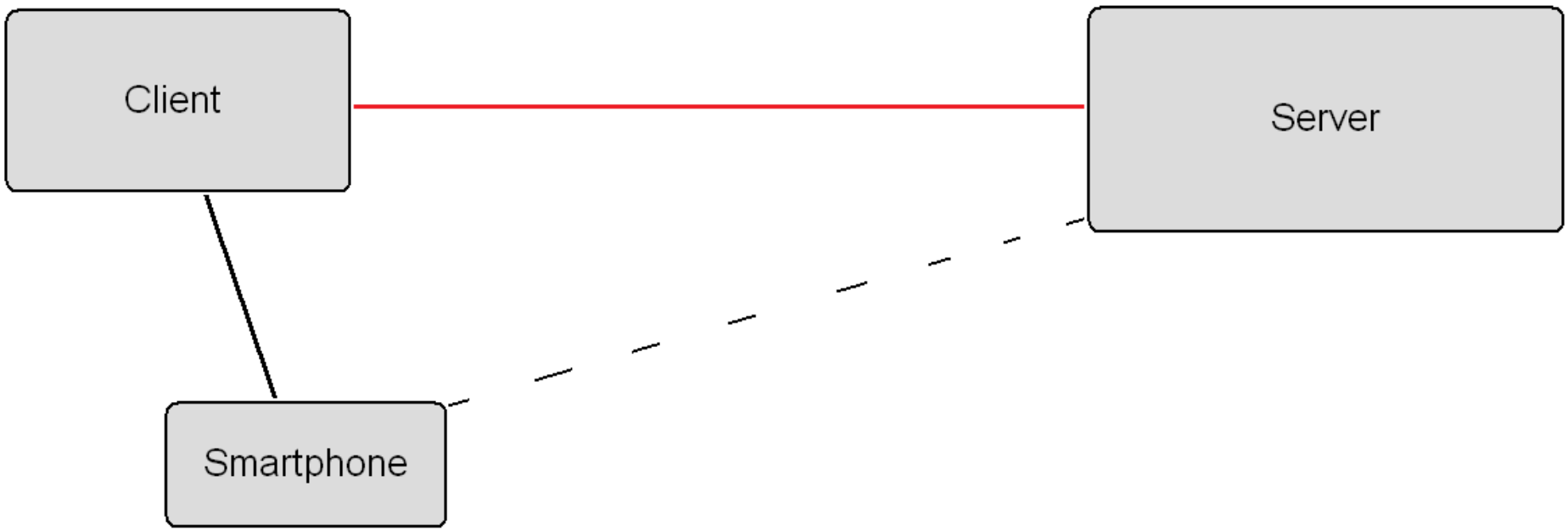
Horatio Overview



- Add user's smartphone in between client and server
- Smartphone always local to user
- Smartphone caches client's interactions with server
- Allows good performance & user expereince with low-quality or non-existent network connections

