The Slothful Ways of D-Bus

Will Thompson
Collabora
Pre-emptive thanks:

Robin Bate Boerop
Alban Crequy
“Slow”

- **How** slow?
- High latency?
  - ...perhaps due to context switching?
- Low throughput?
- Scales poorly as traffic/number of clients increases?
- Something else?
Preamble on methodology

- Benchmarks are benchmarks
- libdbus
- Statistics taken from various devices—not directly comparable
Journey of a message

- Alice constructs a message, and gives it to the bus;
- The bus stamps the message header with Alice's unique name, and sends it out to its destination (maybe) and other interested recipients.
Context switches?

- 1-byte ping-pong using UNIX sockets:
  - 10µs on a modern-ish laptop.
  - (40µs on a Nokia N900.)
- 1-byte (body) D-Bus message ping-pong:
  - 225µs
Message throughput
For comparison

- Best D-Bus throughput: ~37MB/s (32KB messages)
- Best bare socket throughput: 2GB/s (1MB messages)
- Bare socket throughput with 32KB messages: 647MB/s
...and furthermore

- Best D-Bus throughput: ~37MB/s (32KB messages)
- My desktop’s mediocre hard disk writes at 34MB/s and reads at 40MB/s
U+1F63E POUTING CAT FACE
Copies and validations

- Sender marshals message
- Sender writes message to socket
- Daemon reads message from socket
- Daemon validates message
- Daemon writes message to $r$ sockets
- $r$ recipients read the message
- $r$ recipients validate the message
- $4 + 2r$ copies, $1 + 2r$ validations
“AFAICT dbus is just a replacement for a directory tree with unix sockets under /tmp. Either is equally functional.”
What does D-Bus give us?

- Flexible, widely-supported serialization format
- Name service
- Broadcast notifications
- Causal-order delivery among n processes
- Authentication and security policies
- Service activation
- De-facto standard desktop IPC
What does D-Bus give us?

- Flexible, widely-supported serialization format
- Name service
- Broadcast notifications
- Causal-order delivery among n processes
- Authentication and security policies
- Service activation
- De-facto standard desktop IPC
How about epoll?

<table>
<thead>
<tr>
<th>50000 ping-pongs on...</th>
<th>an empty bus</th>
<th>a bus with standard services, all idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-Bus 1.4 using poll</td>
<td>0:54</td>
<td>1:45</td>
</tr>
<tr>
<td>Branch using epoll</td>
<td>0:56</td>
<td>0:55</td>
</tr>
</tbody>
</table>

(The fine work of Simon McVittie.)
U+1F639
CAT FACE WITH TEARS OF JOY
Revenge of the context switch

- A service emits 10 signals in immediate succession
- 10 other services have match rules matching them
First message

- One context switch from sender to D-Bus daemon;
- Daemon writes the message into 10 sockets;
- A context switch to each recipient, in turn;
- One final switch back to the sender.
- Total: 12
Subsequent messages

- One context switch from sender to D-Bus daemon;
- Daemon writes the message into one socket; gets descheduled;
- That recipient reads the message; daemon gets rescheduled;
- Repeat nine times;
- One final switch back to the sender.
- Total: 22
Raise the daemon's priority?

- Can o' priority inversion worms!
- (But makes this case twice as fast.)
Self-promotion time!
Bustle now uses pcap files
Bustle now uses pcap files

- A big stream of eavesdropped messages…
- …plus ListNames() and a bunch of GetNameOwners() to make the logs more usable.
- Possibly inaccurate timestamps.
- Service activation time invisible.
- Good enough for now!
Method call
Path: /org/gnome/PowerManager
Member: org.freedesktop.DBus.Properties.Get
Arguments: "s" "org.gnome.PowerManager"

"s" "Icon"
Method return
Path: /org/gnome/PowerManager
Member: org.freedesktop.DBus.Properties.Get
Arguments: · "v" (Variant "s" . GThemedIcon battery—full—charging—symbolic gpm—battery—100—charging battery—full—charging )
Unticking a service hides its column in the diagram, and all messages it is involved in. That is, all methods it calls or are called on it, the corresponding returns, and all signals it emits will be hidden.

- org.gnome.PowerManager
- org.freedesktop.Notifications
- org.freedesktop.Telepathy.Client.GnomeShell_3a1_2e18.n0
- org.gnome.Magnifier
- org.gnome.Panel
- org.gnome.Shell
  - org.freedesktop.Telepathy.Connection.gabble.jabber.will_2ejoseph_2ethompson_2e111
  - org.freedesktop.Telepathy.Connection.gabble.jabber.will_2ethompson_40colla
  - org.freedesktop.Telepathy.Connection.gabble.jabber.will_40willthompson_2e111
- org.freedesktop.Telepathy.ConnectionManager.gabble
- org.freedesktop.Telepathy.Client.Empathy.ChatroomManager_3a1_2e1611.n0
- org.freedesktop.Telepathy.AccountManager

Method return
Path: /org/gnome/PowerManager
Member: org.freedesktop.DBus.Properties.Get
Arguments: • "v" (Variant "s" "GThemedIcon battery-full-charging-symbolic gpm-battery-100-charging battery-full-charging ")
### Message frequencies

<table>
<thead>
<tr>
<th>Method/Signal</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>org.freedesktop.Telepathy.Channel.Type.Text.Send</td>
<td>2</td>
</tr>
<tr>
<td>org.freedesktop.DBus.Properties.Get</td>
<td>2</td>
</tr>
<tr>
<td>org.gnome.PowerManager.Changed</td>
<td>2</td>
</tr>
<tr>
<td>org.gnome.PowerManager.ChangePrimaryDevice</td>
<td>1</td>
</tr>
<tr>
<td>org.gnome.PowerManager.GetDevices</td>
<td>1</td>
</tr>
<tr>
<td>org.freedesktop.Telepathy.Client.Handler.HandleChannels</td>
<td>1</td>
</tr>
<tr>
<td>org.freedesktop.ChannelDispatcher.EnsureChannel</td>
<td>1</td>
</tr>
<tr>
<td>org.freedesktop.Channel.Type.Text.AcknowledgePendingMessages</td>
<td>1</td>
</tr>
</tbody>
</table>

### Method durations

- will_2joseph_2... event Count (4ms)
- will_2joseph_2... event Call (1ms)
- will_2joseph_2... event Count (1ms)
- will_2joseph_2... event Call (0ms)
- will_2joseph_2... event Count (4ms)
- will_2joseph_2... event Call (1ms)
- will_2joseph_2... event Count (4ms)
- will_2joseph_2... event Call (1ms)
- will_2joseph_2... event Count (4ms)
- will_2joseph_2... event Call (1ms)
- will_2joseph_2... event Count (4ms)
- will_2joseph_2... event Call (1ms)
- will_2joseph_2... event Count (4ms)
- will_2joseph_2... event Call (1ms)
- will_2joseph_2... event Count (4ms)
- will_2joseph_2... event Call (1ms)
- will_2joseph_2... event Count (4ms)
- will_2joseph_2... event Call (1ms)
- will_2joseph_2... event Count (4ms)
- will_2joseph_2... event Call (1ms)
- will_2joseph_2... event Count (4ms)
- will_2joseph_2... event Call (1ms)
- will_2joseph_2... event Count (4ms)
- will_2joseph_2... event Call (1ms)
Coming soon to a distro near you!
BOF “On D-Bus”

Wednesday, 17:00, room 1.401/2
Any questions?

http://www.flickr.com/photos/pelhamgrenville/5931208532/