Horatio Overview - Data Links

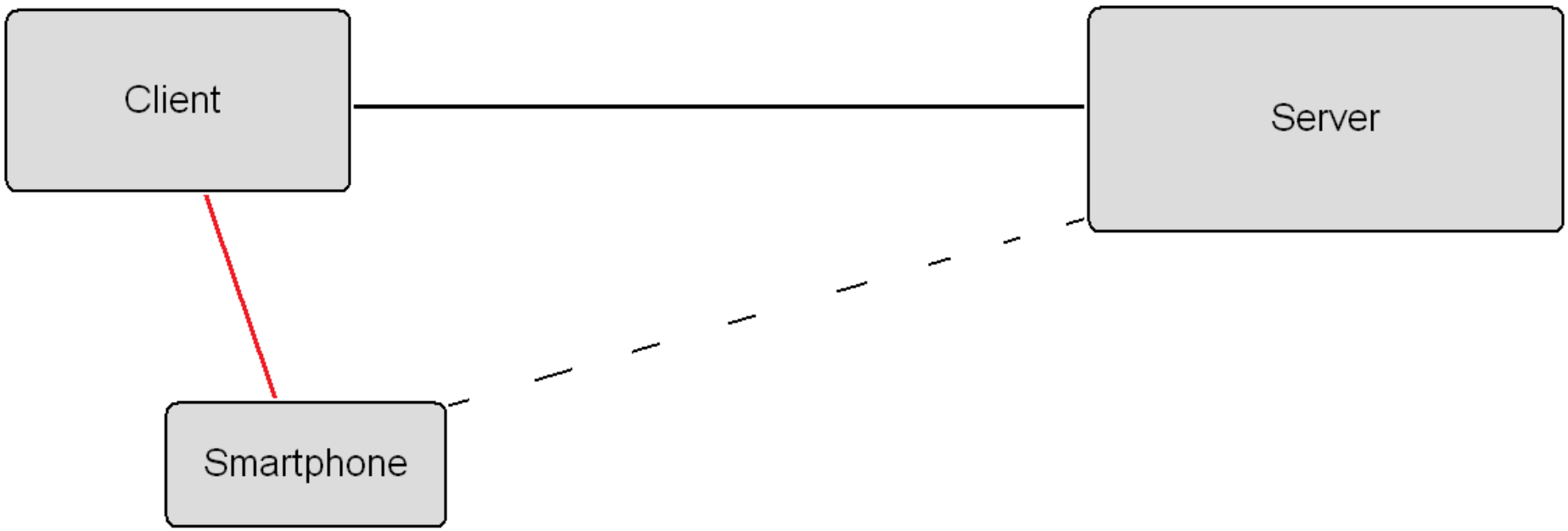
Client-Server



- High-bandwidth, unreliable data link
- Link "unreliable" because client travelling between "oases of connectivity"
  - e.g. client boots with wired Internet connection in hotel room, then travels to area with no connection whatsoever

Horatio Overview - Data Links

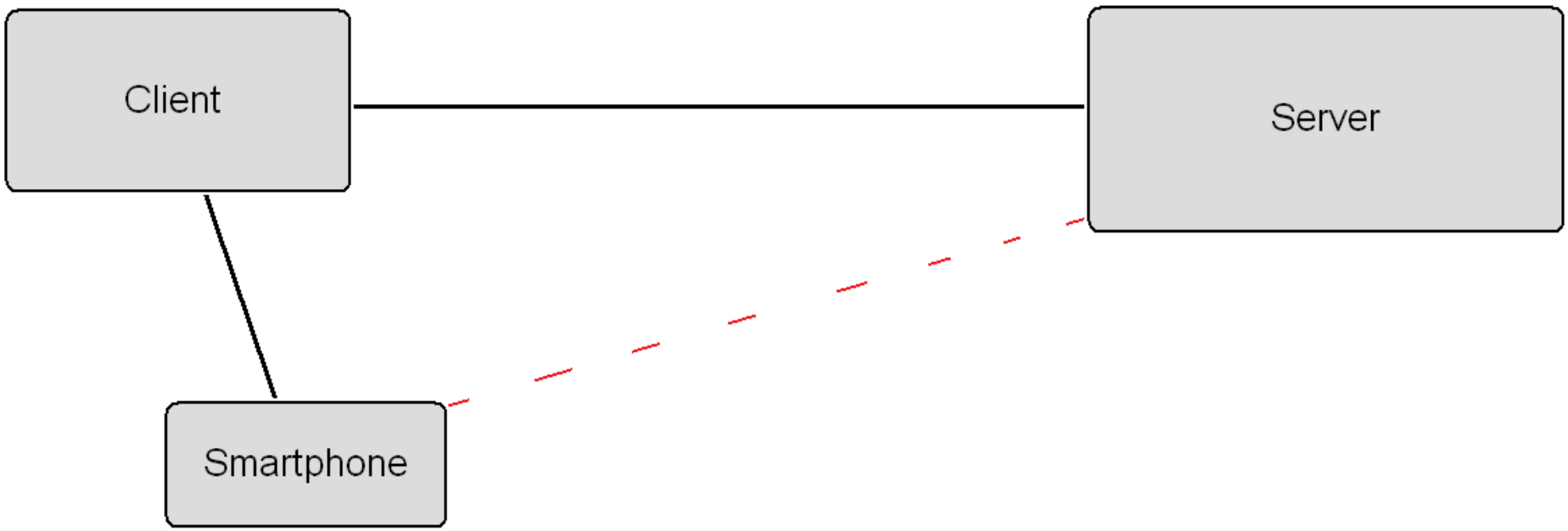
Client-Smartphone



- High-bandwidth, reliable data link
  - e.g. USB
- Client and smartphone assumed to be in close proximity

Horatio Overview - Data Links

Smartphone-Server

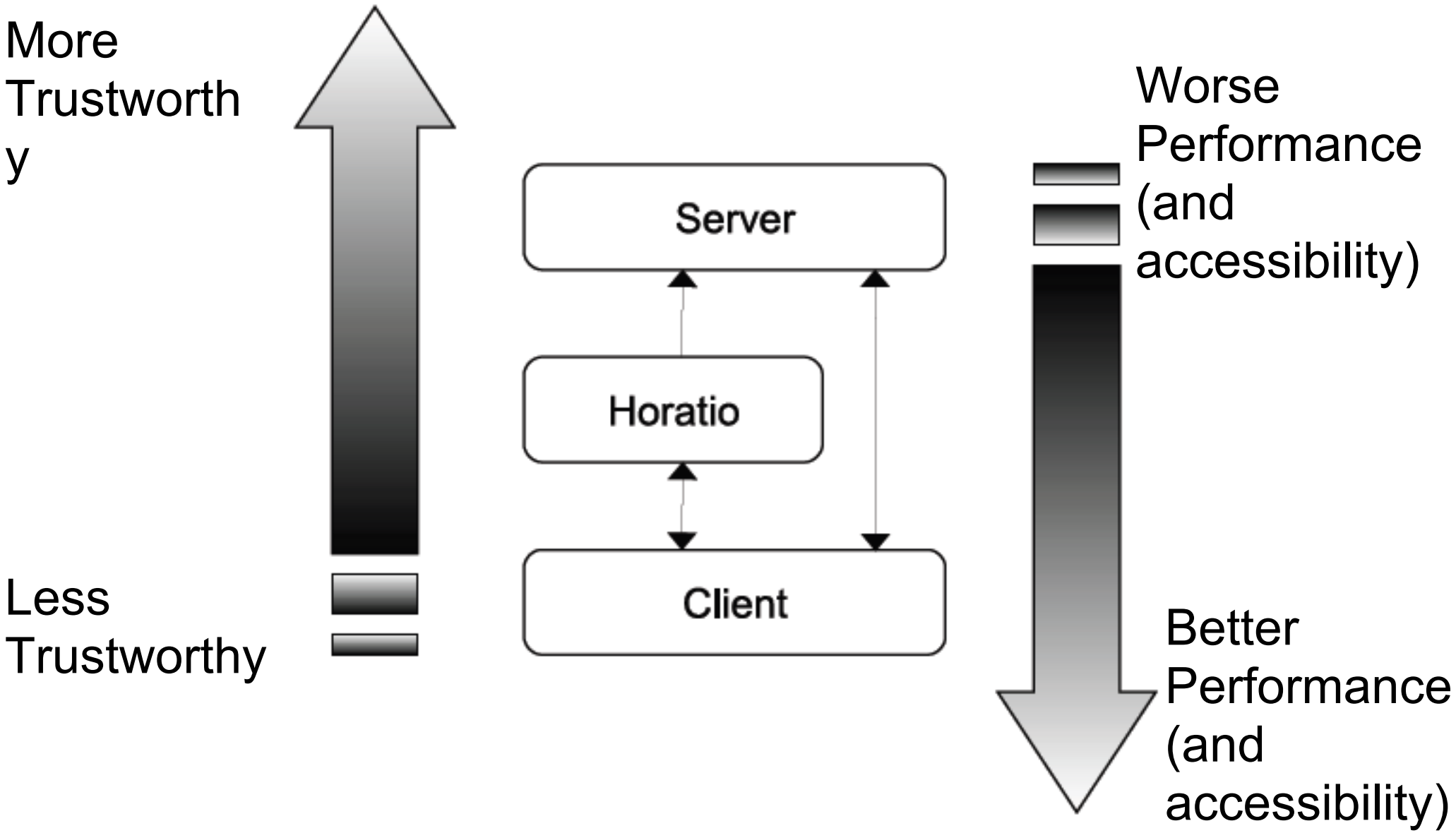


- Low-bandwidth, unreliable data link
  - e.g. 3G, Wi-Fi, WiMax
- Link should supplement client-server link when client-server link is unavailable

Potential Caveats

- Smartphones very constricted in terms of storage, processing power, memory, battery, network interfaces, etc.
  - using a smartphone as a "mini-server" may not be feasible
- Smartphone not always available (flat battery)
- Not every user will have a smartphone

Trust versus Performance



Ordinary Client-Server ISR Operation

(without Horatio)

1. Client attempts resume, requests VM image from server
2. VM sent to client over Internet
3. Client boots VM image to replicate user's desktop environment
4. Upon shutdown of client, changes in VM sent back to server

Ordinary Client-Server ISR Operation

(without Horatio)

1. Client attempts resume, requests VM image from server
2. VM sent to client over Internet
   - high network utilisation
3. Client boots VM image to replicate user's desktop environment
   - client's work after boot vunerable until successful shutdown
4. Upon shutdown of client, changes in VM sent back to server
   - high network utilisation
   - lack of high-quality network connection renders client's work lost

ISR with Horatio Operation

1. Client attempts resume, requests VM image
2. VM served to client over Internet or smartphone
   - depending on ownership, and internet connection speed
3. Client boots VM image to replicate user's desktop environment
   - Horatio device may use Eager State Propagation to reduce following suspend latency
4. Upon shutdown of client, changes in VM sent back to Horatio
   - fast transfer of state to the smartphone, no connection to server required
   - Horatio device deals with sending dirty state to server
   - client can send the data to server if conditions allow

Parcels, Ownership, and "Dirtiness"

Parcels

- A "parcel" is a VM image
  - Two parts ("states") to a parcel: "control" and "data"
    - Data part: encrypted virtual disk image and memory image
    - Control part: metadata about parcels (keyring for decryption, configuration, ownership nonce, etc.)

State Name	Type	Typical Size	Description
Memory Image	data	200 MB	Encrypted and compressed memory image and registers.
Disk Image	data	3.5 GB	Individually encrypted and compressed chunks of virtual disk.
Keyring	control	5.5 MB	Encryption keys and cryptographic hashes of virtual disk chunks. Encrypted with a key stored in the configuration file.
Configuration File	control	500 bytes	Operational parameters of a parcel and encryption key used to encrypt the keyring and memory image.
Ownership Nonce	control	10 bytes	A unique identifier generated when a parcel is checked out from an ISR server.

Parcels, Ownership, and "Dirtiness"

Dirtiness

- Parcel defined as "clean" when first pulled from server
- If parcel is modified, then parcel declared "dirty"
  - Dirty parcels need to be sent back to the server
- Smartphone must keep copies of all dirty data which has not yet been successfully sent to the server
- Caching dirty parcel on smartphone reduces network dependency

Parcels, Ownership, and "Dirtiness"

Ownership

- Parcel is owned by one of: client, smartphone, server
  - Ownership nonce required for decryption of data parcel, thus enforcing ownership
  - Server generates nonce for each valid request for a parcel; transfers ownership upon fulfilling request
    - Server can only serve parcel which it owns

Entity that owns parcel has the most recent version of parcel

Parcels, Ownership, and "Dirtiness"

Ownership Transfer

1. Source confirms successful transfer of parcel data to destination
2. Keyring and configuration transferred after confirmation of successful data parcel transfer
3. Nonce transferred between entities through encrypted two-phase commit protocol
4. Client and smartphone trusted to forget nonce (i.e. relinquish ownership) after successfully transferring ownership

Opportunism

- Horatio assumes that only the server is robust
  - Client and smartphone fragile and prone to losing data
- Horatio tries to send data cached on smartphone to server at earliest available opportunity ("self-cleaning")
  - utilises any available smartphone network connection (3G, Wi-Fi, etc.)

Optimisations

"Concurrent Upload from Multiple Sites"

- i.e. from both client and smartphone

"Memory Image Differencing"

- *upload only the differences of parcels that need to be updated, rather than the whole parcel*

"Eager State Propogation"

- Transfer parcels of modified VM image to smartphone upon modification of that section of image, rather than at shutdown of client
  - reduces time taken to shutdown

Evaluation of Horatio

Stated Goals

- Horatio seeks to *"alleviate...ISR limitations"* and reduce:
  - *"resume latency"* - large download to client at startup
  - *"slowdown"* - parcels transferred to server during a session
  - *"suspend latency"* - remaining parcels of modified VM state uploaded to server at shutdown

Evaluation of Horatio

Questions Asked

- How big is the user experience improvement
- How well does self-cleaning perform
- What is the impact on mobility footprint
- How effective is eager propagation

Evaluation Approach

- Microbenchmarks
  - synthetically generated ISR 'dirty' states
- Macrobenchmarks
  - scripted workloads of typical activities scheduled over time.
- Hardware Setup:
  - Openmoko Neo FreeRunner (Linux-based, WiFi, GPRS)
  - Nokia N95-8GB (Symbian v9, WiFi, 3G)
  - Sandisk Mobile Ultra (2 GB storage, Hi-Speed USB connection - 480 Mbps)

User Experience Improvement

Microbenchmark Results

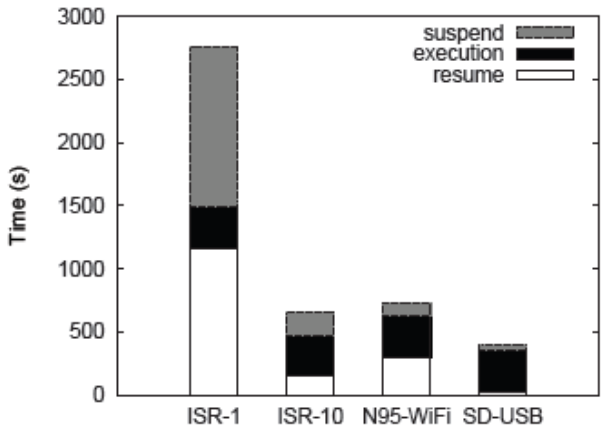
- Suspend results:

Horatio Device	Dirty State Size 1 MB	10 MB	100 MB	500 MB
ISR-1 (No Horatio)	1433.5(6.5)	1487.5(0.5)	2118.0(3.0)	4936.0(15.0)
N95-WiFi	39.0(2.5)	68.7(2.4)	301.3(5.0)	1239.2(27.5)
OM-WiFi	32.7(2.1)	48.0(0.8)	260.3(5.3)	1040.0 (8.6)
N95-USB	29.3(4.4)	44.0(2.5)	142.0(1.3)	625.0 (4.3)
SD-USB	23.3(0.5)	25.3(0.5)	40.3(1.3)	136.0 (2.2)

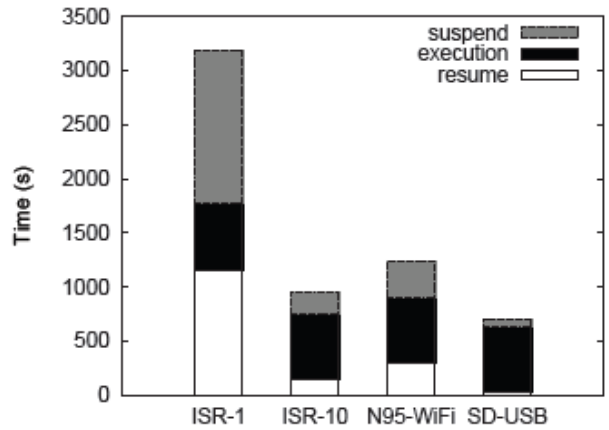
Horatio Device	Dirty State Size 0 MB	1 MB	10 MB	100 MB	500 MB
ISR-1 (No Horatio)	281.5(1.5)	-	-	-	-
N95-WiFi	-	397.7(2.5)	409.3 (2.3)	507.5(6.0)	951.2(20.1)
OM-WiFi	-	292.3(1.7)	314.7(12.0)	372.0(2.2)	692.0 (2.2)
N95-USB	-	226.0(3.2)	237.0 (6.7)	283.8(0.7)	520.8 (2.5)
SD-USB	-	34.7(0.5)	34.3 (0.9)	35.7(0.5)	51.3 (0.9)

User Experience Improvement

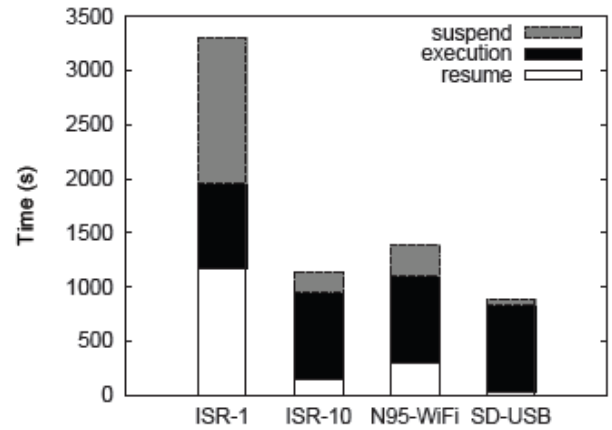
Macrobenchmark results



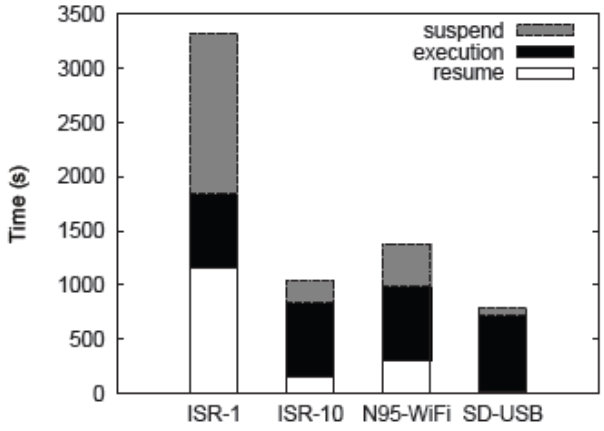
(a) Email



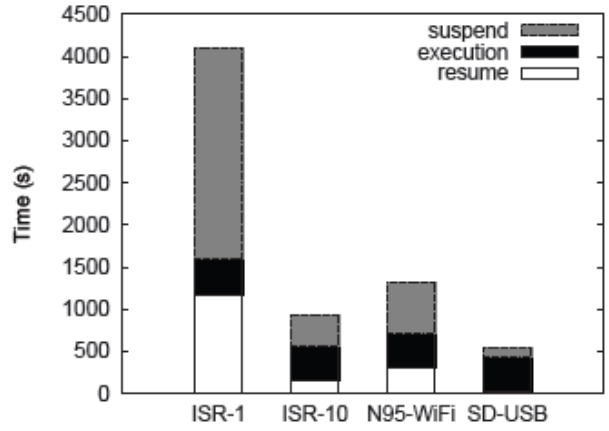
(b) Word



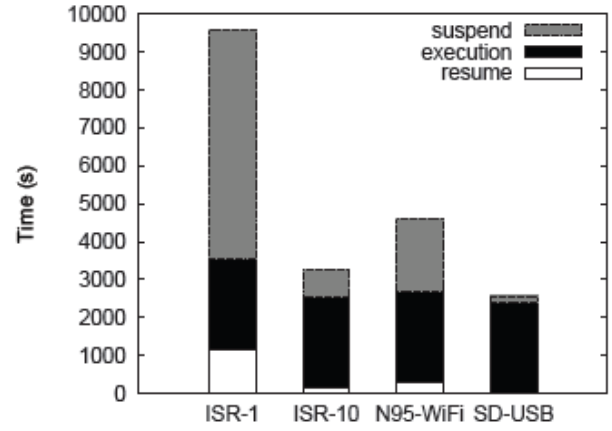
(c) Photo



(d) Shop



(e) Podcast



(f) Video

Effectiveness of Self-Cleaning

Experiments performed on the Nokia N95 only

Microbenchmark:

	Dirty State Size		
Horatio Device	1 MB	10 MB	100 MB
N95-WiFi	36.3(0.9)	97.0 (0.0)	869.0 (15.1)
OM-WiFi	13.0(0.0)	82.3 (0.5)	775.0 (0.8)
N95-3G	152.7(1.3)	477.7(14.4)	3848.3(102.8)

Macrobenchmark:

Workload	N95-WiFi	N95-3G
Email	213.3 (14.0)	739.3 (5.3)
Word	953.0 (9.4)	4353.5 (90.5)
Photo	839.3 (19.0)	3381.0(129.0)
Shop	1103.0 (50.3)	4830.0(313.0)
Podcast	2199.7(176.3)	*6398*
Video	*8034*	*23665*

Impact on Mobility Footprint

Energy expended by the smartphone on Horatio-related tasks

Operation	Horatio Device	Dirty State Size 1 MB	10 MB	100 MB	500 MB
Suspend	N95-WiFi	27.5 (1.0)[0.2%]	71.2 (3.9)[0.4%]	400.1 (1.4) [2.5%]	1788.8 (8.9)[11.0%]
Suspend	N95-USB	12.0 (2.5)[0.1%]	31.3 (0.8)[0.2%]	146.8 (3.8) [0.9%]	608.5(14.2) [3.7%]
Resume	N95-WiFi	507.1(40.6)[3.1%]	612.8 (1.0)[3.8%]	756.3 (15.5) [4.6%]	1455.7(33.4) [8.9%]
Resume	N95-USB	95.5 (1.6)[0.6%]	96.8 (1.7)[0.6%]	120.0 (1.2) [0.7%]	226.6 (0.5) [1.4%]
Self-Clean	N95-WiFi	35.7 (0.7)[0.2%]	102.6 (1.4)[0.6%]	915.6 (3.2) [5.6%]	-
Self-Clean	N95-3G	180.6 (4.3)[1.1%]	565.1(13.4)[3.5%]	4552.7(107.8)[27.9%]	-

Eager State Propagation

Performance Evaluation

Purely performance/usability improvement point-of-view evaluation

Workload	Suspend State	Eager State	Lazy State	Cleaning Cycles
Email	3.1	129.6	19.5	3.0
Word	1.7	220.8	44.1	3.3
Photo	1.6	199.1	28.4	6.7
Shop	29.3	485.7	44.4	11.0

Evaluation Summary

Horiatio is capable of:

- enabling usage even with no internet connection
- improving experience significantly with poor internet connectivity of the client
- reducing mobility footprint

but also:

- has the potential of draining the smartphone's battery excessively is come cases
- introduces vulnerability to loss/damage compared to pure ISR

Future Work

In the future, research on Horatio will include:

- resume location prediction
- more robust handling of transient network connectivity
- user interface

Conclusion

Horatio is a good contribution to cloud computing:

- data and control decoupling (advanced VM handling)
- providing a robust alternative to reducing mobility footprint
- mitigating poor internet connections

However constantly evolving and fast growing areas of smartphones and virtualisation mean:

- uncertainty of smartphones as a feasible platform for Horatio
- hardware and VM novelties, design changes, may soon deem Horatio inadequate and/or unnecessary
- also, difficult to deploy widely due to limited popularity of ISR